UNIVERSITÄT
DES
SAARLANDES

Unexpected Routes: BGP Prefixes Beyond Recommended Practices

A dissertation submitted towards the degree
Doctor of Engineering (Dr. - Ing.)

of the Faculty of Mathematics and Computer Science of Saarland University

by
Khwaja Zubair Sediqi

Saarbrücken, 2025

Date of the colloquium:	17.02.2026
Dean:	Professor Dr. Roland Speicher
Chairman of the examination board:	Professor Dr. Ingmar Weber
Reporter:	Professor Anja Feldmann, PhD Professor Dr. Oliver Hohlfeld Dr. Oliver Gasser
Scientific Assistant:	Tiago Heinrich, PhD

Notes on style:
As most of the work presented in this dissertation was done in collaboration with other researchers, the scientific plural "we" is used.

Saarland University
Faculty MI – Mathematics and Computer Science
Department of Computer Science
Campus - Building E1.1
66123 Saarbrücken
Germany

Eidesstattliche Versicherung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form in einem Verfahren zur Erlangung eines akademischen Grades vorgelegt.

Declaration of original authorship

I hereby declare that this dissertation is my own original work except where otherwise indicated. All data or concepts drawn directly or indirectly from other sources have been correctly acknowledged. This dissertation has not been submitted in its present or similar form to any other academic institution either in Germany or abroad for the award of any other degree.

Saarbrücken, October 2025

gez. / signed

Khwaja Zubair Sediqi

Acknowledgements

I began my Ph.D. during COVID-19, working from home while our family moved from Afghanistan to Germany. Starting any doctorate is demanding; doing so amid lockdowns, remote work, and parenting three children, then welcoming our fourth was not an easy mission. This dissertation exists because many people helped in ways large and small, and I am deeply grateful for their guidance, patience, and practical support.

First and foremost, I am deeply grateful to my advisor, Prof. Anja Feldmann. Her guidance, high standards kept this dissertation on track. Her meeting questions made me state the problem precisely, and her notes on drafts improved the writing. From her, I learned to aim for quality while staying productive.

Thank you to Oliver Gasser. I'm extremely grateful for your mentorship, advice, and continues support over the years. You taught me how to frame a problem, plan an evaluation, and write a paper that says precisely what it means. Your discipline and attention to detail improved my standards.

Thank you Said Jawad Saidi, for being an exceptional friend. I learned a lot from your experience and Ph.D. journey, and your steady encouragement carried me through hard weeks. I enjoyed our lunches and tea breaks discussion, which became plans and next steps. Thank you Lars Prehn. You are a talented researcher with a great sense of humor that never hesitated to help. Working with you on the HSP paper was fun.

I am grateful to my co-author, Fariba Osali, for her consistent collaboration and effort on our sibling prefixes paper. I also thank Romain Fontugne, Amreesh Phokeer, Massimiliano Stucchi, and Massimo Candela for their substantial contributions to our RPKI synchronization paper. Working with you made our research enjoyable.

To my fellow researchers and colleagues in the INET group, many of whom I now call friends, thank you. You are among the brightest people I have met, and it was a privilege to work with you and learn from our discussions. Our daily lunches at the Mensa and the small talks are memories I will keep. Thank you to the INET administrative team for making research and administrative matters easy to handle. Special thanks to Iris Wagner for your kindness, support, and help to me and my family throughout this Ph.D. journey. I sincerely thank Saarland University and the Max Planck Institute for Informatics for providing me with the opportunity to conduct my research and pursue my Ph.D. degree.

I am grateful to my extended family, including my grand father, uncles, aunts, cousins, and my in-laws. Your encouragement and prayers over the years sustained me. My childhood friend Abdul Ghafar Sediqi always believed in me and kept my energy up during my Ph.D.; his and my old friends from school and village encouragement meant more than they know.

Last but not least, I thank my family, the most valuable part of my life. My wife, Fatema Sediqi, supported me and our children, Zahra, Abdul Shakoor, Aiesha, and Osman Sediqi, with patience and strength. She carried the extra load when deadlines were heavy and still stood by me. I am so grateful to my parents, Toorpaykai and Khwaja Gul Sediqi. Their sacrifices and steady prays, and guidance made this path possible, and they supported our family at every turn. My siblings, Khwaja Monib, Aseela (Haseeba), Husna, and Khwaja Shoaib Sediqi, encouraged and supported me through good times and dark times. I love you all more than I can say.

Abstract

The Internet is composed of a vast collection of interconnected networks, also known as Autonomous Systems (ASes). ASes use Border Gateway Protocol (BGP) to exchange the reachability information of IP prefixes. The Resource Public Key Infrastructure (RPKI) enhances BGP security by providing cryptographically verifiable objects that confirm the ownership of the IP prefix by an AS. A set of well-documented best practices and guidelines for route announcements governs the Internet's interdomain routing between ASes. For proper operation of Internet routing, adherence of network operators to the recommended norms and best practices is important. Among these best practices are the recommendations for using prefix lengths up to /24 for IPv4 and up to /48 for IPv6, single origin AS for IP prefix announcement, and registering a single prefix per Route Origin Authorization (ROA) object in RPKI.

However, not all network operators follow these recommendations, and instead, their route announcements might be based on their policies, business needs, or technical limitations. Deviating from best routing practices can lead to routing inconsistency, complicate prefix origin validation, and disrupt network performance. This dissertation examines the routing ecosystem for violations of the aforementioned best practices. We define these cases as "unexpected routes" because they represent routes that are less anticipated and remain underexplored in prior research.

First, we examine the routing ecosystem of the Internet for IP prefix sizes that are too specific. More precisely, we focus on IP prefixes more specific than /24 in IPv4 (i.e., /25 to /32) and than /48 for IPv6 (i.e., /49 to /128), and we refer to these prefixes as Hyper-specific Prefixes (HSPs). We analyze over eleven years of BGP data from well-known route collector projects to understand the evolution, examine their BGP communities and CIDR sizes to understand the reasons for HSP existence, and the potential role they might serve in Internet routing. Our findings show that most HSPs are accidental (internal) route leaks, or infrastructure peering subnets, and BGP blackholing.

Next, we examine the origin AS for IP prefixes announced via BGP to the Internet. Using single-origin AS for a prefix is recommended; however, the routing ecosystem of the Internet exhibits several thousand prefixes having Multi Origin AS (MOAS) prefixes. We analyze MOAS prefixes, using over six years of daily BGP Routing Information Base (RIB) snapshots from route collectors to examine the lifespan, propagation pattern, and potential relationship between the origin ASes of MOAS prefixes and the reason for MOAS prefixes' existence on the Internet. Our findings reveal that company mergers are the largest contributors to MOAS prefixes, and examining their CIDR sizes indicates their potential for fine-grained traffic engineering. Hypergiants, including Google and Amazon, are also among the users of MOAS prefixes.

Then, we analyze the relationship between IPv4 and IPv6 address families at the prefix level. We use DNS records hosted on IPv4 and IPv6 prefixes and apply the Jaccard similarity index to identify pairs of IPv4 and IPv6 prefixes having a similar set of DNS records on their IPs and refer to them as sibling prefix pairs. We identify 76k IPv4-IPv6 sibling pairs, and 60% of the sibling prefixes are registered in the RPKI.

Finally, we examine the current ROA structure across five RIRs' RPKI trees for single prefix per ROA recommendation and analyze the RPKI validation delay by setting a testbed. We find that the current ROA structure across five RIRs is not the same, and the network delay and cryptographic verification of ROAs are the main delay contributors in the RPKI synchronization process.

Zusammenfassung

Das Internet besteht aus einer großen Sammlung miteinander verbundener Netzwerke, die als *Autonomous Systems (ASes)* bezeichnet werden. ASes verwenden das *Border Gateway Protocol (BGP)*, um Informationen über die Erreichbarkeit von IP-Präfixen auszutauschen. Die RPKI verbessert die Sicherheit von BGP, indem sie kryptografisch überprüfbare Objekte bereitstellt, die den Besitz eines IP-Präfixes durch ein AS bestätigen. Eine Reihe gut dokumentierter Best Practices und Richtlinien für Routenankündigungen regelt das Interdomain-Routing zwischen ASes im Internet. Für den ordnungsgemäßen Betrieb des Internet-Routings ist es wichtig, dass Netzbetreiber die empfohlenen Normen und Best Practices einhalten. Zu diesen Empfehlungen gehören unter anderem die Verwendung von Präfixlängen bis zu /24 für IPv4 und /48 für IPv6, die Nutzung eines einzelnen Origin-AS für die Ankündigung eines IP-Präfixes sowie die Registrierung eines einzelnen Präfixes pro ROA-Objekt in der RPKI.

Nicht alle Netzbetreiber halten sich jedoch an diese Empfehlungen. Stattdessen können ihre Routenankündigungen auf eigenen Richtlinien, geschäftlichen Anforderungen oder technischen Einschränkungen beruhen. Abweichungen von bewährten Routing-Praktiken können zu Routing-Inkonsistenzen, erschwerter Prefix-Origin-Validierung und Beeinträchtigungen der Netzwerkleistung führen. Diese Dissertation untersucht das Routing-Ökosystem auf Verstöße gegen die genannten Best Practices. Wir definieren diese Fälle als “unexpected routes”, da sie weniger erwartete Routen darstellen, die in bisherigen Forschungen kaum untersucht wurden.

Zunächst untersuchen wir das Routing-Ökosystem des Internets im Hinblick auf zu spezifische IP-Präfixgrößen. Genauer gesagt konzentrieren wir uns auf IP-Präfixe, die spezifischer sind als /24 bei IPv4 (i.e., /25 bis /32) und spezifischer als /48 bei IPv6 (i.e., /49 bis /128). Diese Präfixe bezeichnen wir als *Hyper-specific Prefixes (HSPs)*. Wir analysieren über elf Jahre BGP-Daten aus bekannten Route-Collector-Projekten, um ihre Entwicklung zu verstehen, ihre BGP-Communities und CIDR-Größen zu untersuchen, die Gründe für ihre Existenz zu identifizieren und ihre mögliche Rolle im Internet-Routing zu bewerten. Unsere Ergebnisse zeigen, dass die meisten HSPs auf versehentliche (interne) Route-Leaks, Infrastruktur-Peering-Subnetze oder BGP-Blackholing zurückzuführen sind.

Als Nächstes analysieren wir die Origin-ASes von IP-Präfixen, die über BGP im Internet angekündigt werden. Obwohl empfohlen wird, für ein Präfix nur ein einzelnes Origin-AS zu verwenden, zeigt das Routing-Ökosystem des Internets mehrere tausend Präfixe mit *Multi-Origin AS (MOAS)*. Wir untersuchen MOAS-Präfixe anhand von über sechs Jahren täglicher BGP-RIB-Snapshots aus Route-Collector-Daten, um ihre Lebensdauer, Verbreitungsmuster und potenzielle Beziehungen zwischen den Origin-ASes zu analysieren sowie die Gründe für das Auftreten von MOAS-Präfixen im Internet zu verstehen. Unsere Ergebnisse zeigen, dass Unternehmensfusionen die Hauptursache für MOAS-Präfixe darstellen, und eine Betrachtung ihrer CIDR-Größen deutet auf eine mögliche Nutzung für feingranulares Traffic Engineering hin. Große Anbieter wie Google und Amazon gehören ebenfalls zu den Nutzern von MOAS-Präfixen.

Anschließend untersuchen wir die mögliche Beziehung zwischen den Adressfamilien IPv4 und IPv6 auf Präfixebene. Wir verwenden DNS-Einträge, die auf IPv4- und IPv6-Präfixen gehostet sind, und wenden den Jaccard-Ähnlichkeitsindex an, um Paare von IPv4- und IPv6-Präfixen mit ähnlichen DNS-Einträgen zu identifizieren, die wir als *sibling prefix pairs* bezeichnen. Wir identifizieren 76k IPv4–IPv6 sibling prefixes, von denen 60% in der RPKI registriert sind.

Abschließend analysieren wir die aktuelle ROA-Struktur in den RPKI-Bäumen der fünf RIRs im Hinblick auf die Empfehlung, nur ein einzelnes Präfix pro ROA zu registrieren, und untersuchen die RPKI-Validierungsverzögerung mithilfe eines aufgebauten Testbeds. Wir stellen fest, dass sich die aktuelle ROA-Struktur zwischen den fünf RIRs unterscheidet und dass Netzwerklatenzen sowie die kryptografische Verifikation der ROAs die Hauptursachen für Verzögerungen im RPKI-Synchronisationsprozess sind.

List of Publications

Parts of this dissertation are based on published work in peer-review conferences or journal. These works are co-authored with other researchers as listed below.

Peer-reviewed Journal

- **Khwaja Zubiar Sediqi**, Lars Prehn, and Oliver Gasser. Hyper-Specific Prefixes: Gotta Enjoy the Little Things in Interdomain Routing. ACM SIGCOMM Computer Communication Review 52.2 (2022): 20-34. doi: 10.1145/3544912.3544916 [Results appear in Chapter 3.]

International Conferences

- **Khwaja Zubair Sediqi**, Anja Feldmann, and Oliver Gasser. Live Long and Prosper: Analyzing Long-Lived MOAS Prefixes in BGP. 7th Network Traffic Measurement and Analysis Conference (TMA), Naples, Italy, 2023, pp. 1-10. doi: 10.23919/TMA58422.2023.10198973. **Fast track award** [Results appear in Chapter 4.]

- Fariba Osali, **Khwaja Zubair Sediqi**, and Oliver Gasser. Sibling Prefixes: Identifying Similarities in IPv4 and IPv6 Prefixes. In Proceedings of the 2025 ACM Internet Measurement Conference (IMC '25), October 28–31, 2025, Madison, WI, USA. ACM, New York, NY, USA, 20 pages. doi: 10.1145/3730567.3732917 [Results appear in Chapter 5]

- **Khwaja Zubair Sediqi**, Romain Fontugne, Amreesh Phokeer, Massimiliano Stucchi, Massimo Candela, Anja Feldmann. RPKI Syncing: Delay in Relying Party Synchronization. 9th Network Traffic Measurement and Analysis Conference (TMA), Copenhagen, Denmark, 2025. doi: 10.23919/TMA66427.2025.11097000. **Best paper award** [Results appear in Chapter 6]

Contents

Chapter 1
Introduction

The Internet plays a key role in our day-to-day life. Approximately 5.3 billion people around the world use Internet services for purposes such as communication, accessing information, remote work, video streaming, gaming, and other online services [113, 160, 172, 185, 274]. The Internet operates as a decentralized system of interconnected networks with a routing structure built on cooperation, trust, and standards developed by Internet regulatory bodies such as the Internet Engineering Task Force (IETF) [159, 210]. Internet Service Providers (ISPs) provide Internet services to customers. Larger ISPs route and carry the traffic of small ISPs. The Internet Exchange Points (IXPs) around the world [206] provide easy interconnection and Border Gateway Protocol (BGP) peering facilities for network operators, and Content Delivery Networks (CDNs) deploy content to end users as close as possible [16, 29, 64, 77, 291].

The Internet infrastructure is expanding, and in addition to computers, servers, and phones, it provides the opportunity for new services like smart appliances and Internet of Things (IoT) devices to connect [12]. Devices use Internet Protocol (IP) and IPv4 or IPv6 address, a unique identifier of the device in the network, to communicate with each other on the Internet. Every collection of Internet-connected devices belongs to a different network under a single administrative control called an Autonmous System, identified by a unique Autonmous System Number (ASN) on the Internet.

ASes use Border Gateway Protocol as the de facto interdomain routing protocol to exchange routing information with each other and to deliver IP packets from sources to destination AS. Network operators establish BGP peering sessions with other ASes on the Internet to announce the reachability information for their IP prefix, a range of contiguous block of IP addresses, associated with the ASN to the rest of the Internet [221]. The interconnection of these ASes makes the Internet a network of networks acting as the global backbone for many technology services. In other words, the Internet is a vast collection of independent networks that speak the same language (IP and BGP) and exchange routes.

Every announcement of an IP prefix and ASN forms a route in the global routing table of the Internet for the announced IP address range. Observation of routes through Réseaux IP Européens (RIPE) Routing Information System (RIS) [232], RouteViews [281], or Isolario [144] route collectors, representing a sampled and partial view of the complete

set of IP prefix announcements exchanged between ASes, reveal the drastic increase of BGP routes in the global routing table of the Internet. The number of BGP announced routes in the global routing table of the Internet has evolved from less than 50,000 routes in the 1990s to more than one million routes originated by tens of thousands of ASes in 2025 [46, 131, 212] making the global routing table a complex routing system.

BGP inherently lacks security mechanisms to verify legitimate AS for IP prefixes, making it vulnerable to BGP routing incidents i.e., BGP route hijack. BGP route hijack happens when a misconfigured or malicious network announces IP prefixes for which it is not the owner, causing diversion of the victim network's traffic through it - potentially for traffic interception, blocking, or surveillance. The Resource Public Key Infrastructure was introduced to improve BGP security against routing incidents for instance BGP route hijacks. RPKI provides cryptographically verifiable objects that confirm the ownership of the IP prefix by an AS, which then can be used by a BGP-speaking router to perform the Route Origin Validation (ROV) [20, 179].

To maintain a secure and scalable operation of BGP routing, the adherence of ISPs, IXPs, CDNs, and other network entities to the operational Best Current Practice (BCP), filtering policies, and routing security standards is important. Failure to follow recommended routing best practices can add complexity in Interdomain routing, lead to service disruption, or performance degradation [101, 255, 284, 295]. Disruption in the reachability of the IP prefix of an AS can impact all the services running on IPs of that prefix. This might lead to economic loss, reputational loss, or security and policy concerns for the network operators.

Among these Best Current Practice are the recommendations to limit the IP prefix size announcement in interdomain routing commonly, up to /24 for IPv4 and /48 for IPv6, to reduce routing table growth, avoid unnecessary prefix de-aggregation [82, 202]. Popular filtering guidelines also recommend the rigorous filtering of prefixes encompassing only a few addresses [81, 110, 183]. Originating an IP prefix from a single AS and using a single IP prefix per ROA object in the RPKI operation is also recommended [82, 114, 165, 287, 293]. Unfortunately, the reality of the Internet's interdomain routing often exhibits various anomalies. Notable among them are prefixes with Multiple Origin AS (MOAS), where the same prefix originated by multiple ASes, prefixes having too specific Classless Interdomain Routing (CIDR) sizes than /24 (i.e., /25 to /32) for IPv4 and /48 for IPv6(i.e., /49 to /128), and bundling multiple prefixes to a ROA object of RPKI.

BGP prefix hijack incidents are one reason for the MOAS prefixes. In addition to the BGP prefix hijack cases, there are also MOAS prefixes that are observable for an extended period of time (i.e., at least 30 days) in the global routing table of the Internet. While MOAS prefixes deviate from best routing practices because of their multiple origin ASes, there are also routes in the Internet routing ecosystem that violate the best routing practices for the IP prefix size. Typically, every network announces its IP prefixes with up to /24 in IPv4 and up to /48 in IPv6 CIDR sizes. However, IP prefixes with CIDR size more specific than /24 in IPv4 and more specific than /48 in IPv6 are still in use in interdomain routing and are detectable in the global routing table of the Internet.

In addition, most BGP studies treat IPv4 and IPv6 as separate routing spaces. IPv4 and IPv6 prefixes that share the same or similar set of Domain Name System (DNS) records, such as A and AAAA records pointing to the same domain, which we refer to as sibling prefixes suggest an operational or administrative correlation between the two address families, even when no structural linkage is visible in BGP alone. Sibling prefixes represent a cross-protocol relationship that is not captured in typical routing datasets unless explicitly analyzed.

We define and classify route deviation from established interdomain routing norms into following three distinct categories: (i) Hyper-specific Prefixes — IPv4 prefixes longer than /24 or IPv6 prefixes longer than /48 — which reflect atypical routing policies due to their specificity; (ii) Multiple Origin AS prefixes, which violate the common best practice of announcing single origin AS for an IP prefix, thereby exhibiting structural irregularities; and (iii) Sibling prefixes, defined as IPv4 and IPv6 prefix pairs that their IPs contain identical or closely related DNS records, indicating semantic alignment across IP versions. We refer to the aforementioned three category of routes as unexpected routes in this dissertation.

Moreover, we study RPKI operation, as one of the security elements of interdomain routing. We examine possible violations in forming ROA objects containing multiple prefixes, particularly its impact on RPKI synchronization delay. As the RPKI deployment grows [195], slowness and delay in RPKI synchronization might hamper the RPKI deployment. Therefore, looking beyond the prefixes and examining how the RPKI system operates is also necessary. A key element in RPKI synchronization is the time Relying Party (RP) software requires to download and validate the updated RPKI data [20, 36, 179]. We study the reasons for RPKI synchronization slowness, including the impact of using a single prefix (i.e., recommended) and multiple prefixes per ROA object, and propose techniques to reduce the RPKI synchronization delay.

Existence of unexpected routes contradict our assumptions of Internet routing, where all the networks follow the recommended best practices in IP prefix size, and single origin AS. This means that despite recommended guidelines [82, 165, 193, 261, 262, 270, 293], in practice, Internet routing is not as it is supposed to be, and we should expect the unexpected. Although these routes are not invalid, they challenge the assumption that networks consistently adhere to recommended routing practices, highlighting the need to study such cases better to understand the real-world routing of the Internet. Studying unexpected routes gives important insight into how some prefixes are structurally different or used differently in interdomain routing.

This dissertation aims to explore the prevalence of unexpected routes in the Internet routing ecosystem. We analyze the operational, security, and potential functional implications of unexpected routes in the Internet. We extract unexpected routes from several billion routes for over a decade to understand their evolution pattern. By examining various routing attributes of unexpected routes and the use of supplemental datasets, we infer their usage, lifespan, and potential functions in the Internet routing ecosystem. We perform active measurement by announcing a subset of unexpected routes via BGP to the real-world Internet to measure their propagation, and by setting the RPKI testbed to measure RPKI synchronization delay and examining the RPKI tree of five Regional Internet Registries (RIRs) to the ROA structure recommendation. By improving our understanding of the unexpected routes, our data-driven study helps the decision process and policy-making of Internet-governing organizations and network operators. The overarching research question in this study is:

Q: What is the prevalence of unexpected routes in Internet routing?

We leverage the routing data from well-known route collector projects of RIPE RIS [236], Isolario [144], and RouteViews [281] to provide a broad historical view of how unexpected routes evolved. We use large-scale DNS data from OpenIntel [204] to perform a longitudinal analysis of sibling prefixes and deepen our understanding of their usage

by network operators. To fully address the objectives of this dissertation, we refine the overarching research question into four sub-research questions, each focusing on a specific aspect of the problem. As network operators use their IP prefix and AS number in the BGP announcement, we begin investigating the deviation of routes from best practices and perform a thorough case study for each part as a separate case study under sub-research questions.

The BCP recommends that IPv4 prefixes longer than /24 and IPv6 prefixes longer than /48 are generally neither announced nor accepted on the Internet routing [82, 261, 262]. We examine the IP prefix part of the routes and identify if the announced prefix sizes violate the BCP in BGP route announcements. We focus on Hyper-specific Prefixes or prefixes smaller than /24 in IPv4 and /48 in IPv6 boundaries (i.e., /25 to /32 in IPv4 prefixes and /49 to /128 in IPv6 prefixes). As hyper-specific prefixes have too-specific prefix sizes, an IPv6 prefix of /48, as an example case, has the potential to be converted to several billions of hyper-specific prefixes. The announcement of an excess number of HSPs requires more memory and increases processing overhead for routers maintaining the BGP routing table, making it slow. Therefore, our first sub-question of research concentrates on hyper-specific prefixes as follows:

Q1: How prevalent are Hyper-specific prefixes in Internet routing?

To answer this question, we investigate the prevalence, usage, and propagation of HSPs in interdomain routing. We use a data-driven methodology by collecting over 11 years of routing data of route collectors from RIPE RIS and RouteViews projects as samples of the global routing table [236, 241]. In the global routing table, we observe several thousands of routes with CIDR sizes smaller than /24 for IPv4 (/48 for IPv6) or hyper-specific prefixes. We perform a longitudinal analysis of hyper-specific prefixes in the Internet routing ecosystem and analyze the number of ASes using those prefixes. We use supplemental datasets to understand the users of hyper-specific prefixes and examine CIDR sizes to infer their serving function in the interdomain routing. We expand our analysis by examining the lifespan of hyper-specific prefixes and their ROV state in the RPKI dataset. Moreover, we measure and analyze the HSP propagation in Internet routing by announcing it to BGP peers.

We then focus on the origin AS part of the routes announced by network operators. We study prefixes originated by more than one AS or MOAS as part of the unexpected routes in the Internet routing ecosystem. MOAS prefixes introduce ambiguity into ROV since the same prefix is announced by multiple ASes, making it unclear which origin is legitimate or intended. BGP prefix hijacks and configuration errors cause short-lived MOAS prefix conflicts, which are extensively studied by researchers [28, 57, 146, 247, 297]. We are particularly interested in long-lived MOAS prefixes, and this leads us to the second sub-question of our research:

Q2: What is the prevalence of long-lived MOAS prefixes in Internet routing?

One of the challenges to answering this question is that MOAS prefixes look the same as the BGP prefix hijack i.e., they both have a prefix with more than one Origin AS. Since BGP hijacks do not last long as network operators take action, we use time as a factor to separate the two to some extent [26, 58, 138, 203, 217, 256, 257, 276]. MOAS prefixes that exist for several days in the routing table, or long-lived MOAS prefixes, are understudied, and we have little information about them. Using the daily RIB snapshots

of route collectors, we perform a longitudinal analysis of long-lived MOAS prefixes on the Internet for over six years. Using supplemental datasets, we shed light on potential reasons why MOAS prefixes are used, who their users are, and how they propagate in interdomain routing.

For the third case, we focus on sibling prefixes or the cases where similar sets of DNS records are hosted on the IPv4 and IPv6 prefix pairs. Sibling prefixes are logically coupled; they serve the same application-level function, though separate routing prefixes in two address families. We formulate the third sub-research question as follows:

Q3: How prevalent are sibling prefixes in Internet routing?

To understand the relationship between IPv4 and IPv6 prefix pairs, we use the IPs for the DNS records that use both IP versions. Using large-scale DNS data, we examine whether a pair of IP prefixes, one IPv4 and one IPv6 prefix, are sibling prefixes. We analyze DNS records hosted on IPv4 and IPv6 addresses simultaneously, which we call dual-stack domains. Using the OpenINTEL dataset [204] for a period of four years, we extract all the IP addresses and the corresponding prefixes of dual-stack domains. By calculating the Jaccard similarity index [145] for IPv4 and IPv6 prefixes using large-scale DNS resolution data, we identify IPv4-IPv6 sibling prefixes. We measure the number of sibling prefixes on the Internet, their growth in the past 4 years, their ROV status in the RPKI data, and the business types of their origin ASes.

Lastly, we focus on the security aspect of interdomain routing. While RPKI is used to improve BGP security and its deployment is expanding, network operators have documented operational issues such as long delays for changes in RPKI to take effect [48–50]. We study RPKI's ROA structure compliance to recommended BCP and their corelation to the slowness of RPKI synchronization under the fourth research sub-question:

Q4: How recommended ROA structure impacts RPKI synchronization delay?

We expand our analysis by examining the ROA structure of RPKI tree across five RIRs, their influence on delay of RPKI synchronization, and propose mechanisms to expedite it. We analyze RPKI synchronization delay by examining the current structure of the RPKI ecosystem and measuring the delay in accessing RPKI resources from vantage points around the globe. There are three main components of the RPKI delay: (1) the *publication delay*, which is the time required for a change in RPKI data to be publicly visible in Publication Points, (2) the *synchronization delay*, which is the time Relying Party software will take to fetch, validate and update its list of Validated ROA Payloads, and (3) the *BGP refresh interval*, which is the time interval at which RPKI-enabled routers refresh their list of VRPs to perform Route Origin Validation and update their Forwarding Information Base [36]. We focus on the dynamics of fetching the RPKI tree from the PP hierarchy, examine the cryptographic verification of ROAs, and propose optimization techniques to enhance the RP synchronization delay.

1.1 Contributions

We study BGP prefixes that are different from what is (according to BCP) expected in interdomain routing and call them unexpected routes. Our study contribute to the inter-

domain routing by shedding light on hyper-specific prefixes, MOAS prefixes, and sibling prefixes, together with the RPKI ecosystem for fast synchronization. Our contributions can be summarized as follows:

1. The prevalence, propagation, and usage of hyper-specific prefixes
Network operators often filter out very small prefixes or HSP — any prefix size smaller than a /24 for IPv4 and /48 for IPv6 — because it is a best practice routing policy to limit accepted BGP routes to prevent routing table bloat and reduce the risk of accidental or malicious announcements. Therefore, HSPs are not supposed to appear in the global routing table of the Internet. We analyze large-scale BGP routing data for over eleven years with various supplemental datasets to deepen our understanding of HSPs in the real-world Internet routing ecosystem. In particular, we analyze their prevalence, use cases, and whether operators use them intentionally or accidentally. While their total number increases over time, most HSPs are visible for less than a day and have limited propagation in the interdomain routing. Nonetheless, some HSPs can be seen constantly throughout the year and propagate widely. Most HSPs represent (internal) routes to peering infrastructure or are related to address block relocations or blackholing. While hundreds of operators intentionally add HSPs to well-known routing databases, we observe that many HSPs are possibly accidentally leaked routes.

2. Methodology to detect Long-Lived MOAS prefixes and their users
We develop a robust methodology to separate long-lived MOAS prefixes from the prefixes that appear for a short period of time with multiple origin ASes. Using the Kneedle algorithm [243], we identify 30 consecutive days of visibility of a MOAS prefix as a suitable approach to classify the prefix as a long-lived MOAS prefix. We identify around 24k long-lived MOAS prefixes in IPv4 and 1.4k in IPv6. By analyzing the RPKI data, we find that more than 40% of MOAS prefixes have all origins registered correctly, with only a minority of MOAS having invalid origins. Moreover, the most prominent CIDR size of MOAS prefixes is /24 for IPv4 and /48 for IPv6, suggesting their use for fine-grained traffic steering and network resilience. We attribute a considerable number of MOAS prefixes to mergers and acquisitions of companies. Additionally, more than 90% of MOAS prefixes originate from two origin ASes, with the majority of detected origin AS relations being customer-provider. IT companies use most of the MOAS prefixes, and just 0.9% of IPv4 and 6.3% of IPv6 MOAS prefixes are used for anycast.

3. Longitudinal study and technique to detect sibling prefixes
We develop the concept of *sibling prefixes*, a cross-protocol relationship between the address families where two IP prefixes—one IPv4 prefix and its sibling IPv6 prefix — share the same or similar set of DNS records on their IPs. We calculate the Jaccard similarity index [145] on DNS data's fully qualified domain name records for IPv4 and IPv6 prefixes to identify sibling prefix pairs. Applying our technique to large-scale DNS resolution data, we identify 76k IPv4-IPv6 sibling prefixes. We find sibling prefixes to be relatively stable over time. For more than half of sibling prefix pairs, the origin AS number is identical for every pair, showing their usage within the same organization, and indicating how IPv6 is being adopted and deployed alongside IPv4. Moreover, we identify 24 hypergiants and Content Delivery Network (CDN)s using sibling prefixes, and 60% of all sibling prefixes have at least one of the prefixes with a valid ROV status in RPKI.

4. In-depth analysis of RPKI ROA structure and synchronization delay
We analyze RPKI ROA structure compliance to BCP and study causes of delay in RPKI synchronization using two popular Relying Party software. We examine key characteristics of the RPKI system, such as Route Origin Authorization structures, RPKI operational modes, certificate chains, and the networking delays to access PPs. Our findings reveal that bundling multiple prefixes into a single ROA reduces validation delays by up to threefold. Depending on the ROA structure and publication infrastructure of delegated Certificate Authorities (CAs), they can reduce RP synchronization delays or inflate them by up to 90%. North America and Europe observe delays of a few milliseconds, and other regions experience delays of several hundred milliseconds accessing RPKI resources. Every additional 100 milliseconds of Round Trip Time (RTT) can delay the RP synchronization process by up to 25 extra seconds. We propose the potential of expediting the RPKI synchronization by RP software optimization, ROA management, and Publication Point hosting on a robust and quick-to-access infrastructure.

5. Research recognition, impact, and community service
Our research on RPKI delay and ROA structure, the fourth research sub-question, received the best paper award from Network Traffic Measurement and Analysis (TMA) conference 2025 [2]. Our study on Long-lived MOAS prefixes, second research sub-question, received the fast-track paper award from TMA conference 2023 [3].

Given the challenges' timeliness and relevance to operational concerns, our findings were welcomed at major community forums for presentation. We presented our hyper-specific prefixes study at the RIPE 86 meeting [154], the Long-lived MOAS prefixes topic at RIPE's South East Europe (SEE) 12 meeting [253], the RPKI delay study at Mutually Agreed Norms for Routing Security (MANRS) routing security summit 2024 [1], and at the RIPE 88 meeting [251]. These venues enabled us to present our findings to a broad audience of network operators and researchers, ensuring the study's relevance to research and operational communities. To help the community, we published a blog post of our findings on hyper-specific prefixes at Asia Pacific Network Information Centre (APNIC) website [250]. Moreover, we developed a dashboard where we collect and publish HSPs daily since January 2021, available at *https://hyperspecifics.io*. Our dashboard is helpful for network operators to check whether their network's HSPs are leaked in the interdomain routing system and take action when necessary.

1.2 Overview and Structure

The structure of this dissertation is designed to address the research questions mentioned before and their associated challenges.

Chapter 2 provides the necessary background information to understand this dissertation. We cover the fundamentals of routing topics such as Routing, AS, BGP, and routing security topics like RPKI.

Chapter 3 focuses on the first set of unexpected routes and thorough analysis of hyper-specific prefixes in the Internet routing ecosystem. We perform a longitudinal study together with active measurement to understand the reasons for hyper-specific prefixes existence and propose suggestions for handling them.

Chapter 4 focuses on the second set of unexpected routes, long-lived MOAS prefixes, in the Internet routing ecosystem. A thorough investigation of the causes of MOAS prefixes and a deep dive into their origin, business types, users, and usage by network operators are also covered in this chapter.

Chapter 5 examines the similarity and corelation between IPv4 and IPv6 prefixes to identify sibling prefixes. It further expands the analysis to a longitudinal study, usage of sibling prefixes by the same or different organizations, their business types, and the ROV state of sibling prefixes in the RPKI dataset.

Chapter 6 explains the reasons for slow RPKI synchronization and ROA structure. We perform an active measurement of RPKI synchronization using two Relying Party software and in different modes of operation. This chapter details the network delay's role in RPKI synchronization and identifies each step and element in RPKI synchronization that is causing the delay.

Chapter 7 concludes this dissertation by providing a summary of the findings and impact of this study and proposes future work.

Chapter 2
Background

This chapter provides the necessary background information to help understand the rest of the thesis. We explain the Internet protocol in §2.1, followed by explaining the Autonomous System concept as the basic elements of Internet routing. We then explain BGP in §2.2 as a standard interdomain routing protocol between ASes in the Internet along with BGP policies and BGP Prefix Hijack and the business relationship between ASes. We extend the background section by explaining the key players in Internet routing structure in §2.3, explaining Internet service providers, content and cloud providers, and IXPs. We use various BGP data during our analysis and §2.4 provides highlights main list of these data sets, others are listed for each analysis in their corresponding chapter. Finally, we provide a detail information on the RPKI building blocks and operation in §2.5.

2.1 Internet Protocol

Internet Protocol is a network layer or layer three protocol in the Open Systems Interconnection (OSI) model that enables devices to identify and communicate with each other in the network [31, 159]. End users receive their IP addresses from Internet Service Providers. ISPs obtain IP addresses from the Local Internet Registries (LIRs) or National Internet Registries (NIRs), or from Regional Internet Registries that have the IP address blocks assigned by Internet Assigned Numbers Authority (IANA) [124, 137]. Devices connected to the network encapsulate their data in datagrams, each with an IP header specifying source and destination IP addresses, and use IP addresses as logical identifiers to communicate. There exist two versions of IP version 4 (IPv4) and IP version 6 (IPv6).

Since the standardization of IPv6 in 1998 [74], both versions of the Internet Protocol have coexisted on the Internet. IPv4 was widely adopted from the beginning of the Internet in 1983 by network operators, and it has been almost exhausted [137]. The IPv4 address consists of 32 bits in length, shown as four octets of 4 octets in "dotted decimal" notation (for example, 192.0.2.53). The deployment of IPv6 began in 1999 [137]. Its adoption has been growing over the past years with more and more networks deploying IPv6 devices and services [13, 106]. IPv6, which is supposed to help the IPv4 address shortage,

has a larger address length of 128 bits, shown as colon-separated hexadecimal strings (for example, 2001:0db8:582:ae33::29) [75, 116]. Every IP address, IPv4 or IPv6, has two parts: the host part bits and the network part bits. The network part is used for routing purposes (except routes that are directly used for a /32 IPv4 or /128 IPv6 address) and is usually represented with a slash notation like /24 or /48.

2.1.1 Special Purpose IPs

In IPv4, a group of address space is reserved for special purposes such as address allocation for private Internets [222], IPv4 address blocks reserved for documentation [15], the Internet address assignment and aggregation plan [93]. These prefixes are supposed to serve dedicated purposes and are not in use by network operators for routine communication on the Internet. Some examples of these IP ranges are any network starting with *10.0.0.0/8*, *172.16.0.0/12* and *192.168.0.0/16*, which are reserved for private IP addresses that are then convertible to public IP addresses using Network Address Translation (NAT) techniques to enable them to use the Internet. Similar to IPv4, ranges of IPv6 address space is also reserved for special-use IPv6 addresses [27] like *::1/128* is the loopback address [116] *::/128* is the unspecified address [116], textit::FFFF:0:0/96 are the IPv4-mapped addresses [116], *::ipv4-address/96* are the IPv4-compatible addresses [116], and *fe80::/10* are the link-local unicast [116] addresses, *fc00::/7* are the unique-local addresses [118], *2001:db8::/32* used for documentation [127], that should not appear in the public Internet. There is also other special purpose IP address prefixes in IPv6 such as the 6to4 addresses [51], Teredo addresses [126], 6bone experimental network [88, 117, 119], and Overlay Routable Cryptographic Hash IDentifiers (ORCHID) addresses [201]. Similarly, for global anycast, the *ff00::/8* multicast addresses [116], or *fe80::/10* to be used in link-scoped unicast addresses [116], and as listed in Request for Comments (RFC) 5156 [27] other group of IPv6 addresses also require specific considerations.

2.1.2 Autonomous System

An Autonmous System (AS) is one or more network operators that announce an IP prefix or group of IP prefixes to the Internet that presents a common routing policy. To announce the reachability information of their prefixes and exchange routing information with other networks, network operators need a ASN. Unique Autonmous System Numbers (ASNs) are assigned in blocks by IANA to RIRs. Organizations can request ASN directly from RIRs [114]. Upon successful completion of the application, the RIRs then can assign the ASNs to the network operators in their designated region [11, 229]. Figure 2.1 illustrates the geographical service regions of the five Regional Internet Registries as defined by IANA. While early in 2007, all ASNs were 2-byte that could provide up to 65536 unique ASNs. Today, the ASNs are presented in either a 2-byte (16-bit) or 4-byte (32-bit) format [128, 286]. A range of these ASNs are reserved for private use by IANA [192]. As of 2025, nearly 120k ASNs are allocated to network operators [181].

The ASes can operate as stub, multi-homed or transit AS in the Internet. An AS that connects to only one other AS is referred to as a stub AS. If it connects to more than one AS for its Internet connection, it is called a multi-home AS. A third category is a transit AS that links two or more ASes, allowing them to exchange data.

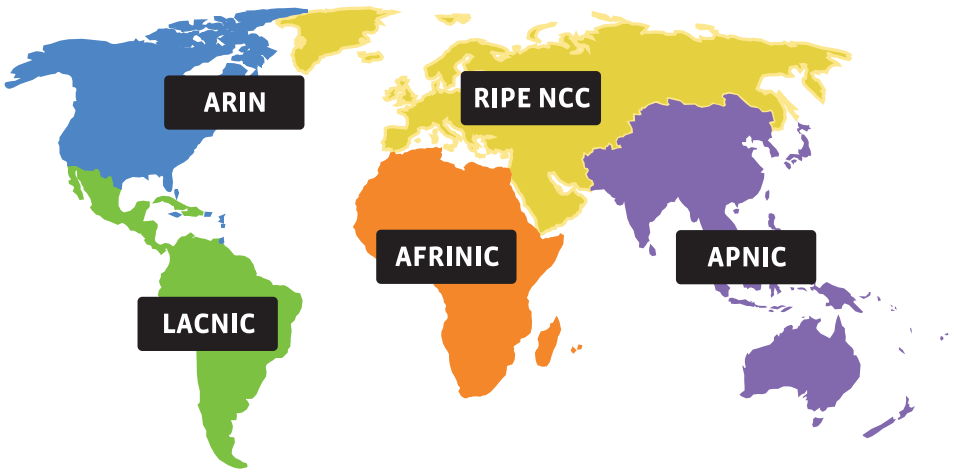

Figure 2.1: Service regions of the five Regional Internet Registries as defined by IANA. Image source: [139].

2.2 Border Gateway Protocol

Border Gateway Protocol is the de facto routing protocol used between ASes to exchange routing information. BGP is a path-vector routing protocol that makes routing decisions based on network paths or routing policies [266]. BGP uses the AS PATH attribute of the route, which contains a sequence of ASes a route advertisement has traversed, for routing policy decision and routing loop prevention. BGP can be configured as internal Border Gateway Protocol (iBGP) between network devices inside an AS, or as an external Border Gateway Protocol (eBGP) session with other ASes. BGP uses Transmission Control Protocol (TCP) port 179 for reliable exchange of routing information [86, 289].

BGP first performs a full exchange of routing information with its peers to converge. Once the initial routing table is built, BGP switches to incremental updates, advertising only changes such as newly available routes or withdrawn paths. BGP uses two routing databases of RIB and Forwarding Information Base (FIB). The RIB database of BGP contains routes received from neighbors, selected best route based on BGP decision, and routes selected to be sent to peers [221]. BGP uses FIB to forward packets. The FIB is populated from best routes in the RIB. BGP-speaking routers can accept or reject the routes based on their routing policies.

2.2.1 BGP Policies

Each AS sets BGP policies locally and can be asymmetric for inbound and outbound traffic. Network operators can use BGP attributes, such as local preference, AS path length, Multi-Exit Discriminator (MED), and communities to apply routing policies. The local preference is used internally to prefer a particular path over others for a packet. BGP uses the shortest AS path to prefer a path over others, or the AS Path prepending can make a path look longer and less attractive for traffic. The MED values signal the peer AS of which entry point is preferred. The BGP communities are metadata tags that network operators can use for many services, like tagging of customers, peers, geographically

originating routes, and no export [54, 187].

2.2.2 BGP Prefix Hijack

BGP prefix hijack happens when a misconfigured or malicious network announces IP prefixes for which it is not the owner causing diverting of the victim network's traffic through it - potentially for traffic interception, blocking, or surveillance. The AS7007 incident in 1997 was one of the first BGP routing incidents where a software bug caused a router to convert and announce a large part of the Internet routing table as more specific or /24 IP prefixes [67]. The reachability issue of Cloudflare's DNS [115] in 2024, Orange Spain's Internet outage [166], loss of approximately two million dollars in cryptocurrency for South Korean hosting company in 2022 [180], Facebook outage in 2021 [91], to name a few, are some examples of BGP prefix hijacks. While BGP routing incidents might happen intentionally or accidentally, they can impact the legitimate flow of traffic causing slow Internet and security risks such as packet interception or manipulation [180]. Unfortunately, IP prefix hijacks happen relatively frequently [24, 38] and are subsequently studied extensively by researchers [26, 58, 138, 203, 217, 256, 257, 276].

2.2.3 AS Business Relationship

BGP routers follow the valley-free routing model to conform to the business agreements and ensure that each AS in the AS path acts in its interest and does not provide free transit [94]. Typically, large ISPs (e.g., Tier-1 networks) peer with each other and have no provider. Smaller networks pay for their Internet connection to transit network operators and peer with other ASes to reduce costs. In BGP, the economic relationship, not just technical metrics, determines who advertises what to whom. The three main relationship types are as follows:

- Customer to Provider: The customer pays the provider for Internet access, hence the provider accepts and advertises customer routes to the Internet.
- Peer to Peer: Two ASes exchange their own, and their customers', traffic for free. To prevent free-riding, peers do not exchange routes of their providers or other peers.
- Provider to Customer: Customer accepts all provider routes but does not advertise them further. They charge the customer for the Internet connection.

2.3 Internet Routing Structure

The Internet's routing infrastructure is primarily built from the Interconnection of Autonomous Systems or independently administered networks using BGP to exchange routing information.

2.3.1 Internet Service Providers

Internet Service Providers facilitate the interconnection of networks to the rest of the Internet. ISPs manage one or more ASes having unique Autonomous System Numbers

assigned by Regional Internet Registries like ARIN, RIPE NCC, AFRINIC or APNIC. Based on their coverage and peering relationship, the ISPs can be classified into a tiered hierarchy as Tier 1, Tier 2, and Tier 3 ISPs [100, 112, 215, 288].

- **Tier 1 ISPs** are large backbone networks that can reach other networks without paying transit fees. Tier 1 ISPs form the core of the Internet's interconnection by establishing settlement-free peering with each other. AT&T, NTT, Deutsche Telekom Global Carrier, and Orange are examples of Tier 1 ISPs.
- **Tier 2 ISPs** are the second level in the ISP tiered hierarchy that purchase transit from Tier 1 networks. Tier 2 ISPs also establish peering with other networks primarily for cost-saving and performance optimization.
- **Tier 3 ISPs** are usually smaller organizations that provide internet services to end users and often rely on upstream for transit.

To exchange traffic with multiple peers more efficiently, ISPs also interact with IXPs. IXPs can provide an easy BGP peering facility, reducing the latency and cost of transit [147, 233].

2.3.2 Content and Cloud Providers

Content providers generate and provide digital information and services in various formats such as video, music, news, software, and websites. Netflix, YouTube, Meta (Facebook), and Spotify are examples of content providers. Typically, they operate large-scale CDNs to replicate content across globally distributed servers to reduce latency and improve user experience. Some content providers also deploy their services directly within the ISP or IXP [47, 98, 162]. These operators also deploy and operate their own backbone network, and some of them, like Google and Meta, also deploy private fiber infrastructure [59, 161].

Cloud providers offer computing resources and services, including compute, storage, networking, and application deployment platforms. Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP) are examples of cloud providers. Moreover, cloud providers operate large data centers around the world, and provide Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) [108, 171, 218]. Like content providers, cloud providers also participate in IXPs and establish BGP peering directly with other networks to reduce latency and improve user experience. Cloud and content providers are converging, and many companies (eg, Google, Amazon) are both cloud and content providers [76, 161].

2.3.3 Internet Exchange Points

IXPs are physical network infrastructures in a location where multiple networks, such as ISPs, content providers, and other network entities, interconnect to exchange traffic directly. IXPs provide an easy BGP peering facility for network operators without paying a third-party transit provider, reducing the latency and improving end-user experience. IXPs mostly use a Layer 2 switch and do not perform routing themselves. The connected participants do routing via BGP [7, 147, 233].

2.4 BGP Data

Our study relies on multiple data sources at varying granularities between the 1st January 2010, and the 15th January 2021. In this section, we briefly describe each data source. We provide details on how we preprocess and aggregate these dataset and the duration and specific purpose in the the analysis section.

2.4.1 Route Collectors

Route Collectors (RCs) are dedicated devices that peer with many networks and dump the routing information they receive from them. While we later want to utilize *"snapshots"* of the passively collected routing information from the RC projects Isolario [144], RIPE RIS [236], and Routeviews [241] for our analysis.

2.4.2 AS Relationships Dataset

The business relationships between ASes can have a significant impact on their routing. Center for Applied Internet Data Analysis (CAIDA) provides monthly estimates of AS relationships dataset for those business relationships based on the AS Rank algorithm [175]. We use CAIDA's AS relationship dataset to analyze inter-AS relationships of origin ASes associated with non-standard routes in our study. This dataset is released by CAIDA every three months. The dataset classifies the inter-AS relationships into "customer-provider or provider-customer", "peer to peer", or "unknown" using the method described by Luckie et al. [175].

2.4.3 AS to Organizations Mapping Dataset

We use the inferred AS to organization mapping dataset provided by CAIDA [39] for the duration of our study. The dataset is created by querying WHOIS databases from the five Regional Internet Registries (ARIN, LACNIC, RIPE, AFRINIC, and APNIC), and two National Internet Registries (KRNIC and JPNIC) every quarter, and then applying CAIDA's AS to organization technique to map each AS to an organization [40]. Due to the lack of historical data, we are unable to use two recently presented improvements [17, 56] to the CAIDA AS to organization dataset.

2.4.4 AS Business Type Dataset

To gather detailed information about the type of organization using non-standard routes, we use the Autonomous System Database (ASdb) dataset [298]. ASdb provides a mapping of each AS into 17 industry categories and 95 sub-categories using data from established business intelligence databases and machine learning. We use ASdb to identify AS organization types and business relationships between the origin ASes of non-standard routes.

2.4.5 Internet Routing Registries

The Internet Routing Registry (IRR) is a distributed database where network operators submit and share their routing information. In particular, network operators created

route objects that describe which IP prefixes are announced by which AS, and the maintainer information [188]. IRR objects can be used by other ASes to develop routing plans, apply glsacl, and permit or deny their traffic. The five RIRs and a few other network operators maintain the distributed database for IRR objects [143]. However, the outdated and conflicting IRR data, together with security issues due to a lack of authentication, are problems of this database [14, 153, 174, 197, 235, 264].

2.4.6 WHOIS Database

WHOIS is a protocol operating on TCP port 43 that allows users to query information about the ownership of Internet services, including domain names, IP prefixes, and autonomous system numbers. WHOIS is offered as a public service, which helps identify who is responsible for a given resource and how to contact them. Different entities manage the WHOIS database, the IP WHOIS maintained by RIRs like ARIN, RIPE NCC, APNIC. The WHOIS database for domains is maintained by registrars and registries (e.g., Verisign, GoDaddy). Despite the usefulness of WHOIS, it has limitations, such as being a text-based format, having outdated or inaccurate data, and privacy concerns about revealing personal data. The Internet community is gradually transitioning to a newer protocol called Registration Data Access Protocol (RDAP). RDAP provides structured, secure, and extensible data access in a machine-readable JSON format [123]

2.5 Resource Public Key Infrastructure

The Resource Public Key Infrastructure [168] is a security system that uses a chain of X.509 certificates to validate the authenticity of routing announcements made by ASes on the Internet. RPKI aims to prevent malicious or accidental routing incidents due to prefix mis-origination, i.e., when an attacker falsely advertises itself as a specific IP address block holder, which can result in redirecting or blackholing Internet traffic intended to the legitimate holder. RPKI provides a cryptographically verifiable mechanism for routers to make informed decisions on whether to accept or drop incoming routing announcements. However, RPKI does not mitigate all BGP attacks (e.g., path injection). Figure 2.2 presents an overview of the various building blocks of the RPKI system that we explain through the action statements below.

Certificates: RPKI follows the IANA prefix allocation hierarchy. Each RIR operates a self-signed certificate, called a Trust Anchor, for the Internet resources they manage.

When RIRs allocate resources to Local Internet Registries e.g., an Internet Service Provider or to National Internet Registries like TWNIC or NIC.br, the TA issues a child certificate with the member resources [Action ❶]. This certificate (and the associated private key) is used to create special attestations called ROAs (Route Origin Authorization).

Modes of operation: NIRs or LIRs that have been issued resource certificates can choose between two modes of operations: (1) The *Hosted mode*, where their RPKI environment (RPKI objects, private key material, and PP) is managed entirely by the RIR. This is currently the most used mode of operation, or (2) the *Delegated mode*, where an LIR operates its own RPKI environment including the management of a PP. Some RIRs provide Publication as a Service (PaaS), where delegated CAs can publish their objects in their RIR's PP.

Route Origin Authorization: Once the members are issued a certificate that proves their

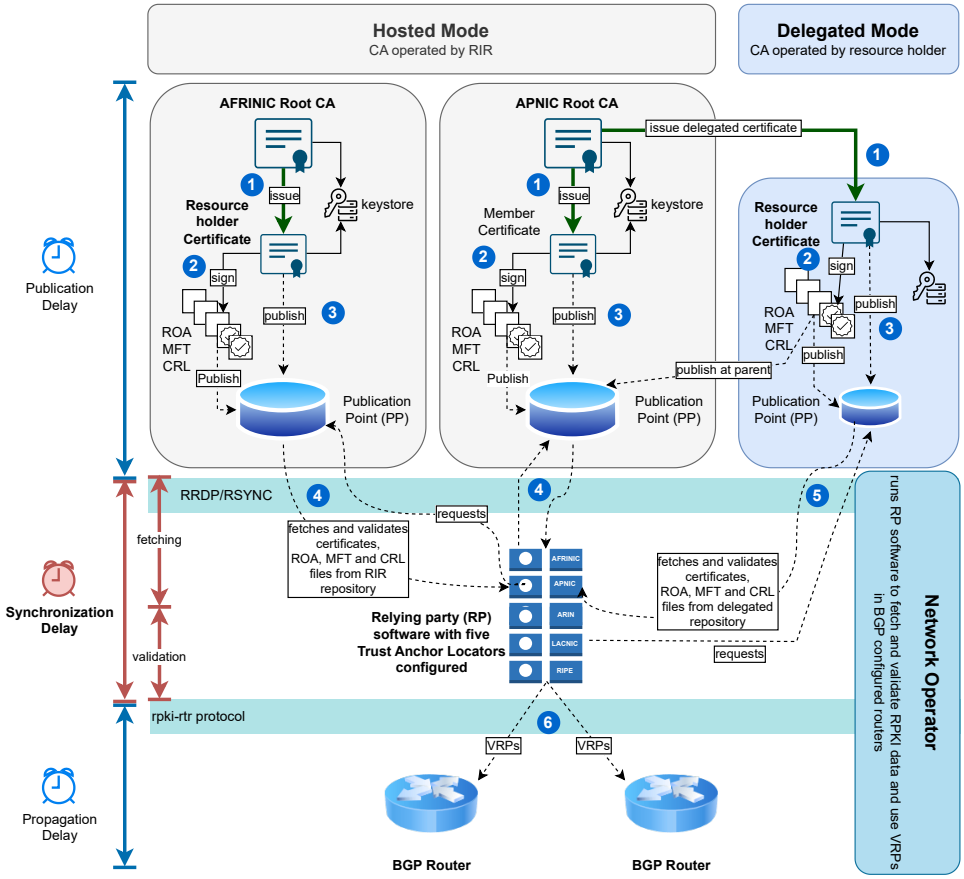

Figure 2.2: Components of the Resource Public Key Infrastructure and their modes of operation.

holdership of allocated resources, they can issue a signed attestation called a ROA to authorize an AS to advertise their prefixes [Action ❷]. The ROAs are then published in repositories[1] called Publication Points, which are operated by either the RIR or by the resource holder itself in the case of the delegated mode [Action ❸].

Publication Points: Each Certificate Authority in the RPKI hierarchy needs to publish their objects (certificates, ROAs, manifest files, and Certificate Revocation Lists (CRLs)) in a publicly accessible repository, called a Publication Point. Anyone can operate a PP, but in practice, each RIR operates a PP to publish the RPKI objects of their own CA and those of the child CAs (if the latter is operating in 'hosted mode'). These objects are then fetched using either the RRDP [33] or rsync protocol and validated by Relying Party software. RP software is used to validate the RPKI data and also produce a list of Validated ROA Payloads or VRPs.

Route Origin Validation: [Action ❹] RPs periodically fetch and validate the RPKI objects from the PPs, and they generate a set of VRPs. VRPs contain a set of prefixes (with prefix and max length) and the associated authorized ASes. Validation follows a top-down approach, starting with the TAs. RP software usually comes bundled with

[1]We use the terms *repository* and *Publication Point* interchangeably.

the RIR's Trust Anchor Locators (TALs), where each TAL contains the root certificate's Uniform Resource Identifier (URI). In the case of delegated mode, the parent CA will point to the child CA's repository to build the chain of trust, allowing the RPs to fetch and validate the RPKI objects from the delegated CAs' PPs [Action ❺]. Using *rpki-rtr* protocol, the list of VRPs is periodically pushed to the RPKI-enabled routers [35]. Using this information, incoming routing announcements in the Routing Information Base are tagged as either Valid, Invalid or Unknown. Based on the routing policy, routers will either accept or drop incoming routing announcements before sending them to their FIB [Action ❻].

RPKI Dataset: The RPKI contains legally binding mappings between Internet resources and ASes. The Route Origin Authorization records contained in the RPKI database can be utilized to validate the legitimate announcements of prefixes by ASes. We use archived RPKI records generated by Chung et al. [60] and from Réseaux IP Européens Network Coordination Centre (RIPE NCC) [234] in a monthly granularity for the duration of our study. With the RPKI dataset we identify the route ROV state of MOAS prefixes.

Chapter 3
Hyper-specific Prefixes

This chapter explores the Hyper-specific prefixes phenomenon in the Internet's routing ecosystem, focusing on routes that violate commonly accepted prefix length boundaries. We begin by analyzing how such prefixes appear in BGP announcements and examining their potential causes—whether they result from operational practices, accidental internal leaks, or indicators of malicious behavior such as prefix hijacking. The IPv4 shortage and acceleration in IPv6 adoption, can cause the use of HSPs to grow, raising concerns about routing table scalability and the effectiveness of current policy enforcement mechanisms.

Autonomous Systems use the Border Gateway Protocol to announce prefixes to their peers [221]. Each BGP-speaking router of an AS can decide to accept or reject incoming announcements based on the prefix itself, the AS path, or other attributes that are attached to a route (e.g., BGP community values). Due to this concept, every single AS (and, in fact, also all its routers) may have a unique viewpoint into the Internet's routing ecosystem [239].

Many popular BGP guidelines recommend the rigorous filtering of prefixes that encompass only a few addresses [81, 82, 110, 183, 202, 261, 262] and, hence, those prefixes have been shown to propagate neither far nor reliably [273]. While the possible reasons for announcing these types of prefixes are broad and range from traffic engineering over multi-homing configurations to prefix-hijack prevention [63, 130], the boundary for announcements which are deemed "widely acceptable" are usually considered to be a /24 prefix in IPv4 and a /48 prefix in IPv6.

We begin our analysis of unexpected routes by performing an in-depth analysis of prefixes that are more specific than those boundaries, i.e., /25 to /32 IPv4 prefixes and /49 to /128 IPv6 prefixes. We refer to those prefixes as **hyper-specific prefixes** (HSPs, see §3.1.3 for more details) and analyze their prominence in the global routing ecosystem, the functions that they serve, and whether they represent intentional or accidental announcements. More specifically, we make the following main contributions in this chapter:

Observability. We perform a decade long analysis of HSPs as seen by 67 route collectors (see §3.2). We find that the number of HSPs has increased substantially since 2010 and peaked in 2018 at around 115K IPv4 and 18K IPv6 prefixes. While we observe that especially HSPs which are announced consistently for an entire year are visible by

hundreds of collector peers, the average HSP can only be seen by a handful of them.

Use Cases & Functions. We analyze potential use cases of HSPs by combining insights from different analyses of CIDR sizes, BGP communities, and service hit rates across multiple years (see §3.3). We find that IPv4 HSPs mostly represent (internal) routes towards peering subnets and blackholing, whereas IPv6 HSPs are mainly used for address block relocations and, in substantially fewer cases, blackholing. We further find that HSPs are unlikely to contain many end hosts and that they are rarely used for traffic engineering.

Intended or Accidental Use. We compare the HSPs visible in BGP with those that were explicitly entered into routing databases—in particular, the Internet Routing Registries and Resource Public Key Infrastructure—to investigate intended or accidental use of HSPs (see §3.4). We find that while thousands of ASes explicitly specify their intent to use HSPs, many HSPs likely represent accidentally leaked routes.

The Future of HSPs. We discuss how the research and operator communities could make use of HSPs in the future. Finally, we publish a dashboard providing up-to-date HSP statistics to help AS operators in detecting leaked internal routes at `https://hyperspecifics.io`.

3.1 Datasets

Our study relies on multiple data sources at varying granularities between the 1st January 2010, and the 1st May 2021. In this section, we briefly describe each data source as well as how we preprocess and aggregate it.

Route Collectors are dedicated devices that peer with many networks and dump the routing information they receive from them. While we later want to utilize *"snapshots"* of the passively collected routing information from the RC projects Isolario [144], RIPE RIS [236], and Routeviews [241] for our analysis, it is unclear how a given observation window size w translates into snapshot consistency. To estimate this translation function, we initially retrieve data for all days in 2010, 2013, 2016, and 2020. For each day, we download the first Routing Information Base snapshot as well as all available update messages produced by each RC. If an update file is missing, we, additionally, download the first available RIB snapshot after the missing update file. After extracting the HSPs for each day, we analyze consistency as the fraction of HSPs seen at day $n + w + 1$ that are also visible within the observation period $[n, n + w]$. Notably, we try all possible window size positions, i.e., $n \in \{0, ..., d - w - 1\}$ where d is the number of days in the given year. Figure 3.1 shows the mean as well as the interquartile range (IQR) across all possible n for window size w between 1 and 60 days for IPv4 and IPv6 HSPs in 2020. We observe that a seven-day window allows us to achieve a consistency of 97 % and 98 % for IPv4 and IPv6, respectively. Notably, further expanding the window size to 60 days would only increase the consistency by ~0.5 %. Given that we now have a snapshot aggregation window, we still need to pick a snapshot interval. When comparing the number of visible HSPs for different snapshot intervals, we observe that a three-month interval provides an optimal balance: While the number of data points is still capable of capturing all visible trends in more-frequent snapshot intervals, the reduced amount of data (i.e., only seven days every three months) still allows us to perform computationally expensive observations for the entire decade promptly.

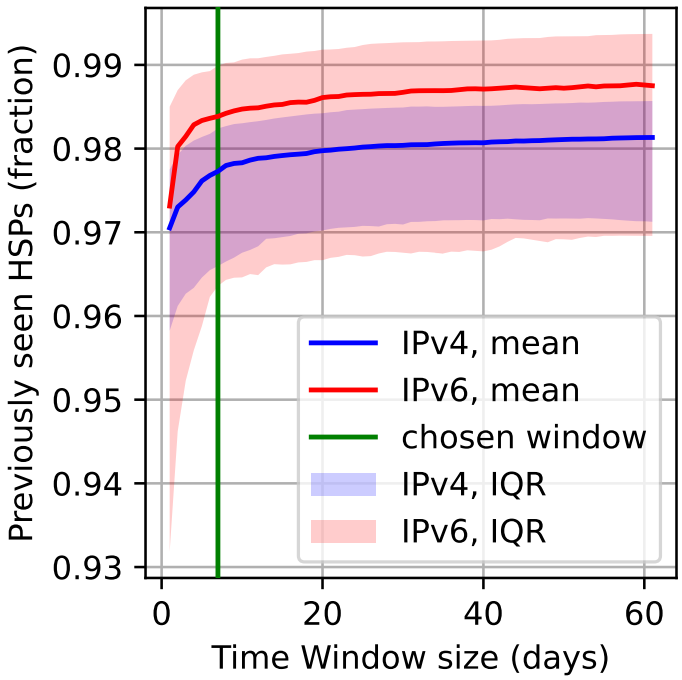

Figure 3.1: Impact of window size on visible HSPs.

3.1.1 Filtering Pipeline

When an AS peers with a Route Collector, the router that feeds the collector may provide all routes that are not removed during (or before) egress filtering. Hence, misconfigured egress filters can lead to misinterpretations. For our analysis, we filter out HSPs which are originated by feeder AS directly connected to a route collector. However, we use the HSP if it has been propagated to at least 2 AS hops, including feeder AS. In addition, we filter all private, reserved, multicast, and experimental IP prefixes. Furthermore, we also filter prefixes originated by a private AS. Finally, we remove the HSPs we identify as outliers during the data cleaning process. §3.1.2 provides detail information on HSPs we have filtered out.

Manual Data Isolation. When looking at the remaining raw routing data, we still find a very noisy data set with high variance between successive snapshots, yet most of this variation is contributed by very few ASes. To prevent the bias that those ASes introduce to affect our analysis, we manually filter our route collector data based on biased vantage points (i.e., RC peers) and Origin ASes. We calculate the change of the number of unique HSPs, unique HSP routes, and unique anchor prefixes between successive snapshots. Based on those metrics, we manually isolate routes from our data set that were fed by vantage points (or originated by ASes) that showed a substantial contribution of HSP routes for a very limited amount of time. E.g., for our snapshot in July 2016, AS 35908 feeds IPv4 HSPs within 21455 anchor prefixes despite usually feeding HSPs for mostly single-digit numbers of anchor prefixes. Table 3.1 in the appendix summarizes from when to when we isolate which routes as well as a reason for our choice. Note that those data points are already isolated before computing Figure 3.1.

Table 3.1: Applied filtering and isolation rules.

Timeframe	Filter name	Filter Details	Reason
entire period	Private Origin ASes 2 Bytes	Origin AS number from 64512 to 65534	private IPv4 ranges.
entire period	Private Origin ASes 4 Bytes	Origin AS number from 4200000000 to 4294967294	private IPv4 ranges.
entire period	Private IPs	IPv4 Private IP ranges	private IPv4 ranges.
entire period	Class D and E	IPv4 Prefixes > 223.x.x.x	IPv4 multicast and class E IP ranges.
entire period	Abnormal Prefixes	for IPv4 prefix > /32	abnormal IPv4 prefixes.
		for IPv6 prefix > /128	abnormal IPv6 prefixes
entire period	No Origin	Routes having no origin AS	AS-internal routes.
	Internal	Feeder AS is the Origin AS	
2015/10/01-07	IPv4 Noisy Origins	Origin AS == 9498	origin AS.⋆
2016/10/01-07	IPv4 Noisy Origins	Origin AS == 36937	origin AS.⋆
2017/04/01-07	IPv4 Noisy Origins	Origin AS == 9498	origin AS.⋆
2019/07/01-07	IPv4 Noisy Origins	Origin AS 7122	origin AS.⋆
entire period	IPv4 Noisy Origins	Origin AS 12400	origin AS.⋆
2016/07/01-07	IPv4 Noisy Peer AS	Peer AS 35908	peer AS.⋆
2017/01/01-07	IPv4 Noisy Peer AS	Peer AS 60924 and 27630	peer AS.⋆
2017/10/01-07	IPv4 Noisy Peer AS	Peer AS 37497	peer AS.⋆
2018/10/01-07	IPv4 Noisy Peer AS	Peer AS 14361	routes from particular peer AS⋆
2019/01/01-07	IPv4 Noisy Peer AS	Peer AS 262757	peer AS.⋆
2020/04/01-07	IPv4 Noisy Peer AS	Peer AS 268430	peer AS.⋆
2021/04-07/01-07	IPv4 Noisy Peer AS	Peer AS 398465	peer AS.⋆
2021/01-10/01-07	IPv4 Noisy Peer AS	Peer AS 203125	peer AS.⋆
2020/04-07/01-07	IPv4 Noisy Peer AS	Peer AS 268430	peer AS.⋆
entire period	IPv6 Noisy Origins	Origin AS 4761	origin AS.⋆
2017/07/01-07	IPv6 Noisy Origins	Origin AS 17451 and 45899	origin AS.⋆
2019/04/01-07	IPv6 Noisy Origins	Origin AS 7713	origin AS.⋆
2021/07/01-07	IPv6 Noisy Origins	Origin AS 8100	origin AS.⋆
2018/07/01-07	IPv6 Noisy Peer AS	Peer AS 199036	peer AS.⋆

3.1.2 Applied Data Isolation Rules

We have excluded a set of HSPs originated by ASes as illustrated in table 3.1. ⋆These ASes contributed/announced either (1) an extraordinary high number of HSPs (i.e., 100 or more times higher than in other snapshots) or (2) HSPs in an extraordinary high number of anchor prefixes for a limited time.

3.1.3 MSPs vs. HSPs

In this section, we want to briefly contrast the definitions of More-Specific Prefixes (MSPs) and Hyper-Specific Prefixes. P is an MSP of P' when the address space that P describes is entirely contained in P', e.g., 1.0.0.0/24 is an MSP of 1.0.0.0/22. In contrast, we call a prefix hyper-specific if its CIDR size is larger than /24 or /48 for IPv4 and IPv6, respectively. While labelling a prefix as an MSP requires another (covering) prefix, the HSP label relies entirely on the CIDR size of a given prefix and does not require a second, related prefix. Notably, many—but not all—hyper-specific prefixes are also MSPs of less-specific prefixes. As the definitions of MSPs and HSPs are very different, further classifications of HSPs (as in, e.g., Geoff Huston's blogpost [130]) are not directly applicable to HSPs.

3.1.4 Orthogonal Datasets

RPKI Snapshots. The Resource Public Key Infrastructure contains legally binding mappings between Internet resources and ASes. The Route Origin Authorization records contained in the RPKI database can be utilized to validate the legitimate announcements of prefixes by ASes. Rather than compiling the RPKI database ourselves (e.g., using Routinator), we utilize the daily snapshots generated by Chung et al. [60]. The snapshots that we use (from 1st April 2015, until 1st April 2021) are publicly available [231].

IRR Snapshots. The Internet Routing Registries are databases containing information about routing policies formatted in the Routing Policy Specification Language (RPSL) language. We utilize private, three-monthly snapshots of all databases listed at [142] between 1st January 2018, and 1st May 2021.

AS Relationships Inferences. The business relationships between ASes can have a significant impact on their routing. CAIDA's AS relationships data set provides monthly estimates for those business relationships based on the AS Rank algorithm [175]. We used the first available IPv4 and IPv6 relationship file within every three-month interval between 1st January 2010, and 1st April 2021 [43].

AS Classification Inferences. Similar to business relationships, the business class of an AS can affect its routing. Yet, there is currently no data set capable of (close to) perfectly mapping ASes to classes. We utilize CAIDA's AS classification data set as a best-effort estimator for business classes. While CAIDA currently works on improving the algorithm, restricted access to the inferences between 1st November 2018, and 1st May 2021 is possible [42]. We overwrite the classes for publicly known Tier 1 ASes [68][2] and Hypergiants [30]. Further, if an AS from the "Transit/Access" class has at least one customer AS (based on the "AS Relationships Inferences" data) we re-classify it as "Transit", otherwise we re-classify it as "Access."

[2]The list of Tier 1 ASes [68] (1) closely resembles the Tier 1 clique inferred by CAIDA's business relationship inference algorithm throughout 2020, and (2) barely ever changed throughout the last decade.

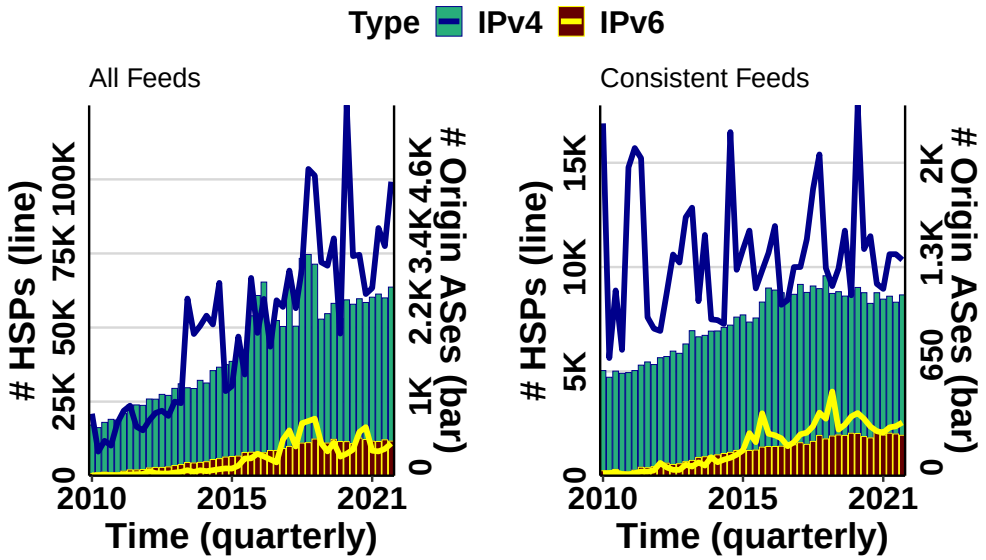

Figure 3.2: Growth of HSPs and HSP origin ASes as visible in all feeder ASes (left) and a consistent set of feeder ASes (right).

3.2 Observability

We begin our exploration of hyper-specific prefixes by analyzing their current and past presence in the Internet's routing ecosystem.

In particular, we examine the routing information from hundreds of globally distributed ASes—called "feeder ASes" or "route collector peers"—collected by the Isolario [144], RIPE RIS [236], and Routeviews [241] projects. Starting from January 2010, we generate snapshots consisting of a week of RIB and update files every three months until October 2021. We provide further details about the choice of this window size in §3.1. We employ various filtering steps to sanitize the data from, e.g., announcements of unallocated Internet resources, certain noisy origin ASes[3], or temporarily misconfigured feeder ASes. We also reached out to operators of noisy origin ASes. Two of these operators were not aware of this problem, but addressed it quickly upon our notification. A comprehensive list with justifications for the individual steps can be found in §3.1.1.

First, we investigate the evolution of HSPs from January 2010 to October 2021. Figure 3.2 shows the number of hyper-specific prefixes (lines) and ASes that originate them (bars) over time. Looking at the left sub-plot, we observe that the number of seen HSPs (despite being noisy) consistently increases throughout the eleven years. We see more than 10k IPv6 and 100k IPv4 HSPs by the end of 2021, i.e., approximately one-tenth of all visible prefixes are hyper-specific (see §3.4 for further details). Relative to the increase in HSPs, we also observe an increase of ASes that originate them, with 584 and 2.5K ASes announcing hyper-specific prefixes via IPv6 and IPv4 by the end of 2021, respectively.

Given that the route collector projects acquired feeder ASes within our observation period, the increasing trend could simply be a sampling error. To test this hypothesis, we replicate the analysis using only data from the 105 IPv4 and 45 IPv6 feeder ASes

[3]These ASes announced either (1) an extraordinary high number of HSPs (i.e., 100 or more times higher than in other snapshots) or (2) HSPs in an extraordinary high number of anchor prefixes for a limited time.

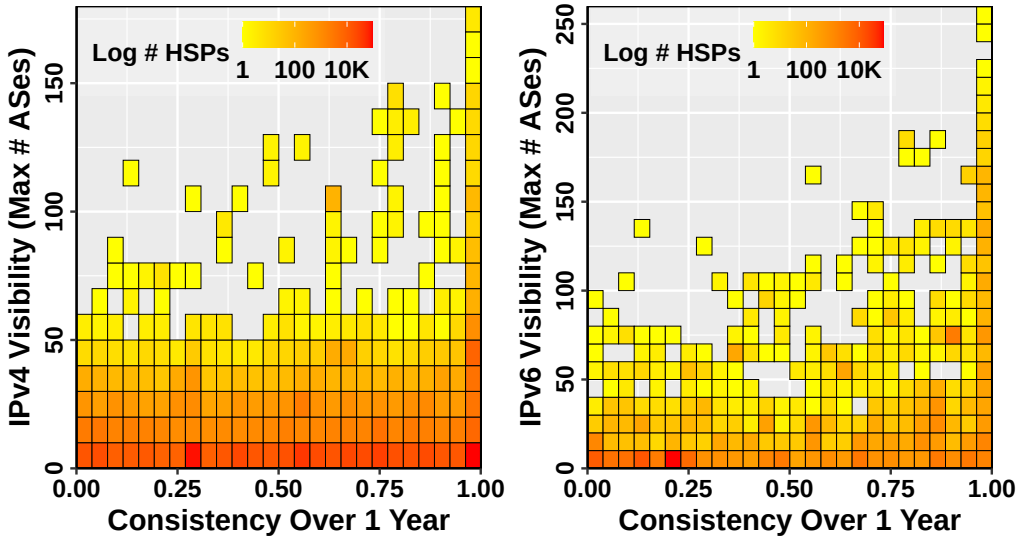

Figure 3.3: Heatmap showing HSP visibility and consistency for IPv4 (left) and IPv6 (right).

that are consistently peering with route collectors throughout all snapshots. While our observations remain similar for IPv6, there are two changes for IPv4: (1) the number of hyper-specific prefixes that can be seen by a consistent set of ASes appears more stable (if any trend exists, it remains hidden behind the massive fluctuations); and (2) despite an initial increase, the number of ASes originating HSPs stagnates after 2016. Therefore, the number of IPv4 HSPs does not show a constant increase over time, but rather we observe more IPv4 HSPs due to an increase in feeder ASes at route collector projects.

This hypothesis check leads to another observation: When shrinking the set of feeder ASes, the number of HSPs and their respective origin ASes drops substantially (note the different y-axes for the left and right subplot of Figure 3.2). To improve our understanding of this insight, we analyze the visibility of HSPs, i.e., by how many peers each HSPs is seen. At the same time, we want to understand what causes the substantial fluctuations in the number of HSPs; hence, we also analyze their consistency, i.e., the fraction of time for which the prefix was seen by at least one feeder AS. Given that a one-week observation period would not provide much insight into consistency patterns, we conduct this analysis using data from the entirety of 2020. We first read the RIB snapshots from January 1, 2020 and then apply all updates for the whole year sequentially. By tracking the state of each routing table on a per-update basis, we can extract consistency in seconds granularity.

Figure 3.3 reports the visibility of an HSP on the y-axis against its consistency on the x-axis. For both heatmaps—IPv4 (left) and IPv6 (right)—each cell represents groups of ten feeder ASes on the y-axis and two weeks of time on the x-axis. We first observe that there is no particular consistency trend: While some HSPs can only be observed for less than two weeks, others can be observed throughout the entire year. Our second observation is that the vast majority of hyper-specific prefixes can only be observed by a small number of collector peers, although we do also observe HSPs being visible during the entire year by hundreds of peers. This observation aligns with the restricted propagation characteristics of HSPs reported by previous blog posts [5, 6, 273] and observed by our own active experiments (an in-depth description of the experiments,

their analysis, and subsequent results can be found in §3.5). We hypothesize that the substantial fluctuations in the number of totally observed HSPs is a result of these two observations; the restricted propagation of HSPs might inflate the importance of the individual placement of feeder ASes and HSP origin ASes, and the tens of thousand of short-lived HSPs might cluster around certain real-world events, such as DDoS attacks or data center outages.

In summary, we observe that the presence of hyper-specific prefixes in the Internet's routing ecosystem has increased through the last decade and HSPs make up about one-tenth of all the prefixes that are observed by route collectors. In IPv4 the increase in HSPs is driven by an increment in feeder ASes, whereas in IPv6 we see an increase also for a constant set of feeder ASes. While most HSPs only propagate locally, some of them are globally visible and can be consistently observed throughout an entire year.

3.3 Use Cases and Functions

Given their past and current presence in the global routing system, we want to get a deeper understanding of the functions that hyper-specific prefixes potentially serve.

3.3.1 Which CIDR Sizes are most Common for HSPs?

As a first step in this direction, we use the fact that specific CIDR sizes often hint towards certain use cases. Consider the following example: If an AS wants to defend one of its servers against an ongoing DDoS attack, it may use blackholing announcements. Up to 98 % of these announcements are /32 (/128) IPv4 (IPv6) prefixes, i.e., they only cover the specific addresses of the attacked servers [78, 79, 103]. Larger CIDR sizes are rarely used for blackholing, as they would impair the services running on non-attacked servers as well, i.e., they would introduce unnecessary collateral damage [198]. Using similar lines of reasoning, we rely on the following associations between CIDR sizes and intended use cases: We associate (1) /25 and /26 IPv4 prefixes with traffic engineering (e.g., selective announcements [55, 219]), (2) /29 and /30 IPv4 prefixes with (Point-to-Point) peering subnets (i.e., the subnets needed to form inter-AS connections) [224], (3) /31 and /32 IPv4 prefixes with blackholing [78, 79, 103], (4) /49 to /64 IPv6 prefixes with address block reassignments [208], and (5) /113 to /128 IPv6 prefixes again with blackholing[4].

Figure 3.4 shows the number of IPv4 (left) and IPv6 (right) HSPs over time colored by their respective CIDR size groups. We first observe that the overall trends are stable over time. In IPv4, we observe that the most common CIDR size is /31–/32, i.e., the most prominent use case seems to be blackholing. Yet, we also observe that /29–/30 HSPs are comparably common; hence, many HSPs may actually represent peering subnets. Given that only about 10 % of HSPs have a CIDR size of /25 or /26, we believe that traffic engineering is a rare use case. For IPv6, we mainly observe the /49–/64 CIDR size range that we associate with address block relocations. In some ASes we also observe instances of /64s being used by hypergiants for off-nets [98]. We further observe a small fraction of /113–/128 CIDR sizes that we associate with blackholing. The share of blackholing HSPs is smaller in IPv6 compared to IPv4, which is in line with reports that blackholing in IPv6 makes up less than 2 % compared to IPv4 [103, 198]. Those observations also explain

[4] In private conversations a large European IXP confirmed that around 90 % of all blackholed IPv6 prefixes fall into the /113 to /128 prefix range.

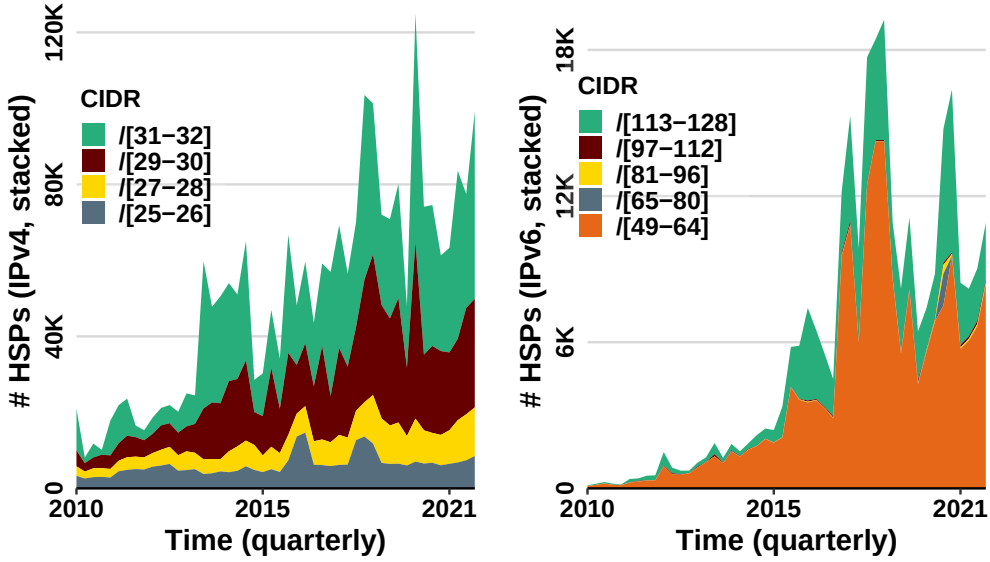

Figure 3.4: HSPs per CIDR size over time.

some of the fluctuations that we observed in the previous section—blackholing events, and their subsequently announced prefixes, are often short-lived [198] and subsequently can cause substantial changes in the number of unique HSPs seen throughout a week.

3.3.2 What Services are Running in HSPs?

As our CIDR-based analysis only provides us with hints on the actual usage, we now also analyze the services hosted in hyper-specific prefixes. For this analysis, we leverage archived scanning data from Rapid7's Open Data platform [220] for 2019, 2020, and 2021. Rapid7 frequently scans the entire routed IPv4 address space[5] for more than 100 well-known TCP and User Datagram Protocol (UDP) ports. To compare regular with hyper-specific prefixes, we rely on the difference in protocol hit rate, i.e., we compare the fraction of responding hosts and total tested hosts[6] on a per-protocol basis. We observe that four out of the top five protocols with the highest hit rate for regular and HSP prefixes overlap; BGP is only present in the HSP top five while CPE WAN Management Protocol (CWMP) is only present in the IPv4-wide top five. For those six protocols, Figure 3.5 shows a the relative difference of hit rates between regular and hyper-specific prefixes, where a positive value indicates an increase of hit rate in hyper-specific prefixes. While Hypertext Transfor Protocol (HTTP) and HyperText Transfer Protocol Secure (HTTPS) overall only see an increase of +100 %, we observe strong differences when drilling down on a per-CIDR level: When considering only /32 prefixes, HTTP's hit rate increases by more than +500 % compared to its hit rate for IPv4-wide scans—which substantiates the association of the /32 CIDR size for blackholing. Even more pronounced than HTTP(S),

[5]Except for prefixes on their blocklist which were explicitly requested by network operators.

[6]Given that Rapid7 does not publish the state of their blocklist, we assume that all (at the time of the scan) routed IP addresses were tested. Additionally, we focus on analyzing what services are prominent in HSPs. We can not ensure that Rapid7 (or its upstream) does in fact receive the HSP announcements, as information about their probing vantage points and routing is not available.

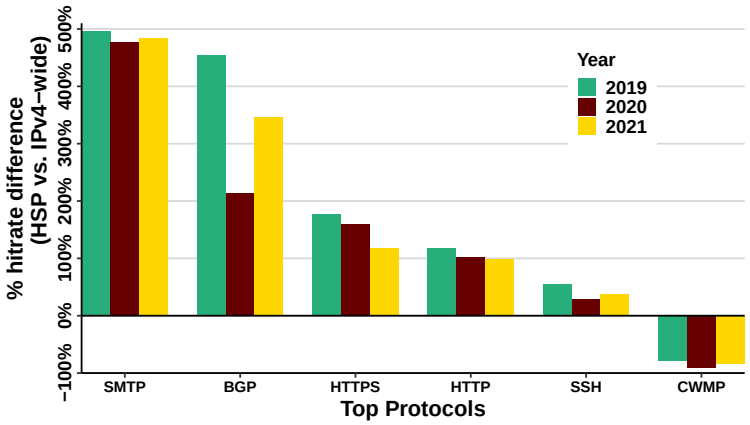

Figure 3.5: Hit rate comparison of HSPs vs. IPv4-wide.

Simple Mail Transfer Protocol (SMTP) and BGP see increases of up to +500%. When digging deeper we further observe that BGP is mainly prevalent in /30 and /29 prefixes, which underlines that these sizes might be dedicated to routing infrastructure. In contrast, we observe the only hit rate decrease (of more than 90%) for CWMP —a protocol used to remotely manage Customer Premises Equipment (CPE) devices such as home routers [278].

3.3.3 Are HSPs only Used for Blackholing?

We next take a close look at the association of hyper-specific prefixes to blackholing. One way of signaling a blackholing request is by attaching a pre-defined BGP Community to a hyper-specific (mostly single-address) announcement [271]. We investigate BGP communities attached to HSP announcements. BGP communities are used for many different reasons, such as information tagging, blackholing, or route redistribution. The most common BGP communities attached to hyper-specific prefixes are route steering or prepending instructions. In our analysis we look for BGP communities which are specifically used for blackholing (BH) [155] or Restrict Route Propagation (RES)[7]. Figure 3.6 shows the use of BGP communities among HSPs from snapshots between 2019 and 2021. The bars indicate the median share of HSPs with the respective community, the whiskers denote the standard deviation over time. The "Any" keyword is used to specify groups of community targets, e.g., "Any RES" describes all prefixes that have any restriction community attached (i.e., it refers to the union of prefixes with "NO_ADV" community and prefixes with "NO_EXP" communities); similarly, the "Any Comm." bar refers to the highest aggregation, i.e., prefixes for which we saw any community attached. As we can see, 60% of all IPv4 HSPs and almost three quarters of IPv6 HSPs come with some form of BGP communities. The vast majority of these communities is, however, not related to blackholing or restricting propagation. Only about 13% and 7% of prefixes can be associated with blackholing for IPv4 and IPv6, respectively. The by far most popular blackholing community is `X:666`. Moreover, we see no propagation restriction communities ("no advertise" or "no export") in IPv6 and only about 0.5% in IPv4. Furthermore, we see that RES communities are a subset of BH communities,

[7]We also test for communities such as NOPEER or NO_EXPORT_SUBCONFED, but these are not prevalent among HSPs.

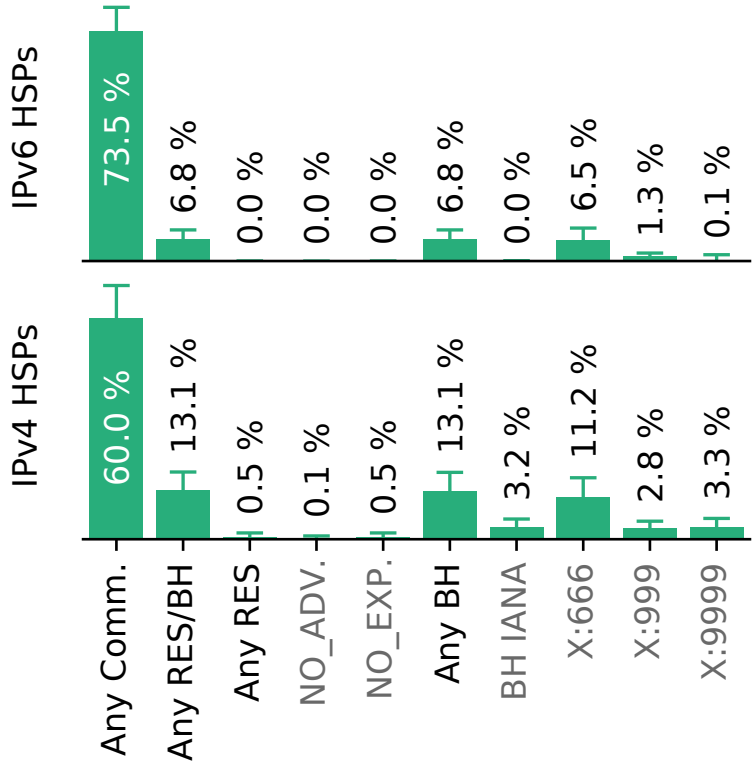

Figure 3.6: BGP communities distribution for HSPs.

hinting that operators do not want their blackholing prefixes to propagate. Blackholing is therefore one contributor of HSPs, but blackholing communities are not present on the majority of HSP announcements. We note that the blackholing communities that we see at route collector peers is a lower bound: Blackholing communities—similar to other communities—could be cleaned along the path but the prefix itself could continue to propagate [156].

In summary, we observe that for IPv4 many and for IPv6 some HSPs are likely related to blackholing activities due to the used HSP prefix size. We find concrete evidence for 7–13% of HSPs explicitly tagged with blackholing communities. While we also observe many HSPs dedicated to routing infrastructure (e.g., peering subnets or address relocations), we observe that hyper-specific prefixes rarely contain any CPE devices.

3.4 Intended or Accidental Use?

Now that we have a basic understanding of the use cases of HSPs, we want to analyze whether HSPs are used intentionally or accidentally by ASes and their operators. If operators take the time and effort to explicitly enter hyper-specific prefixes into voluntarily-maintained databases, then it is likely that they plan to use them. Hence, we look at the Resource Public Key Infrastructure and Internet Routing Registry operator databases.

We use private, three-monthly IRR snapshots [142] between January 1, 2017, and October 7, 2021, which contain information about routing policies. The RPKI database contains

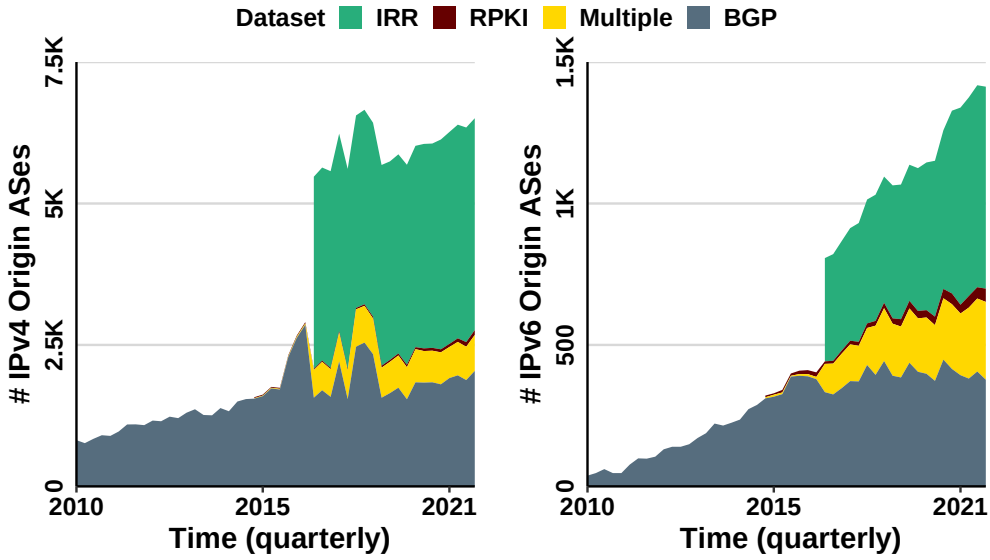

Figure 3.7: Visibility of origin ASes across datasets.

legally binding mappings between Internet resources and ASes. We use daily snapshots of the RPKI database [231] from April 1, 2015, until October 7, 2021, generated by Chung et al. [60] to verify the validity of HSP announcements by ASes.

While we extract HSPs directly from the `route(6)` objects contained in the IRR databases, the Route Origin Authorization objects in the RPKI snapshots describe CIDR size ranges [129]. Hence, a ROA can explicitly describe an HSP when both the minimum and maximum prefix length are hyper-specific, or implicitly when only the maximum prefix length is hyper-specific. When extracting HSPs and their origins from the RPKI database, we rely solely on explicit definitions as these clearly represent the desire to use HSPs (as all covered prefixes are hyper-specific). As implicit definitions might describe the future—but not necessarily current—use of HSPs (e.g., an AS might currently announce a /24 but has already entered a currently unused max-length of /25), we decide to ignore them. We compare the HSPs on those two databases against the HSPs visible via BGP route collectors.

Figure 3.7 shows the number of unique origin ASes for both IPv4 and IPv6 within each dataset over time. We classify those origin ASes available in more than one dataset into the "Multiple" category. Our first observation is that for both IPv4 and IPv6, the IRR dataset contains the largest fraction of HSP origin ASes. While this might imply that network operators tend to actually use HSPs, it is well-known that route objects can become stale given that the database is only maintained on a voluntary basis [258]. Yet, some entities, e.g. certain IXP Route Servers [73], require route objects in the IRR database to redistribute prefixes (i.e., HSPs). Even for the RPKI database we observe hundreds of explicitly defined HSPs[8]. Notably, for the last snapshot in October 2021, implicit HSPs would have increased the number of RPKI origin ASes from 294 to 990 for IPv4 and from 172 to 794 for IPv6, respectively. Beyond these intentional HSPs, we also observe that many of the HSPs from Route Collectors have no entries in operator databases, hence, they could potentially represent accidental announcements or misconfigured

[8]Most of these HSPs are also in the BGP data set and hence end up in the multiple class.

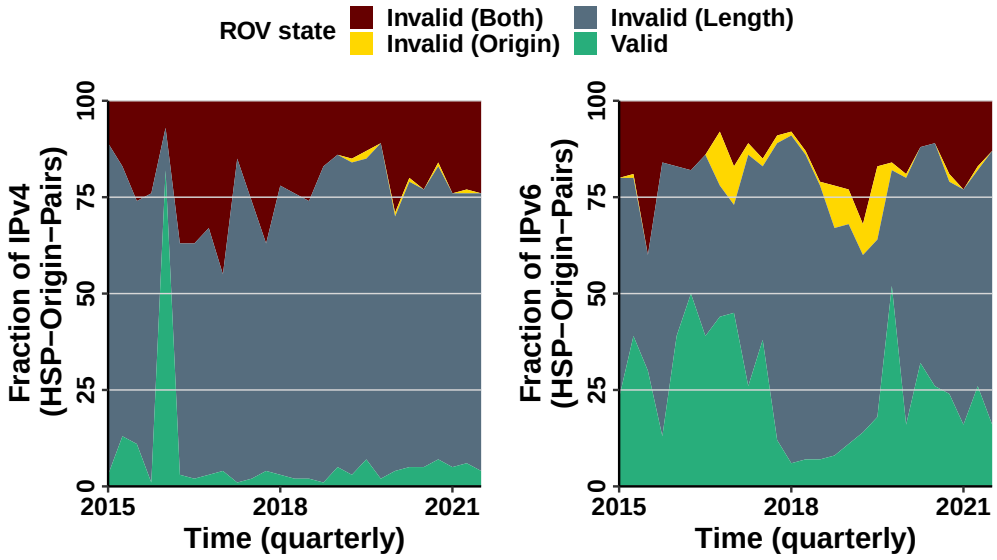

Figure 3.8: ROV status for HSPs

route collector sessions that leak internal routes.

While it is hard to link malicious intent to a more-specific announcement (since it could be, e.g., an address leasing agreement [214] or traffic engineering of sibling ASes [94]), we want to understand if the visible HSPs in the BGP are legitimate prefix advertisements by valid origin ASes or associated with possible prefix hijacks. Therefore, we perform route ROV of HSPs and its origin AS by checking them against the ROA records from the RPKI dataset. If a ROA covers the address space described by the prefix, then this prefix can violate the ROA in two ways: it can be too specific—which we mark as "Invalid (Length)"—and it can be announced by a different origin—which we mark as "Invalid (Origin)." If both of these conditions are met at the same time, we mark a prefix as "Invalid (Both)." If none of these conditions are met, we consider the prefix as "Valid." Notably, we observe that 22 % of IPv4 and 19 % of IPv6 HSPs have a covering ROA entry (median percentages across snapshots in 2020 and 2021).

Figure 3.8 shows that legitimate ASes, i.e., the valid and invalid length categories together, advertise around 75 % of all HSPs. With an average of 25 % and peaking to around 50 % in 2016, 2017, and 2019, IPv6 has a higher percentage of valid HSPs than IPv4. The HSPs with invalid length form the largest group in IPv4, and mostly the second largest group in IPv6. The third largest group of HSPs has the "Invalid (Both)" ROV state, while the invalid origin category forms a minor fraction of HSPs' ROV state. Legitimate ASes advertise around 75 % of HSPs, which indicates that HSPs are not majorly associated with BGP prefix hijacks. Beyond malicious ASes, the "Invalid (Origin)" and "Invalid (Both)" status could also be caused by not properly entered sibling ASes [94] or from a DDoS Protection Service (DPS) [148]. We analyze how many hyper-specific prefixes are originated from a DPS as identified by Jin et al. [148] and find that only around 1 % of HSPs in IPv4 and IPv6 are related to DPS companies.

In summary, we observe that for both IPv4 and IPv6, hundreds of ASes intentionally entered hyper-specific prefixes into operator databases. Yet we also see that many of the HSPs that are visible from route collectors have no respective entries and are likely

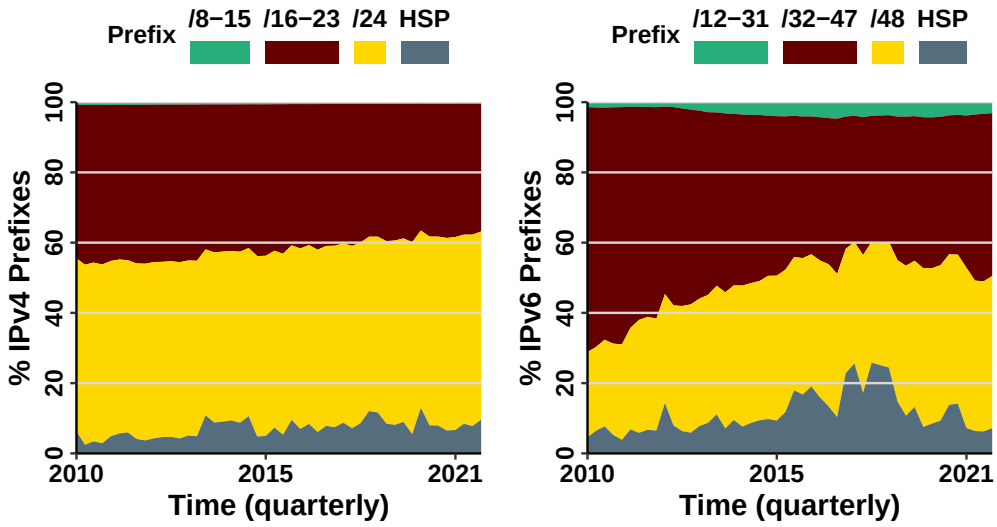

Figure 3.9: HSP prefix contribution over time

related to the accidental announcement or disclosure of internal routes. This is further substantiated by the observation that most HSPs are actually ROV invalid since they are more specific then intended by their covering ROA entry.

3.4.1 How prominent are HSPs?

To understand the prevalence of hyper-specific prefixes, we aggregate the routing tables of all collector peers and compare the distribution of prefixes depending on CIDR sizes. Figure 3.9 shows those distributions as stacked bar plots for each snapshot. We observe that up to 13 % (in 2015) and 25 % (in 2018) of totally visible prefixes are hyper-specific for IPv4 and IPv6, respectively. Yet, the usual contribution of HSPs is approximately 10 % for most months. Note that this does not mean that any single routing table contains that many HSPs on its own.

3.4.2 How visible are HSP?

To further elaborate on this point, Figure 3.10 shows the number of hyper-specific prefixes per IPv4 (left) and IPv6 (right) snapshot separated based on the number of route collector peers that can see them. For IPv6, we observe that most hyper-specific prefixes can be seen by two or more peers, with around a fifth of all HSPs being visible by 11+ peers for most snapshots. Similar to the previous plot, we again observe a peak of (~20K) hyper-specific prefixes at around 2018. While we are not able to account this peak to a single factor, we observe that the increase is rather uniform across collector peers, origin ASes, intermediate ASes, and address space and, hence, is unlikely to stem from a measurement artifact or some local misconfiguration. When comparing the situation before and after the peak, we still can see an increase from ~7K HSPs in 2016 to ~11K HSPs in 2021. In contrast to IPv6, many HSPs in IPv4 can only be seen by one peer. While we observe few HSPs that can be seen by 100+ peers, the vast majority of HSPs can only

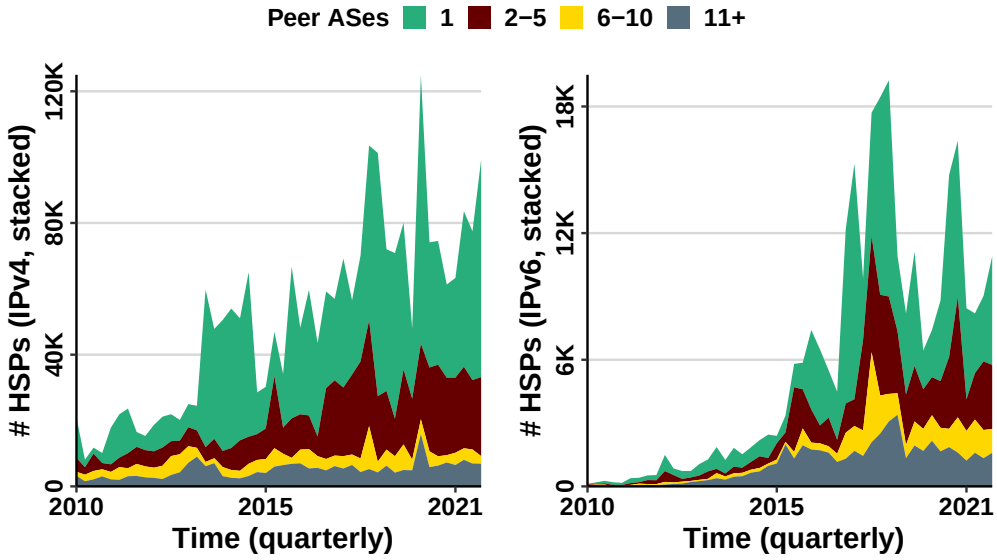

Figure 3.10: HSPs by # Peer ASes over time

be seen by 10 or less peers. Even though the number of low-visibility HSPs strongly fluctuates between snapshots, it increases rather continuously across many snapshots. Both such characteristics are significantly less pronounced for IPv4 HSPs that can be seen by 6+ peers. This difference may be accountable to various reasons including the association of a prefix to a certain function or a prefix's lifetime.

3.4.3 HSP Aggregation

ASes often have economical incentives to keep their BGP routing table size low. To realize this goal, some ASes aggregate (multiple) more-specific routes into a single less-or-equally-specific route [63]. If an anchor-prefix results from aggregating prefixes with different CIDR sizes (prefix-based aggregation), we know that one of such pre-aggregation prefixes must have been hyper-specific. Yet, confidently identifying such aggregations is challenging. According to RFC 4271 [221], a router *MAY* set the `AGGREGATOR` field when it performs prefix-aggregation—which can serve as indication that *some* form of aggregation must have happened. Thus, we first extract all routes for anchor-prefixes which have the `AGGREGATOR` field set. At this stage, our selected routes might be a result prefix-based aggregation or the aggregation of different routes—e.g., with different `AS_PATH` attributes (path-based aggregation)—for the same prefix (or both). To reduce the likelihood of falsely identifying HSP usage due to path-based aggregation, we rely on the `ATOMIC_AGGREGATE` field as well as the presence of `AS_SET` elements in the `AS_PATH` attribute. A router *SHOULD* set the `ATOMIC_AGGREGATE` field if the newly generated `AS_PATH` attribute of the post-aggregation route does not contain all AS numbers present in the pre-aggregation routes, e.g., the paths AB and AC can be aggregated to AB (which hides the existence of C). If the `ATOMIC_AGGREGATE` field is not set, ASes often use `AS_SET`s to signal path-aggregation, e.g., the paths AB and AC can be aggregated to $A\{B, C\}$ (where {...} denotes the `AS_SET` containing all ASes after A). As the `ATOMIC_AGGREGATE` field and `AS_SET`s indicate path-based aggregation, we remove all anchor-routes that contain at least one of them.

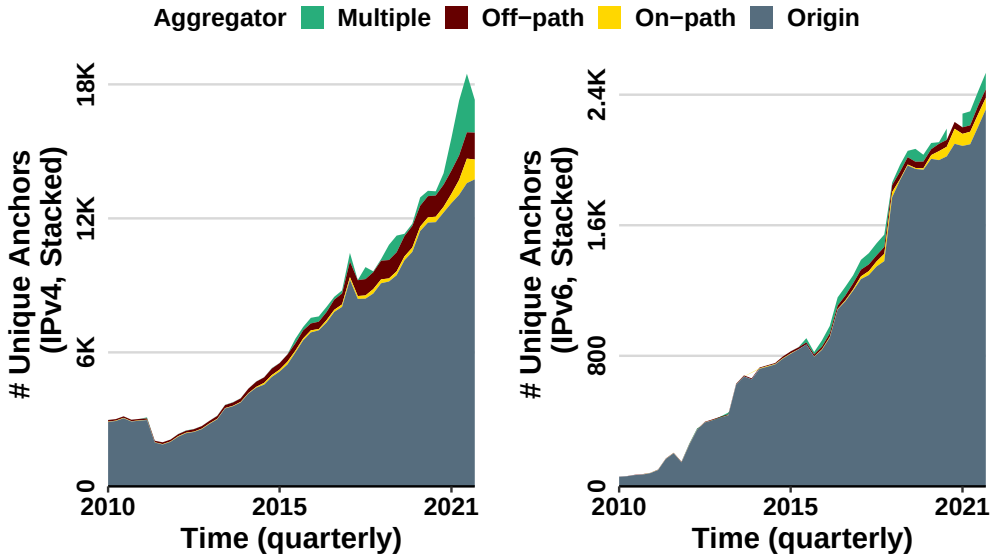

Figure 3.11: Position of HSP Aggregation

3.4.4 Where does HSP Aggregation Happen?

Now that we have a set of anchor-prefixes that are likely the result of prefix-based aggregations, we can analyze how close to the origin HSPs are aggregated. We compare the AS number in the `AGGREGATOR` field with the `AS_PATH` and differentiate between the following cases: (1) **Origin**—the origin itself performed the aggregation, (2) **On-path**—an AS within the AS path that is not the origin performed the aggregation, and (3) **Off-path**—some AS that does not occur in the AS path performed the aggregation[9]. Figure 3.11 shows the number of anchor prefixes in each class over time. Notably, the figure also contains the class **Multiple** that contains anchor prefixes for which there are multiple paths with inconsistent classes. We observe that the the vast majority of anchors are actually aggregated at the origin with only few hundreds of anchors being aggregated on-path. Origin and off-path (especially `AGGREGATOR` fields with private ASNs) aggregation often occurs due to the use of BGP confederations [61, 150] where the AS is internally split into multiple private sub-ASes. Depending on how an AS border router handles the aggregation of internal confederation routes, it might either correctly set the external AS number or leak the internal confederation AS Number in the `AS_PATH` or `AGGREGATOR` attribute. Notably, those HSP routes are likely not available to other ASes (including neighbors of the origin).

3.4.5 Projected Actual Usage

While our IRR snapshots produced actual HSPs, our final prefix-aggregation and ROAs only produced a list of anchor-prefixes that is likely to contain HSPs. Therefore, we decided to analyze the potential extent of HSP usage on the basis of anchor-prefixes. Figure 3.12 shows the number of IPv4 (left) and IPv6 (right) anchor-prefixes per data set (stacked) over time. Notably, the aggregated class only contains on-path aggregated

[9]This class also includes reserved AS numbers.

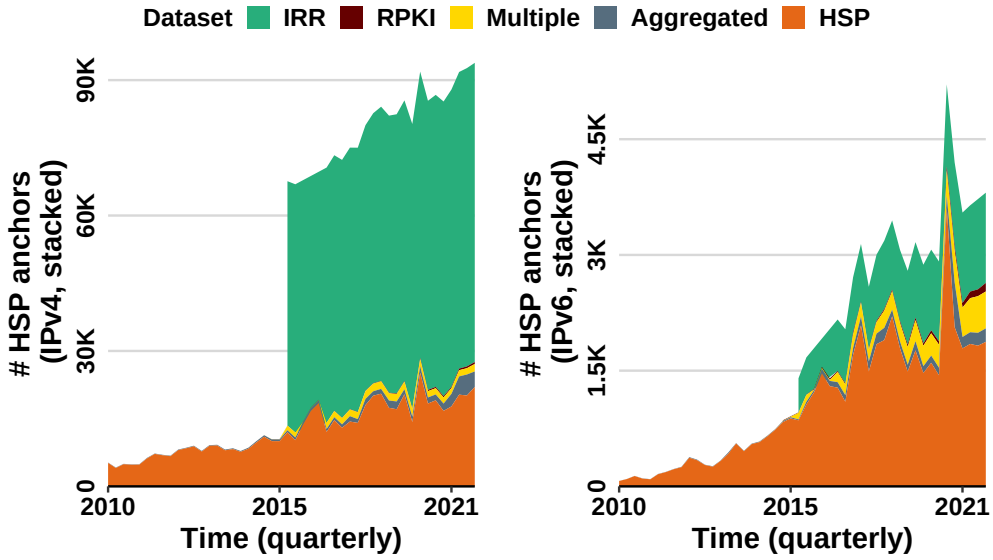

Figure 3.12: HSP anchors across datasets

anchor prefixes and the RPKI class only contains anchor prefixes for explicit HSP ROAs. The "multiple" class covers those entries that are visible via multiple data sources. We observe that the current route collector infrastructure misses roughly one-third of the of the anchor prefixes that potentially contain HSPs. We further observe a less noisy, linear increase in the number of anchor prefix for which HSPs are visible compared to the raw count of visible HSPs. Notably, some part of this increase can potentially be accounted to the increasing numbers of route collectors and route collector peers over time.

3.4.6 Who uses hyper-specific Prefixes?

We leverage the "AS Classification Inferences" dataset described in ASDB [298] to classify ASes as Content, Education, Hypergiant, ISP (Stub), ISP (Transit), Tier 1, and Others. Figure 3.13 compares the classes of all BGP-visible ASes (left) to HSP origin ASes (right) over time. We find that in contrast to all origin ASes, HSP origins are more likely to be ISP (Transit) ASes. Interestingly, the majority of Tier 1 ASes is also originating HSPs. During the period of January 2019 until October 2021, we identify between 12 and 15 of the total 19 Tier 1's as HSP origins. In contrast to the high share of Tier 1 HSP origins, we find that most hypergiants do not originate HSPs.

3.5 Real-World Experiment

Finally, we want to understand how much the lack of additional BGP vantage points impacts our observations on reachability. Hence, we configure a real-world experiment using the PEERING testbed [248] in which we announce an anchor prefix as well as multiple hyper-specific prefixes. Once those prefixes have converged, we run traceroutes from RIPE Atlas [230] probes and compare their resulting paths to those visible at route collectors.

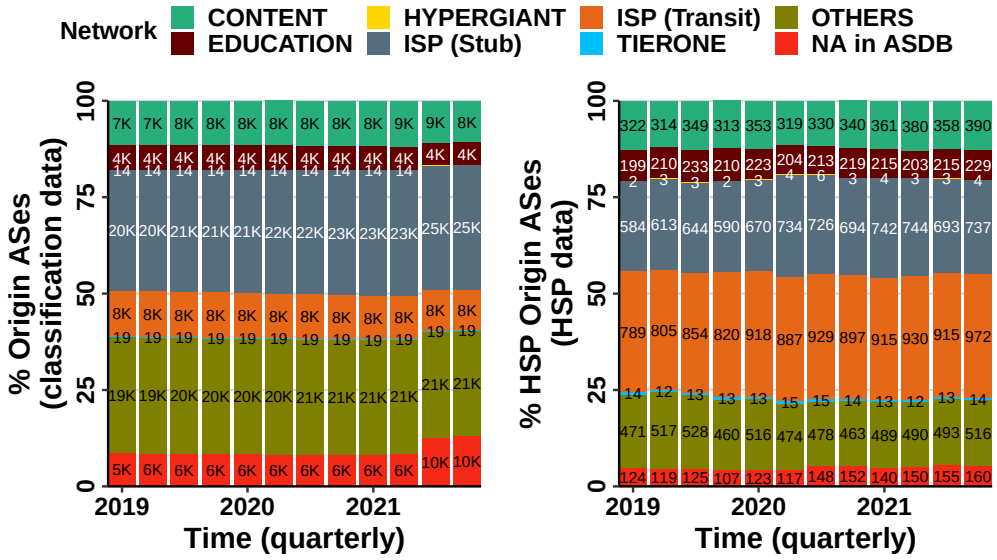

Figure 3.13: HSP origin AS classification over time.

3.5.1 Vantage points and Resources

The PEERING testbed allocates Internet resources (specifically, IPv4/IPv6 address space and AS numbers) to its users based on approved experiment proposals. Once allocated, users can announce those resources via the testbed's infrastructure. Given that the PEERING testbed strongly relies on third party resources (e.g., for hosting infrastructure), announcements must be designed carefully to not cause trouble or irritation for other network operators. For our experiment we use the address ranges 184.164.240.0/23 and 2804:269c:4::/46. More specifically, we utilize 184.164.240.0/24 and 2804:269c:4::/48 as anchor prefixes (i.e., they represent our control group) and announce HSPs only from the remaining address space[10].

RIPE Atlas [230] is a measurement platform with probing devices (henceforth called probes) all over the world. To maximize probing coverage and minimize probing load, we choose at most one probe per AS. To reduce the likelihood of probe outage, we select only probes that are not tagged with system-problematic tags[11]. We further validate that selected IPv4 and IPv6 probes are tagged with `system-ipv4-works`, and `system-ipv6-works`, respectively. If an AS hosts multiple probes, we prefer dual-stack probes (such that we can use a consistent probe for our IPv4 and IPv6 measurements) over anchor probes (i.e., better equipped probes) over any other probes. If we still have multiple choices, we pick the probe that is tagged with the highest stability tag (e.g., `system-ipv4-stable-90d`). Our final probe set consists of 3097 probes distributed across 2990 IPv4 and 1239 IPv6 ASes.

[10] In particular, we announce 184.164.241.0/25, 184.164.241.128/28, 184.164.241.255/32, 2804:269c:5::/49, 2804:269c:6::/64, 2804:269c:6:8000::/65, and 2804:269c:7::/128.

[11] tags: `system-flash-drive-filesystem-corrupted`, `system-v1`, `system-no-flash-drive`, `system-flash-drive-bad-or-too-small`, `system-firewall-problem-suspected`, `system-trying-to-connect` `system-readonly-flash-drive`, `system-no-controller-connection`, `system-bad-firmware-signature`, `system-flakey-connection`, `system-flakey-power`, `system-flash-drive-problem-detected`, and `system-v2`

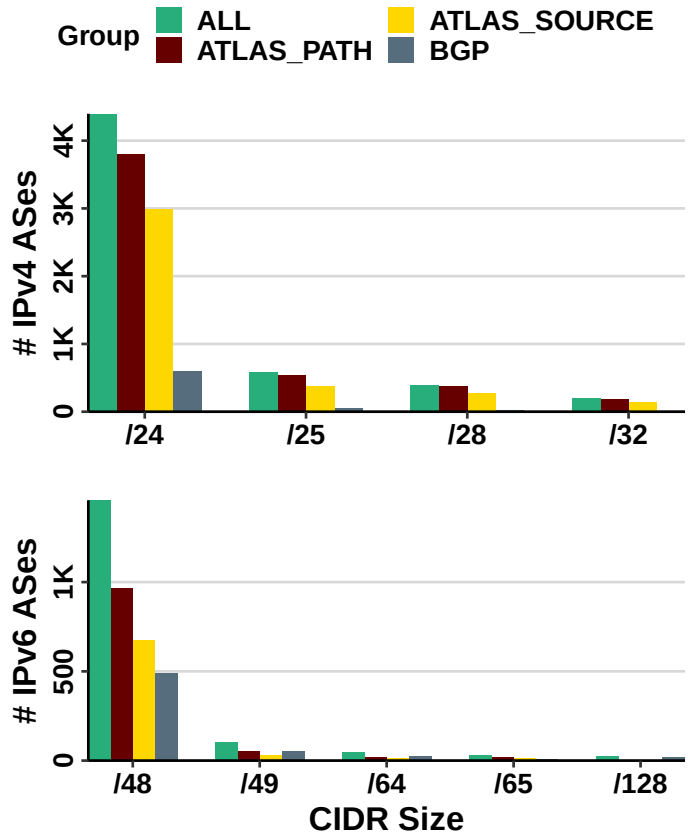

Figure 3.14: PEERING testbed propagation results

3.5.2 Experimentation Environment

The PEERING testbed currently has a total of 180 IPv4 and 152 IPv6 neighboring ASes. Yet, most neighbors do not support/redistribute HSPs. We identify supportive neighbors by iteratively announcing a /25 or /49 prefix from our allocated address space through each neighbor and analyzing the resulting update stream from RIPE RIS and Routeviews. Since, at this point, we only care about a "life sign" (i.e., whether or not *any* update was received) rather than full convergence, we adopt a short announcement cycle: We announce a prefix at the beginning of every full hour and withdraw it 30 minutes later[12]. We identify a set of 8 IPv4 and 9 IPv6 neighboring ASes that redistribute HSPs. Notably, those ASes are distributed across 4 and 3 geographically separate Points of Presence.

3.5.3 Technical Realization

Throughout May 21 and 22, 2021, we announce /24, /25, /28, and /32 IPv4 prefixes and /48, /49, /64, /65, /128 prefixes through a single neighbor at the beginning of every even hour. After the announcement, we wait 40 minutes to allow the prefix to converge[13].

[12]These experiments ran between the May 1, 2021 and the May 3, 2021.

[13]During previous experiments we observed that usually the 95th percentile of updates reach the collector peers already in the first 15 minutes.

After those 40 minutes, we run active measurements for 10 minutes, and then withdraw the prefixes again. Notably, we choose 70 minutes between a withdrawal and the next announcement on purpose such that we out-wait the expiration of potential Route Flap Damping hold-down timers, which have been shown to usually expire after 60 or less minutes [109].

During our 10 minute active measurement period, we run paris-traceroutes from all probes towards either the network address or the first non-network address of all prefixes (which are configured to be pingable). To reduce the dependence of our results on the underlying protocol, we simultaneously issue Internet Control Message Protocol (ICMP), TCP, and UDP probing. To keep the induced load for the RIPE Atlas platform as well as for the peering testbed manageable, we reduce the number of probing packets used per per hop by paris-traceroute from RIPE ATLAS' default of 3 packets to one packet. Notably, as the resulting load still exceeds the default limitations (e.g., for measurement results per day) for a single RIPE Atlas account, we coordinate our probing efforts with the RIPE Atlas team who generously raised the limits for our experiments.

We map traceroutes to AS Paths using the state-of-the-art mapping tool bdrmapit [184]. As bdrmapit requires a large corpus of traceroutes as input to perform well, we use traceroute data from CAIDA's IPv4 Prefix-Probing data set [44], CAIDA's IPv4 Routed /24 Topology Dataset [44], CAIDA's IPv4 Routed /48 Topology Dataset [45], and RIPE's hourly archives of Atlas traceroutes [199] between May 17, 00:00 and May 24, 00:00. For all the other inputs (e.g., prefix-to-origin mappings or business relationship inferences) we use recent snapshots from the recommended data sources. Finally, we use bdrmapit's output to map our successful (i.e., only those that actually reached the respective target host) traceroutes to AS paths.

3.5.4 Comparison

Figure 3.14 compares the the number of ASes (aggregated over all iterations) that (1) hosted Atlas probes that reached the target (ATLAS_SOURCE, yellow), (2) appeared along the path between ATLAS_SOURCE ASes and the Peering Testbed (ATLAS_PATH, dark red), (3) are visible from route collector peers (BGP, gray). The most drastic observation is that hyper-specific prefixes see a very sharp drop in reachability. Even the best performing CIDR size, /25, only reached ~15 % of of the ASes that are reached by its respective anchor prefix. Especially for IPv6 we observe that most PEERING neighbors redistribute our prefixes (including the anchor prefix) only towards their customers, hence, some of our Atlas probes are unable to reach the peering testbed even for the anchor prefix. We further find that the more-specific the prefix gets, the less likely it propagates. This finding is interesting as most recommended filtering guides [81, 82, 110, 183, 202] treat all hyper-specific CIDR sizes equally. Our third observation is that the reachability reflected by route collector peers substantially underestimates data plane reachability. While we are able to observe approximately a third of the total ASes for our /48 prefix via BGP, this fraction lies at around 14 % for our /24 prefix.

3.6 Discussion

Research Community. While many HSPs seem to be intentional, we also observe a large number that potentially represent leaked internal routes. While the task of reconfiguring a leaking router ultimately belongs to the feeder AS' operators, we believe

that the maintainers of route collector projects play a vital role when it comes to raising awareness for the existing problems. To support and guide this process, we publish and maintain a dashboard that provides up-to-date HSP statistics as well as a rankings of the top HSP contributors at `https://hyperspecifics.io`. Beyond fixing potential leakage errors, we believe that studying the potential correlations between hyper-specific prefixes and their less-specific counter parts may lead to new insights into the routing optimizations used by ASes.

Operator Community. Even though various guides [81, 82, 110, 183, 202] recommend strict filtering of HSPs, we find that many hyper-specific prefixes propagate to 100 or more collector peers. After discussing our results with thirteen operators from different types of networks, we believe that the limited filtering is often a result of popular customer requests. The operator of a major transit network told us that their network recently (throughout Summer 2020) changed from the filtering of all IPv4 HSPs to only filtering prefixes more specific than /28; this shift enabled (especially new and small) customer networks to perform basic traffic engineering despite a limited address allocation[14].

This opens up the question whether operators should filter HSPs in the first place. We believe that for IPv6 the answer is a resounding "yes". Given that there is no shortage of IPv6 addresses and obtaining new blocks is virtually free (compared to the high costs of obtaining IPv4 addresses), we do not see any reason to loosen the current filtering guidelines. For IPv4, we think that the answer should be more nuanced. While loosening the filtering guidelines allows even small ASes to perform traffic engineering, it would also further increase the routing table size. Hence, we believe that shifting the acceptable boundaries by a few CIDR sizes (e.g., /26 or /28) might be an agreeable compromise.

3.7 Related Work

In this section, we report on related work in the areas of hyper-specific prefix analysis and prefix deaggregation.

HSP Analysis: Previous research in this area consists mostly of blog posts. In 2014, Aben and Petrie report on an experiment where they announced /24, /25, and /28 IPv4 prefixes and ran RIPE Atlas measurements to them [5]. Their findings show that HSPs are visible for at most 20 % of RIPE RIS peers [236] with route objects slightly improving the visibility. The RIPE Atlas experiments lead to similar results with fewer than 15 % of probes reaching their targets. One year later, Aben and Petrie revisit the propagation of hyper-specific prefixes and find a marginal increase of a few percent [6]. In 2017, Strowes and Petrie conclude that not much has changed regarding hyper-specific prefix propagation and at most one fourth of all BGP peers receive those announcements [273].

Prefix Deaggregation: In 2002, Bu et al. first characterize prefix deaggregations and the reasons for them, e.g., traffic engineering, multi-homing, and address fragmentation [34]. Meng et al. report in 2005 that even newly assigned address space is deaggregated and that the deaggregation rate of prefixes increases over time [186]. In 2010, Cittadini et al. [63] report that more than 10 % of ASes deaggregate their prefixes while around 1 % of ASes announce more than 10 prefixes for each address block they got assigned. Lutu et al. present a simulation model that estimates that origin ASes can reduce their transit cost by 5 % by using more-specific announcements [176–178]. Notably, the authors neither focused on IPv6 nor on hyper-specific prefixes. In 2016, Krenc and Feldmann analyze the address delegations realized via prefix deaggregations and report

[14] This is a direct result of the current IPv4 Address exhaustion and the subsequently inflated prices [214].

on delegations from customers to providers or between unrelated ASes (often involving CDNs) [157]. In 2017, Huston analyzes the prevalence and different types of more-specific prefix announcements in the Internet as an effect of prefix deaggregation [130]. His taxonomy attributes MSPs to three different root causes, hole punching (different origin AS), traffic engineering (same origin AS, but different AS path), and overlay (same AS path). He concludes that the former two play a useful role for network operators, while the usefulness of overlay more-specific prefixes could be argued about. Huston did not specifically investigate the effect of hyper-specific prefixes.

To the best of our knowledge, this study presents the first scientific analysis of hyper-specific prefixes by providing an in-depth look into the prevalence and possible root causes for HSPs in the wild.

3.8 Summary

In this part of dissertation, we analyzed the presence of hyper-specific prefixes in the Internet's ecosystem throughout the last decade. While we found an overall increase in the number of HSPs, most of them can only be observed by a few route collector peers. Yet, there are still plenty of HSPs that propagate to hundreds of route collector peers and can be consistently observed throughout an entire year. Inspired by those findings, we took a closer look at the function that these prefixes serve. For IPv4, we observed that HSPs are mainly associated with blackholing and infrastructure announcements (e.g., routes to peering subnets). While we only found limited evidence for any connection to traffic engineering, we observed that hyper-specific prefixes are less likely to contain end-user devices. For IPv6, we observe that almost all hyper-specific prefixes are related to address block reassignments, with only a small fraction representing blackholing. Even though we have seen that hundreds of networks use HSPs intentionally, we attributed even more cases to the accidental "leakage" of internal routes. Finally, we discussed the current state of HSPs from an academic as well as an operator point of view.

Chapter 4
Long-Lived MOAS Prefixes

Building on the previous chapter's investigation of IP prefix size violations, we now focus on another critical aspect of Internet routing: the number of Autonomous Systems originating the same IP prefix. This chapter focuses on long-lived Multi-Origin AS prefixes—those announced by more than one AS and visible in the global BGP routing system for at least 30 consecutive days.

In BGP, each IP address prefix (i.e., collection of IP addresses) is usually originated by a single AS [114]. There are, however, also cases where multiple ASes originate the *same prefix*, this prefix is in turn called a Multi Origin AS prefix. Network operators use MOAS prefixes to e.g., provide resilience, load balancing, and multi-homing.

In addition to these uses, MOAS prefixes can also be a result of mergers of companies operating two ASes, misconfigurations [104], and—most problematically—prefix hijacks. This last case occurs if an attacker hijacks traffic to a specific prefix by announcing this prefix with its own AS as an origin. Unfortunately, prefix hijacks are happening relatively frequently [24, 38]. Consequently, prefix hijacks have been the focus of numerous studies over the past years [26, 58, 138, 203, 217, 256, 257, 276].

Surprisingly, the use of long-lived MOAS prefixes—i.e., when they are visible for a longer time in BGP—remains understudied.

When analyzing RIB snapshots of all RIPE RIS [232] and Routeviews [281] route collectors, we find that the number of long-lived MOAS prefixes (i.e., visible for at least 30 days) is growing over time, as can be seen in Figure 4.1, both in terms of prefixes and origin ASes.

IPv4 long-lived MOAS prefixes increase from 10k in 2017 to over 24k prefixes at the beginning of 2023, with the number of origin ASes growing by about 50% in the same time period. Moreover, the fraction of long-lived MOAS prefixes out of all visible BGP prefixes is increasing as well (see §4.2.4). In this chapter, we focus on the analysis of these long-lived MOAS prefixes. We perform a longitudinal analysis across multiple dimensions to unveil prefix and origin characteristics, visibility, and users of MOAS prefixes. More specifically, the main contributions of this chapter are:

- **Methodology to Detect Long-Lived MOAS Prefixes:** We apply a rigorous methodology to detect long-lived MOAS prefixes over a period of six years (see §4.2).

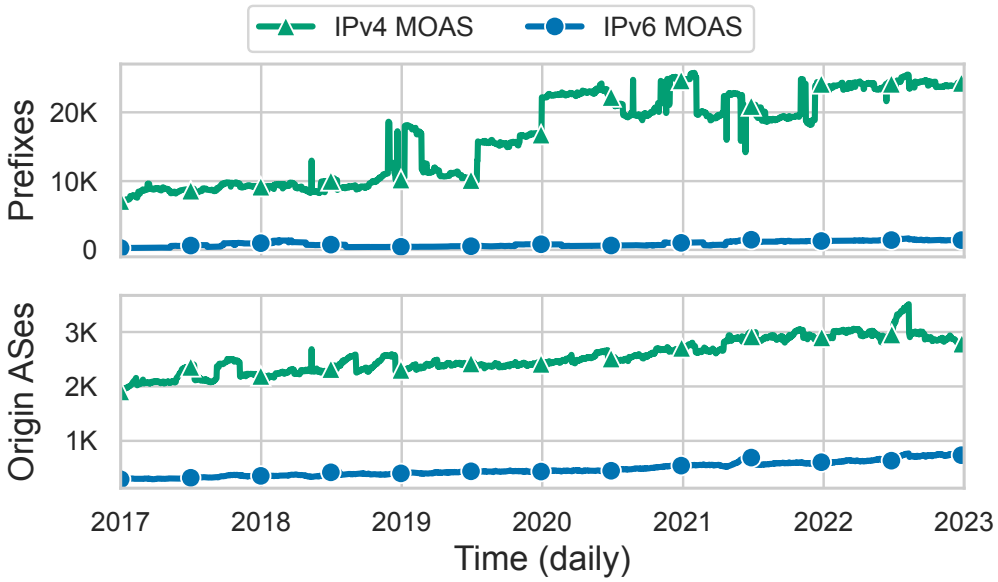

Figure 4.1: Number of long-lived MOAS prefixes and origin ASes over time.

We use the Kneedle algorithm to discern long-lived from short-lived MOAS prefixes and perform a sensitivity analysis to analyze the influence of route collector artifacts.

- **MOAS Prefixes and Origins:** We perform a longitudinal analysis of MOAS prefixes and origin ASes (see §4.3). We find that more than 40% of long-lived MOAS prefixes have a valid RPKI status for all origins, hinting that hijacks are not prevalent among them. Moreover, we identify that the vast majority of MOAS prefixes are announced by two origin ASes. The most commonly used CIDR sizes are /24 for IPv4 and /48 for IPv6, suggesting their use for fine-granular traffic steering. Additionally, we find a considerable number of MOAS prefixes linked to mergers of companies.
- **Visibility Analysis:** We analyze the visibility of MOAS prefixes across route collectors (see §4.4). Most MOAS prefixes are visible on hundreds of Route Collector peers with one origin AS, whereas the other origin AS is visible only on a handful of RC peers.
- **Users and Usage of MOAS Prefixes:** To better understand users and usage of MOAS prefixes we do an in-depth analysis on the AS relationship, their business types, and the prevalence of anycast within MOAS prefixes (see §4.5). Our results show that the majority of origin ASes are used by IT companies being in a customer-provider relationship, and only a small fraction of MOAS prefixes are used for anycast.

4.1 Datasets

To analyze the prevalence and evolution of MOAS prefixes in the Internet, we use the following datasets.

BGP Datasets: To get a good understanding of MOAS prefixes in the Internet, we use Route Collectors from two different route collector projects: RIPE RIS [232] and Routeviews [281]. We use 59 route collectors throughout our six years study period from January 1, 2017, to January 1, 2023. During the study period, we analyze daily RIB snapshots of all RCs for the duration of 2098 consecutive days. To calibrate our methodology (see §4.2) we compile an auxiliary BGP dataset by using three RIBs per day for a period of six months in 2022 from the same RCs.

AS Relationships Dataset: We use CAIDA's AS relationship dataset [37] from January 1, 2017, to January 1, 2023, to analyze inter-AS relationships of origin ASes associated with MOAS prefixes. This dataset is released by CAIDA every three months. The dataset classifies the inter-AS relationships into "customer-provider or provider-customer", "peer to peer", or "unknown" using the method described by Luckie et al. [175].

AS to Organizations Mapping Dataset: We use the inferred AS to organization mapping dataset provided by CAIDA [39] for the duration of our study from January 1, 2017, to January 1, 2023. The dataset is created by querying WHOIS databases from the five Regional Internet Registries (American Registry for Internet Numbers (ARIN), Latin American and Caribbean Internet Addresses Registry (LACNIC), RIPE, African Network Information Centre (AFRINIC), and APNIC), and two National Internet Registries (Korea Network Information Center (KRNIC) and Japan Network Information Center (JPNIC)) every quarter, and then applying CAIDA's AS to organization technique to map each AS to an organization [40]. Due to the lack of historical data, we are unable to use two recently presented improvements [17, 56] to the CAIDA AS to organization dataset.

ASdb Dataset: To gather detailed information about the type of organization using MOAS prefixes, we use the ASdb dataset [298]. ASdb provides a mapping of each AS into 17 industry categories and 95 sub-categories using data from established business intelligence databases and machine learning. We use ASdb to identify AS organization types and business relationships between the origin ASes of MOAS prefixes.

RPKI Dataset: We use archived RPKI records from RIPE NCC [234] in a monthly granularity from January 2017 to January 2023. With the RPKI dataset we identify the ROV state of MOAS prefixes.

Anycast IP Prefix Dataset: Moreover, we use a list of IPv4 and IPv6 prefixes classified as anycast prefixes by the BGP.Tools project [25]. The project uses an approach similar to MAnycast^2 [267] to run periodical anycast detection measurements. We use the published list of anycast IPv4 and IPv6 prefixes for the most recent date for our study, January 1, 2023, to analyze the prevalence of anycast deployments within detected MOAS prefixes.

4.2 Methodology

In this study, we focus on long-lived MOAS prefixes, i.e., prefixes that are announced from multiple origin ASes for a substantial period of time. In contrast to prefix hijacks [26, 58, 138, 203, 217, 256, 257, 276], such MOAS prefixes have received little attention from the research community in the past.

To identify long-lived MOAS prefixes and assess the impact of RC artifacts (e.g., missing RIBs, offline RC peers), we apply the following methodology: First, we extract all prefixes from the BGP dataset, apply a series of filters, and select prefixes appearing as MOAS. Next, we measure the lifetime of MOAS prefixes and perform a sensitivity analysis.

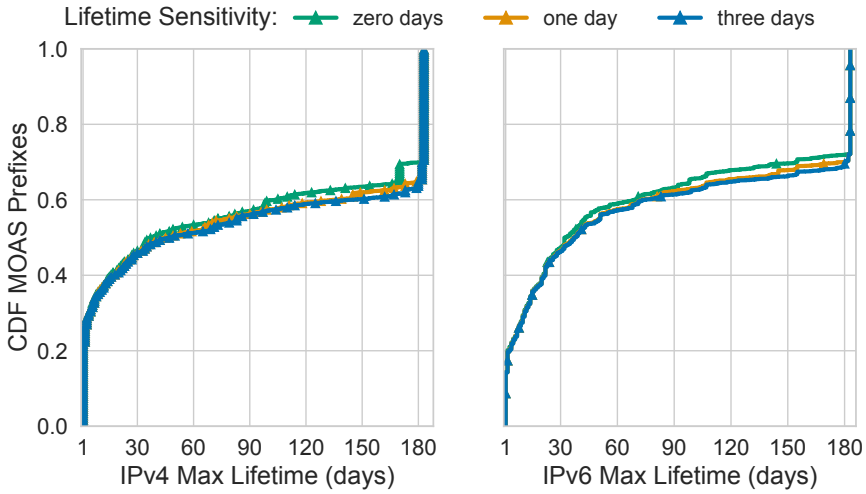

Figure 4.2: Cumulative Distribution Function (CDF) of maximum lifetime with various sensitivity thresholds for IPv4 (left) and IPv6 (right) MOAS prefixes.

Afterwards, we examine the effect of using more than one RIB snapshot per day in our pipeline. Finally, we identify and apply a lifetime threshold to distinguish between short-lived and long-lived MOAS prefixes. All these steps are detailed in the following.

4.2.1 Extracting, Filtering, and Selecting MOAS Prefixes

To develop our methodology, we download and use daily RIB snapshots from all 59 available RCs of the RIPE-RIS [232] and Routeviews [281] projects for a six-months period spanning from June 1, 2022, to November 30, 2022. Since we focus on long-lived MOAS prefixes, we do not use BGP UPDATE files, as those provide only additional short-term information. As a first step, we extract all prefixes from RIB snapshots of RCs and apply a series of filters to remove the following artifacts: private or reserved origin AS numbers [192], reserved IP prefixes, prefixes that should not be globally routed, and special purpose IPv4 and IPv6 prefixes [69, 134, 135]. Furthermore, we remove default route (i.e., `0.0.0.0/0`) or prefixes with network bits set beyond their CIDR size (i.e., `1.2.3.0/16`). As a result of applying filters, we remove around 1% of all prefixes from the BGP dataset.

Then we check the origin AS of every IP prefix across all RIB snapshots at the exact date and time[15], select prefixes with more than one origin AS, and consequently mark them as MOAS prefixes for further analysis.

4.2.2 MOAS Lifetime Sensitivity Threshold

As BGP data from RC projects can be incomplete, e.g., due missing RIBs or RCs temporarily being offline, calculating the lifetime of a MOAS prefixes requires us to be extra cautious. We use the lifetime, i.e., the duration a prefix is seen as a MOAS continuously

[15]We download available RIBs daily at 08:00:00 UTC from all route collectors. It is important to check the RIB snapshots for the exact date and time across all route collectors, otherwise a prefix could be falsely classified as MOAS appearing with a different origin ASes at two different points in time.

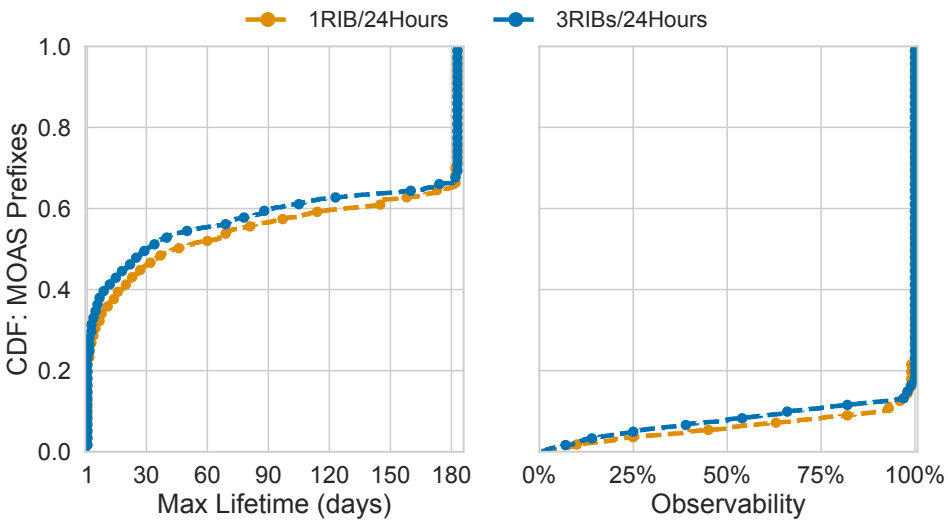

Figure 4.3: CDF of maximum lifetime (left) and observability (right) of MOAS prefixes for single and three RIBS per day.

in RIB snapshots, as a metric to identify and select long-lived MOAS prefixes. For each MOAS prefix observed in the BGP dataset, we calculate the maximum lifetime over the six months period. To assess the effect of BGP dataset incompleteness, we perform a sensitivity analysis of the maximum lifetime using different sensitivity thresholds. We investigate how the sensitivity threshold—i.e., the maximum number of continuous days a MOAS prefix can be missing from RIB data before its lifetime ends—affects the maximum lifetime of MOAS prefixes.

In Figure 4.2, we show the maximum lifetime for IPv4 and IPv6 MOAS prefixes for sensitivity values of zero, one, and three days. We find that more than 40% of IPv4 and around 50% of IPv6 MOAS prefixes have a maximum lifetime of fewer than 30 days. On the other end, around 35% of IPv4 and 30% of IPv6 MOAS prefixes are visible throughout the entire six months. Moreover, around 25% of IPv4 and 20% of IPv6 MOAS prefixes have a maximum lifetime of at most five days.

Looking at the effect of different sensitivity values, we notice that overall the one and three days sensitivity values slightly increase the lifetime of MOAS prefixes compared to the zero days sensitivity. This difference is most visible for IPv4 in the interval of 150 to 180 days, where we see the effect of a route collector outage when the RIB snapshot of Sydney RC for time 08:00:00 UTC is missing on June 13, 2022, but available for the rest of the time. Considering this and as a precaution to mitigate possible effects of unavailable RCs or RC peers on the lifetime of MOAS prefixes, we use the one day sensitivity threshold in the remainder of the chapter. We do not consider the three days sensitivity threshold, because it does not substantially increase the lifetime of MOAS prefixes beyond the one day threshold, and a gap of three days during which a MOAS prefix is not visible is large enough to possibly lead to incorrect calculations of MOAS prefix lifetimes.

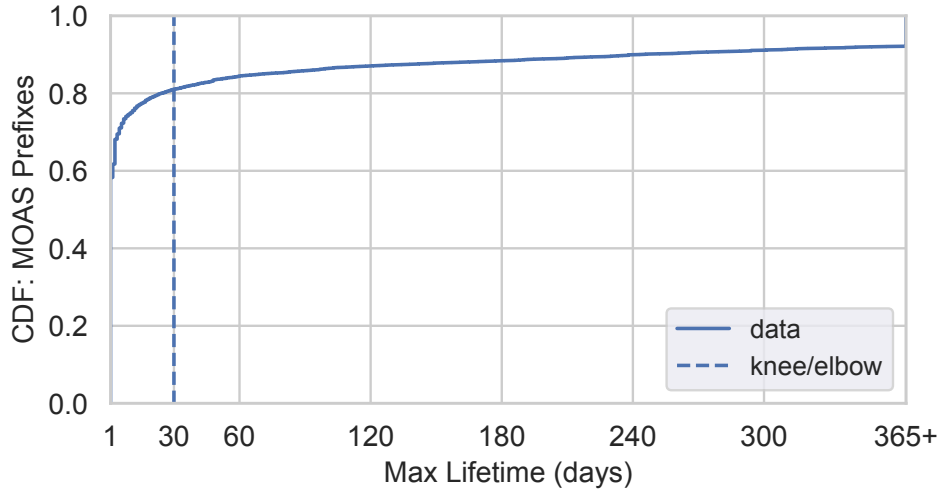

Figure 4.4: CDF of maximum lifetime of IPv4 and IPv6 MOAS prefixes capped at 365 days, vertical line shows the elbow/knee of the distribution.

4.2.3 Impact of Using More Than One RIB per Day

After having performed the lifetime analysis with various sensitivity values, we examine if increasing the number of RIB snapshots per day increases the lifetime of MOAS prefixes. Therefore, we use three RIB snapshots per day (3RIBs/24hours), as this is the maximum number of possible RIB snapshots with the same timestamp across both Routeviews and RIPE RIS RCs. We study the effect of increasing the number of RIB snapshots considering the following two scenarios.

First, we examine by how much the increased number of RIBs increases the maximum lifetime of MOAS prefixes. The left subplot of Figure 4.3 illustrates the comparison and effect of using three RIBs compared to one RIB per 24 hours on the maximum lifetime of MOAS prefixes. As can be seen, increasing the number of RIBs to three per day (blue line) does not substantially increase the maximum lifetime of MOAS prefixes compared to a single RIB per day, orange line. Furthermore, the number of MOAS prefixes visible throughout the complete six months remains similar. The higher share of MOAS prefixes with a maximum lifetime of fewer than five days in three RIBs compared to a single one is because we observe many more short-lived MOAS (e.g., MOAS appearing only once) when using three RIBs per day.

Second, we check if increasing the number of RIB snapshots increases the observability of MOAS prefixes. We define the observability of a MOAS prefix as the number of days out of the total days, when a prefix is observed as a MOAS. Therefore, the observability metric allows us to see how consistently prefixes are visible as MOAS. As shown in the right subplot of Figure 4.2, for nearly 80% of MOAS both one and three RIBs have more than 95% observability. This high value gives us confidence that using the maximum lifetime of a MOAS prefix is a good approximation of its actual lifetime. Moreover, with three RIBs we see a higher fraction of MOAS with 100% observability compared to one RIB. This is because with three RIBs we observe more MOAS prefixes that are visible only on a single RIB snapshot, which by definition have an observability of 100%.

In summary, increasing the number of RIBS from one to three per day does not increase

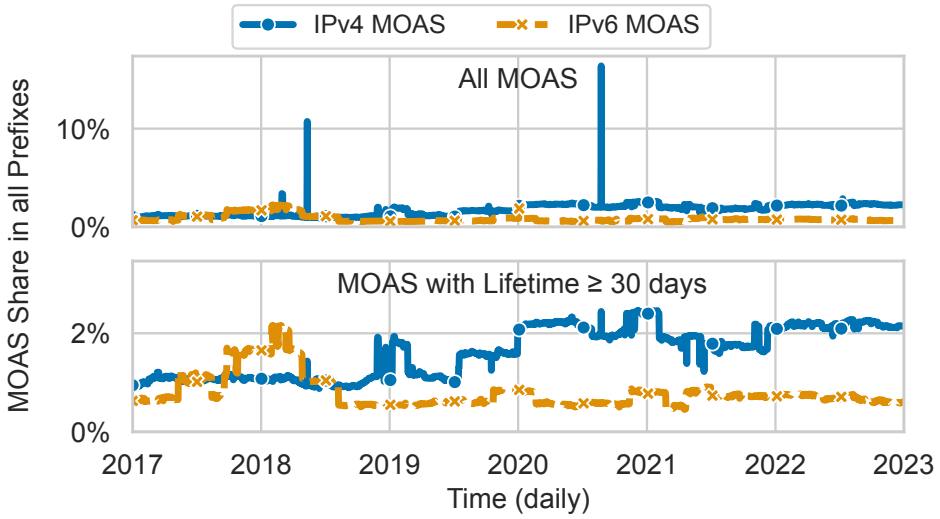

Figure 4.5: Fraction of MOAS prefixes out of all prefixes for IPv4 and IPv6 over time, all MOAS (top) and long-lived MOAS (bottom).

the lifetime or observability of MOAS prefixes, particularly long-lived ones. Instead, three RIBs per day add more short-lived MOAS, which are not the focus of this study. Therefore, in the remainder of the study for this chapter we use one RIB per day.

4.2.4 Identifying Long-Lived MOAS Prefixes

Previous studies have shown that prefix hijacks, route leaks, and configuration errors cause short-lived MOAS [28, 57, 146, 247, 297]. We strive to analyze long-lived MOAS prefixes in BGP. Therefore, we develop a technique to identify a threshold to distinguish between short-lived and long-lived MOAS prefixes.

We measure the maximum lifetime of all MOAS prefixes for our complete BGP dataset as observed from January 1, 2017, to January 1, 2023. Then, we use the mathematical Kneedle algorithm [243] to determine the "elbow" of the maximum lifetime of MOAS prefixes. The "elbow" point identifies the maximum curvature value within the lifetime of all MOAS prefixes and we use it as distinction point between short-lived and long-lived MOAS prefixes.

Figure 4.4 shows the maximum lifetime (capped at 365 days) for all MOAS prefixes. Using the Kneedle algorithm we identify 30 days as the elbow, which we therefore use as a threshold to distinguish between short-lived and long-lived MOAS prefixes. Generally, we find that more than 80% of MOAS prefixes are short-lived, whereas less than 20% have a lifetime of at least 30 days.

Now that we defined the threshold of minimum 30 days lifetime between short-lived and long-lived MOAS prefixes, we examine the overall share of MOAS prefixes in the BGP dataset. In Figure 4.5 we show the fraction of MOAS prefixes among all prefixes for the duration of six years for IPv4 and IPv6, respectively. The fluctuation in the number of IPv4 MOAS prefixes for the upper subplot is due to Angola Cables (AS37468) creating MOAS conflicts for around 90k prefixes, on May 12, 2018, leading to the 10% IPv4 MOAS peak. Moreover, on August 24, 2020, Huge Networks - DDoS Mitigation

(AS264409) creates MOAS conflicts for 143k prefixes, resulting in the 15% peak of IPv4 MOAS prefixes.

The lower subplot is limited to long-lived MOAS prefixes only, and exhibits less variation. We see the fraction of MOAS prefixes in IPv4 increasing from around 1% in 2017 to around 2% in 2023. In IPv6, the fraction is relatively stable at just below 1%, with a peak in February 2018. This peak is a result of GTT Communications (AS3257, AS5580) and DigitalOcean (AS14061, AS200130, AS202018, AS202109, AS393406, AS62567) each announcing hundreds of IPv6 hyper-specific [254] (i.e., /49–/128) MOAS prefixes.

In the remainder of the chapter we focus exclusively on long-lived MOAS prefixes. Therefore, we now use the term MOAS prefix interchangeably with long-lived MOAS prefix.

4.2.5 Limitations

While we try to address multiple caveats with our approach, it still has some limitations. Our observation of MOAS prefixes in the BGP dataset is limited to peer ASes of RCs. Therefore, our detected MOAS prefixes should be seen as a *lower-bound*, i.e., there might be more MOAS prefixes out there which are not visible at route collectors. As the vast majority—95% for IPv4 and 91% for IPv6—of MOAS prefixes are originated by the exact same set of origins throughout the six months, we do not give special treatment to MOAS prefixes with changing AS origins over time. Despite these limitations, we argue that our approach is suitable and our analysis provides valuable insights into the prevalence, characteristics, and use of MOAS prefixes in the Internet.

4.3 Prefixes and Origins

In this section we study MOAS prefixes and origins in detail: We analyze the RPKI status, CIDR sizes, and origin ASes of MOAS prefixes.

4.3.1 RPKI Status of MOAS Prefixes

As MOAS prefixes by definition are announced by at least two origin ASes, we now investigate whether a MOAS prefix has valid RPKI entries for its origin ASes. We examine the Route Origin Validation state of MOAS prefixes using the RPKI dataset [234] for all origins—i.e., for every prefix-origin (PO) pair—on a monthly basis, starting from January 2017 to January 2023. We classify the ROV state of MOAS prefixes as follows: (i) "All Valid Origins" if all PO pairs have a valid Route Origin Authorization record; (ii) "At Least One Valid Origin" if at least one PO pair of a MOAS prefix has a valid ROA record; (iii) "All Invalid Origins" if all PO pairs conflict with a present ROA record; (iv) "Not Found" if we do not observe any ROA for a MOAS prefix.

In Figure 4.6 we show the RPKI classification of MOAS prefixes over time. We note that the percentage of MOAS prefixes with "All Valid Origins" (shown in orange), has increased from less than 5% in 2017 to around 40% in January 2023. The increase in share of valid ROV states is in line with the RPKI deployment in all BGP prefixes [195]. It also underlines that a large part of MOAS prefixes are not due to prefix hijacks, route leaks, or router configuration errors, but are in fact intended for use. We see a peak in the beginning of 2019 for "All Valid Origins", which reaches around 45% and is

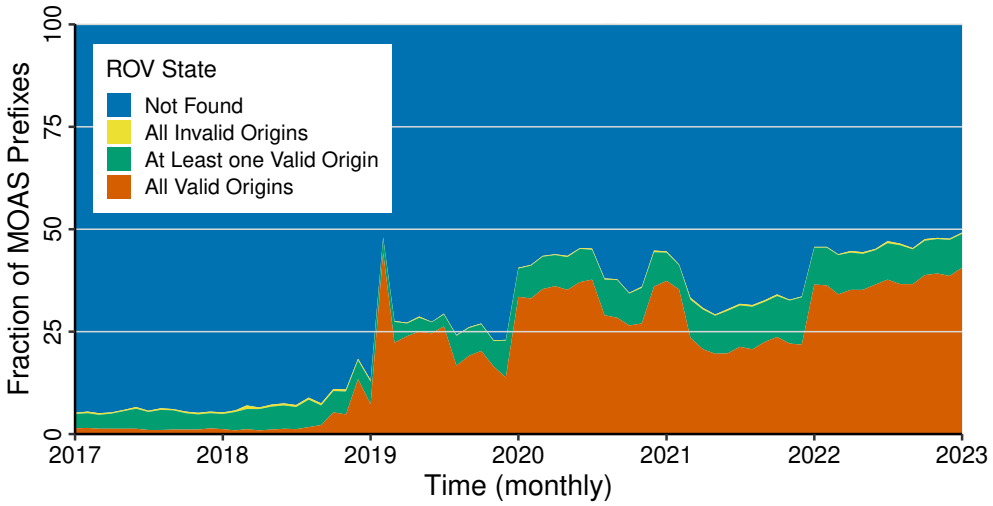

Figure 4.6: MOAS Route Origin Validation state in the RPKI dataset over time.

due to TTNet (AS47331) and Turk Telekomunikasyon (AS9121) announcing nearly 8k IPv4 MOAS prefixes—mostly /24 and /21–/23—as a result of their merger [72]. The share of MOAS prefixes with at least one valid origin (shown in green) has slightly increased from 2017 to 2023. Manually investigating the partially valid cases, we notice that not all origin ASes entered the prefix information properly in the RPKI database. We find cases where both ASes are registered with same organization name, with one having a valid and the other having an invalid ROV state for the MOAS prefix. Another possible reason for partially valid MOAS prefixes are router misconfigurations. The chance of them being hijacked prefixes is relatively small, as hijacks usually have short durations [26, 58, 138, 203, 217, 256, 257, 276] and our analyzed prefixes are seen for at least 30 days. MOAS prefixes with all invalid origins (shown in yellow), contribute less than 1% throughout our measurement period. The share of MOAS prefixes for which no ROA entry exists (shown in blue) decreases from 90% in 2017 to around 50% in 2023.

In summary, Around 40% of MOAS prefixes in January 2023 have all their origins validated in RPKI, with another ≈10% having a partially valid RPKI status. The case of all origins being invalid occurs for less than 1% of cases. This underlines that a large part of long-lived MOAS prefixes is likely intended for actual use and not linked to hijacking, misconfigurations, or route leaks.

4.3.2 CIDR Sizes

Prefixes in BGP are announced at different granularity levels, i.e., CIDR sizes. These CIDR sizes can provide insights into the usage of prefixes. Therefore, we analyze CIDR sizes of MOAS prefixes as shown in Figure 4.7 (IPv4 on the left, IPv6 on the right). For IPv4, we find that MOAS prefixes overall have increased from fewer than 10k in 2017 to more than 24k in 2023. The /24 CIDR size is most prominent in IPv4, followed by the /21–/23 CIDR size group. Hyper-specific [254] MOAS prefixes, i.e., /25–/32, constitute a tiny fraction of MOAS prefixes, as shown in blue.

While the number of MOAS prefixes with a CIDR size of /8–/20 is relatively stable over time, both the /24 and the /21–/23 range increase by more than 50% after mid-2019.

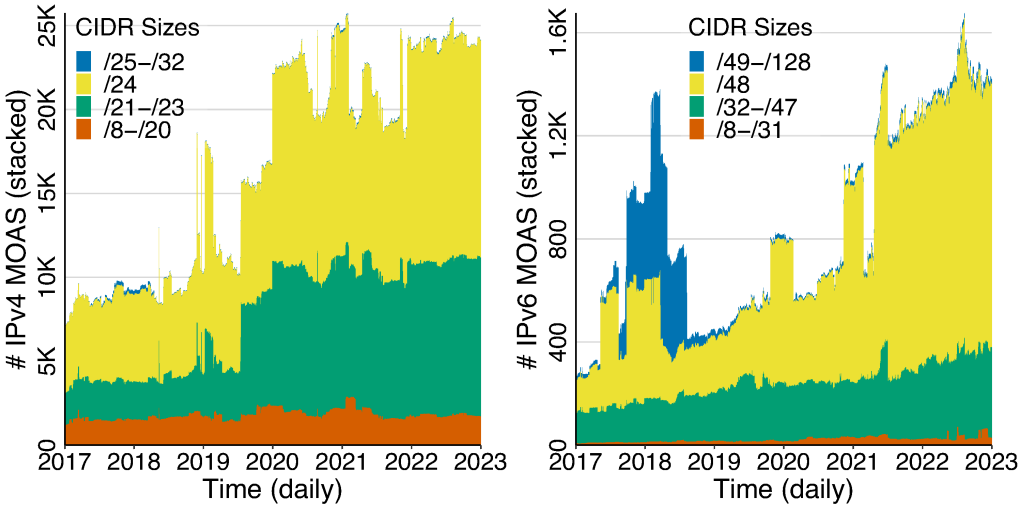

Figure 4.7: MOAS prefix CIDR sizes over time, for IPv4 (left) and IPv6 (right).

We investigate the peaks in the data and attribute them to the merger of TTNet and Turk Telekomunikasyon, see §4.3.1. These prefixes are visible as a substantial increase in the /21–/23 CIDR size group from August 2019 to January 2023. In addition, the peaks of /24 MOAS prefixes in July 2019, January 2020, September 2020, and January 2022 are associated with thousands of MOAS prefixes originating from Orange Spain (AS12479, AS12715). The latter of the two ASes used to belong to Jazztel, which was acquired by Orange in 2014 [4]. Moreover, the acquisition of Koninklijke KPN N.V. (KPN) International (AS286) by Global Telecom and Technology (GTT) (AS3257) in December 2019 [111] still results in 389 /24 MOAS prefixes in January 2023.

In IPv6 we see an increase from fewer than 400 to more than 1400 MOAS prefixes, as shown in the right plot of Figure 4.7. The number of /48 MOAS shown in yellow, which forms the largest CIDR size in IPv6, increases more than five-fold from 2017 to 2023. The CIDR size ranges /32–/47 and /8–/31 form the second and third-largest group of IPv6 MOAS prefixes and grow steadily. We see a large number of IPv6 hyper-specific MOAS prefixes, i.e., /49–/128 CIDR sizes shown in blue, from mid-2017 to mid-2018, which are normally not routable in the Internet [254]. These are again linked to GTT and DigitalOcean, see §4.2.4.

In summary, ASes use mostly fine-granular CIDR sizes—i.e., /24 for IPv4 and /48 for IPv6—to announce MOAS prefixes in the Internet, which allows for more fine-grained control of routing policies. Overall, the distribution of CIDR sizes within MOAS prefixes is similar to non-MOAS prefixes reported by BGP routing table [141]. Moreover, we find that a considerable number of MOAS prefixes are related to mergers and acquisitions of companies.

4.3.3 Origin ASes

Next, we analyze the origin ASes for every MOAS prefix in more detail. In the left subplot of Figure 4.8 we observe that more than 95% of IPv4 and around 88% of IPv6 prefixes have only two origin ASes (i.e., the minimum required for them to be classified as a MOAS prefix). Moreover, around 1% of IPv4 and 8% of IPv6 MOAS prefixes have

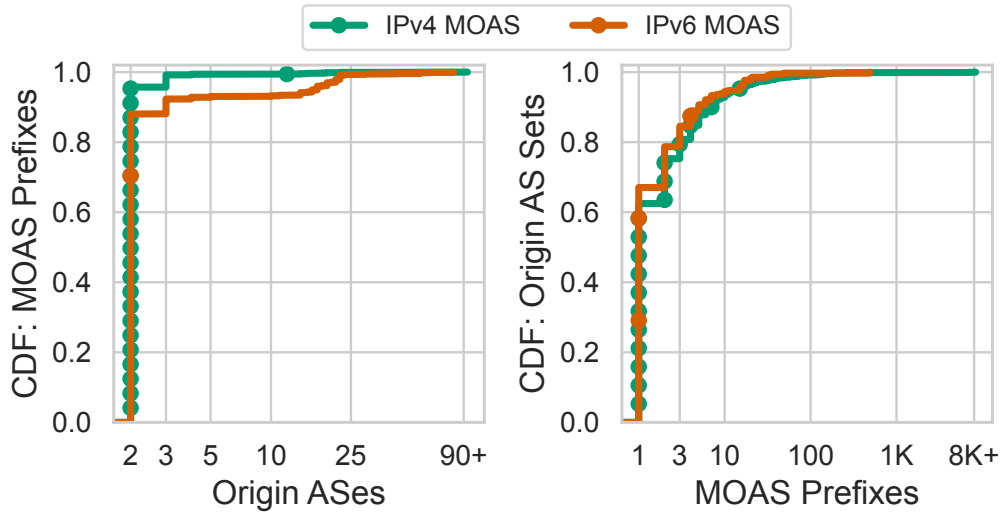

Figure 4.8: Number of origin ASes per prefix (left) and the number of originated MOAS prefixes per origin AS set (right) for January 1, 2023.

more than three origin ASes. We notice that a few prefixes are originated by more than 80 origin ASes. We manually investigate these cases and identify Verisign—a DNS provider which also operates two DNS root servers—to have the highest number of origin ASes for one IPv4 and one IPv6 prefix.

Furthermore, we analyze the number of MOAS prefixes originating from the same set of origin ASes, so-called origin AS sets. In Figure 4.8 the right plot illustrates that more than 60% of origin AS sets announce only a single MOAS prefix and around 90% announce fewer than ten MOAS prefixes. In addition to January 1, 2023, we also check one day per month and observe a similar pattern. Moreover, only a few origin AS sets announce large numbers of MOAS prefixes. Investigations show that this is again the TTNet and Turk Telekomunikasyon merger case, see §4.3.1.

In summary, The vast majority (95% for IPv4, 88% for IPv6) of MOAS prefixes are originated by two origin ASes, with some outlier prefixes being announced by more than 80 origins. Around 90% of origin AS sets announce fewer than ten MOAS prefixes. Therefore, the typical case is that two origin ASes announce a small number of MOAS prefixes.

4.4 Visibility

In this section we analyze the visibility of MOAS prefixes across route collector peers and investigate the minimum and maximum visibility per prefix in detail.

4.4.1 Visibility Across Route Collector Peers

First, we focus on the visibility of MOAS prefixes in route collector peers. RC peers are the ASes that feed routing information to the route collector via BGP sessions. Figure 4.9 illustrates the visibility of every MOAS prefix-origin pair, for IPv4 (left plot) and IPv6

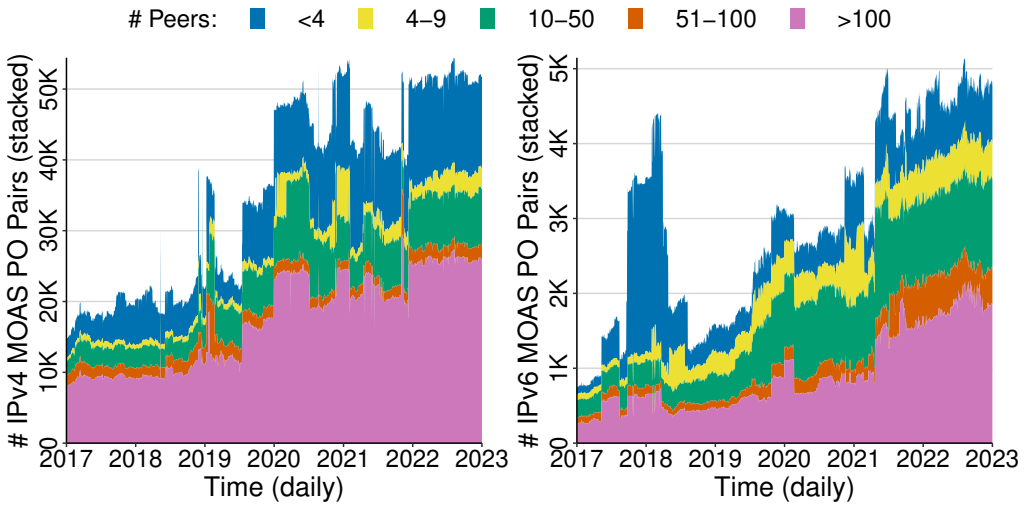

Figure 4.9: MOAS prefix-origin pair visibility across route collector peers over time, for IPv4 (left) and IPv6 (right).

(right plot), across RC peers for the period of the study. For IPv4, the largest group of MOAS PO pairs (i.e., around 50%), is visible in more than one hundred route collector peers (shown in pink), followed by MOAS PO pairs with a visibility of 3 or fewer peers (shown in blue). This "bimodal" visibility distribution is consistent with overall BGP prefixes [276]. PO pairs with a visibility between 10 and 50 peers (shown in green) form the third largest group.

In IPv6 we see a similar picture, with MOAS PO pairs with visibility >100 being most common, followed by PO pairs with visibility between 10 and 50 peers. The <4 category forms the third-largest group during our study period. However, from mid-2017 to mid-208, we see a visible spike in IPv6 to around three thousand PO pairs in the <4 category. We investigate this peak manually and find that it is largely due to hyper-specific MOAS prefixes originated by GTT Communications and DigitalOcean (see §4.2.4) and /48 MOAS prefixes originated from the Thai eyeball network Realmove Company Limited (AS132061, AS7470).

In summary, PO pairs of IPv4 MOAS exhibit a "bimodal" RC visibility behavior: Either they are widely visible in more than a hundred RC peers or very poorly visible in fewer than four RC peer. For IPv6 we see a similar picture, although semi-visible PO pairs are more common than in IPv4.

4.4.2 Minimum and Maximum Visibility

To better understand the bimodal visibility behavior of MOAS prefixes, we investigate the minimum and maximum visibility (i.e., number of route collector peers) of every MOAS prefix depending on its origin AS. This allows us to see if a MOAS prefix has a similar visibility for all its origins or if the minimum and maximum are widely spread. In Figure 4.10 we show the minimum and maximum visibility of each PO pair for IPv4 (top) and IPv6 (bottom) for a single day (i.e., January 1, 2023). We also check one day per month for the entire period of study, and observe a similar pattern.

For the maximum visibility we observe that for 99% of all MOAS prefixes—in both IPv4

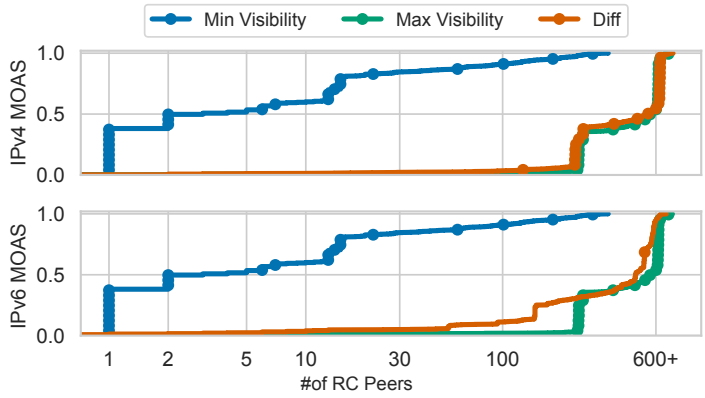

Figure 4.10: Minimum, maximum, and difference visibility of MOAS prefixes on January 1, 2023 for IPv4 (top) and IPv6 (bottom).

and IPv6—at least one PO pair is visible by more than 100 RC peers. Moreover, for more than 50% of MOAS prefixes, at least one PO pair has a maximum visibility larger than 500 route collector peers. We attribute the steady increase in maximum visibility between 100 and 500 RC peers and after 500 RC peers to partial and full BGP feeds [102], respectively.

Analyzing the minimum visibility of MOAS prefixes shows that for around 40% of IPv4 and 20% of IPv6 MOAS prefixes, at least one PO pair is visible only at a single route collector peer. For over half of IPv4 MOAS prefixes, at least one PO pair has a minimum visibility of fewer than five RC peers, and more than 90% are visible on fewer than 100 RC peers. Around 70% of IPv6 MOAS prefixes have at least one PO pair with a minimum visibility of fewer than 100 RC peers.

This indicates, that each MOAS prefix is highly visible from one origin AS and has a very low visibility from another origin AS. To confirm this, we calculate the difference between the maximum and minimum visibility *for each MOAS prefix*. As can be seen in Figure 4.10, the difference distribution is very close to the maximum visibility, underlining that indeed for each MOAS prefix one origin is very visible whereas the other is barely visible.

In summary, Each MOAS prefix has one origin, with which the prefix is very well visible at 100 or more RC peers and another origin which is barely visible at all. This hints at MOAS not being mainly used for anycast purposes, as for these we would expect a more balanced visibility distribution.

4.5 Users and Usage of MOAS Prefixes

In this section we present an analysis on the users and usage of MOAS prefixes in the Internet. Specifically, we first examine the BGP relationships between the origin ASes of MOAS prefixes. Furthermore, we investigate the business relationships among MOAS prefix users and look into the use of MOAS prefixes by Hypergiants. Finally, we evaluate to which extent MOAS prefixes are being utilized as anycast services.

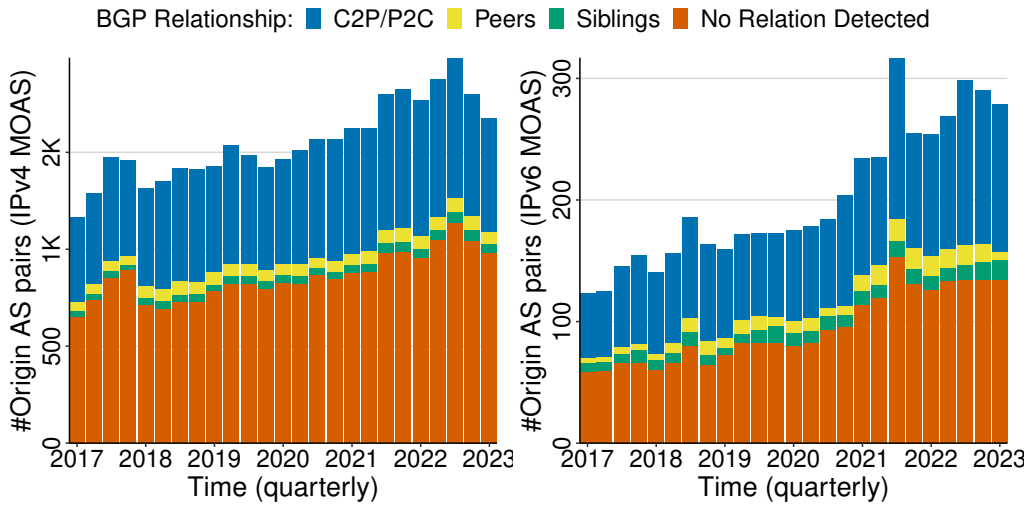

Figure 4.11: BGP relationship of MOAS origin AS pairs for IPv4 (left) and IPv6 (right).

4.5.1 BGP Relationship of MOAS Origin ASes

First, we want to better understand the BGP relationship between origin ASes of MOAS prefixes. We use quarterly snapshots of CAIDA's AS to organization [39] and AS classification inferences [37] datasets to identify the inter-AS BGP relationship of MOAS origin ASes. For simplicity, we focus on MOAS prefixes with two origins. These prefixes make up around 90% of all MOAS prefixes and two origin ASes simplify our analysis, as we do not have to consider ternary or higher order relationships.

We check for the following three categories of BGP relationships for MOAS prefixes. First, if both origin ASes have the same organization name in the dataset, we call them "Siblings". Second, if any customer-to-provider or provider-to-customer relationship between the origin ASes of MOAS prefixes exists, we classify the inter-origin-AS relationship as "C2P/P2C". Next, we examine if both origin ASes fall into the "Peering" category. Finally, if we do not observe any of the above known BGP relationships between the origin AS of a MOAS prefix, we mark the inter-origin-AS BGP relationship as a "No Relation Detected".

Figure 4.11 illustrates the inter-origin-AS BGP relationship of MOAS prefixes for IPv4 (left) and IPv6 (right) over time. We find that almost half of all origin AS pairs for both IPv4 and IPv6 are classified as C2P/P2C (shown in blue). Peering relationships (shown in yellow) are less frequent, with siblings making up the smallest part of MOAS origin AS pairs. We do not detect any relationship using the CAIDA datasets for around half of all origin AS pairs. Overall, the distribution of categories remains relatively stable over time. Furthermore, when using the number of PO pairs as a metric (not shown), we see a similar trend for the share of relationship categories as in Figure 4.11, with a comparatively higher share of "Siblings" for two snapshots around 2020.

In summary, The vast majority of detected MOAS origin AS relationships fall into the customer to provider category. This shows, that many MOAS prefixes are not related to sibling ASes, rather a provider announces the same prefix as the customer.

4.5.2 Relationship and Business Type of MOAS Origin ASes

To shed more light on the business relationship between MOAS origin AS pairs, we now apply the 17 top-level ("layer 1") categories of the ASdb dataset [298] on the most recent snapshot of MOAS prefix origin ASes (January 1, 2023). We also check MOAS origins in all available ASdb datasets for the past two years and observe a similar pattern.

As in the previous analysis, we focus on MOAS prefixes with exactly two origin ASes, i.e., 2618 unique origin ASes in total. For 97.2%, we find a matching business category in ASdb, out of which 77.6% match a single, and 23.4% origin ASes match more than one category. For easier understanding, we focus our analysis on ASes with one business category.

Figure 4.12 shows a heatmap of prevalent business categories for MOAS origin AS pairs. The number and color of every cell within the heatmap represents the prevalence of each category pair for MOAS origins. Examining the heatmap cells, we notice that in roughly 40% of the cases, both MOAS origins fall into the "IT" category. Furthermore, we notice a high number of IT company pairs with other business types, such as "Service", "Retail", or "Media". This indicates that in addition to advertising their own prefix, organizations leverage IT companies to re-advertise prefixes, resulting in a MOAS event. Moreover, for every business type, we see an elevated level on the diagonal line, which means that ASes originate MOAS prefixes from the same business type. Overall, 89% of MOAS prefixes have at least one origin AS classified as an IT company. To exclude a possible bias towards IT companies, we analyze the prevalence of the IT category in the ASdb dataset. We find that MOAS origins from the IT category have a 10 percentage points higher share than general origin ASes in the ASdb dataset (56% vs. 46%).

In summary, MOAS prefixes are mostly originated by IT companies, with the same company type for both origins being most common.

4.5.3 Hypergiants Using MOAS Prefixes

Next, we investigate to what degree Hypergiants make use of MOAS prefixes. Hypergiants are large content providers, CDNs, and cloud networks (e.g., Google, Amazon, Meta) with a heavy-outbound traffic profile [162]. We use the list of 16 Hypergiants by Gigis et al. [98] and find that 11 of them (i.e., Akamai, Alibaba, Amazon, CDNetworks, Cloudflare, Facebook, Google, Incapsula, Limelight, Netflix, and Verizon) announce a total of 313 MOAS prefixes. The top three Hypergiants making use of MOAS prefixes are Verizon (98 MOAS prefixes), Netflix (48), and Google (45). This shows that MOAS prefixes are used by big players in the Internet, possibly to improve their network's resilience, performance, and quality of experience for their users.

In summary, 11 out of 16 Hypergiants use MOAS prefixes to reap the benefits of their users being able to access their networks via multiple origins.

4.5.4 Anycast MOAS Prefixes

MOAS prefixes can be good candidates for anycast services as they originate from multiple ASes. However, we find that only 225 (0.9%) of IPv4 and 89 (6.3%) of IPv6 MOAS prefixes are identified as anycast prefixes [25].

We observe that the number of origins per prefix for anycast MOAS prefixes is higher than for general MOAS prefixes: While around 90% of MOAS prefixes have two origin

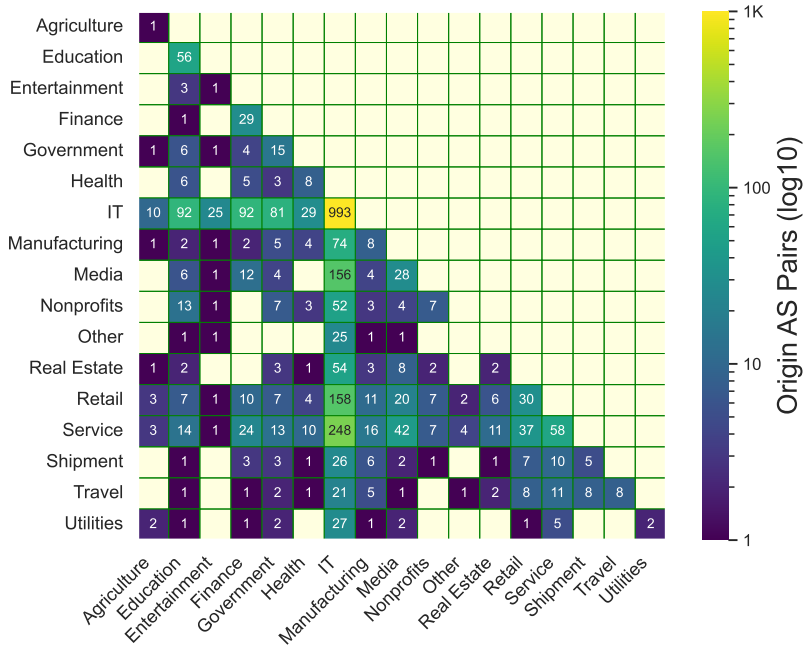

Figure 4.12: Heatmap of business type relationships between MOAS origin pairs.

ASes, this percentage is less than 40% for anycast MOAS prefixes. More than 50% of anycast MOAS prefixes are announced by more than ten origin ASes. The prefix with the most origin ASes (61 origins) is the IPv6 prefix `2001:503:231d::/48`, which is attributed to Verisign.

Even though the official root-server website lists single origin ASes for each of the 13 root DNS servers [136], we observe that the IPv4 address of A and J root DNS servers, operated by Verisign, use MOAS prefixes with a /24 CIDR size. In contrast to IPv4, we do not find any IPv6 MOAS prefix for DNS root servers.

In summary, the use of anycast within MOAS prefixes is low, with under 1% for IPv4 and around 6% for IPv6. When MOAS prefixes are used for anycast, more than 50% of them have more than ten origins. This shows, that MOAS prefixes are rarely used for anycast purposes, but if they are, they use a large number of origin ASes.

4.6 Related Work

In this section we present related work in the areas of prefix hijacking and MOAS prefixes.

Prefix Hijacking: When an attacker announces someone else's prefix in BGP from its own AS, the prefix is being hijacked. A prefix hijack can cause service interruptions and consequently lead to traffic, financial, and reputation loss for the victim. The first well-known prefix hijacking case happened in 2008 when Pakistan Telecom hijacked the YouTube prefix for censorship purposes [228]. Since then multiple prefix hijacking events have occurred. A more recent example of such an incident happened in August 2022 when several crypto services hosted on AWS were hijacked, resulting in financial loss [105]. Previous work has suggested various techniques and tools to detect and

mitigate BGP prefix hijacks [26, 138, 203, 217, 257, 276]. Websites such as BGPmon [24] and CAIDA's Hijack Observatory [38] are monitoring the BGP for possible prefix hijack events. In 2018, Sermpezis et al. conducted a survey among 75 network operators which showed that operators mostly rely on third parties such as BGPmon [24] to detect prefix hijacks [256]. In 2019, Cho et al. presented a classification of BGP hijacks into different categories and found that 4%, 1%, and 2% of hijacks are due to typos, prepend mistakes, and BGP hijacking with a forged AS path, respectively [58].

MOAS Prefixes: In 2001, Zhao et al. investigated MOAS prefixes for the first time on the Internet using Routeviews route collectors [281] finding that the majority of MOAS conflicts were short-lived, lasting only a few days [297]. In 2007, Chin analyzed MOAS prefixes observed over 21 days and concludes that MOAS constitute a small but growing percentage of reachability information [57]. In 2011, Bornhauser et al. present MOAS Analyzer, a tool to identify, analyze, and automatically classify MOAS conflicts [28]. In 2014, Jacquemart et al. performed a longitudinal analysis of MOAS prefixes and showed that short-lived MOAS events are not due to misconfigurations, but because of origin instability or route flapping [146]. In 2015, Schlamp et al. investigated subprefix MOAS events using IRR data, topology data, and Transport Layer Security (TLS) certificates and found that the majority of these announcements with multiple origins are harmless [247].

To the best of our knowledge, this is the first work to perform an in-depth longitudinal analysis of long-lived MOAS prefixes.

4.7 Summary

In this chapter we analyzed long-lived MOAS prefixes over a period of six years. In IPv4 we identified around 24k long-lived MOAS prefixes compared to 1.4k in IPv6. We found that the vast majority of MOAS prefixes with entries in RPKI have a valid ROV state for all origins. Moreover, we found MOAS prefixes to use very specific CIDR sizes, indicating their use for fine-grained traffic steering. We identified that a considerable number of MOAS prefixes are related to mergers and acquisitions of companies. We recommend that network operators should clean up the extra MOAS produced due to mergers or other situations to prevent unnecessarily routing entries in the Internet routing environment. Furthermore, over 90% of MOAS prefixes were announced by two origin ASes, with the majority of origin pairs being in a customer-provider relationship. Finally, we have shown that the majority of MOAS users are Information Technology (IT) companies and that MOAS are rarely used for anycast purposes.

Chapter 5
Sibling Prefixes: Identifying Similarities in IPv4 and IPv6 Prefixes

In this chapter, we study similarities between IPv4 and IPv6 prefixes that suggest operational or administrative correlation between the two address families. Since the beginnings of the standardization of IPv6 in 1998 [74], the traditional IPv4 Internet and the newer IPv6 Internet have existed side-by-side. Initial deployment of IPv6-enabled services as well as clients has been slow, taking up to 2013 to reach 1% of Internet traffic [70]. With the exhaustion of the IPv4 address space [214, 227] and the preference for IPv6 over IPv4 in connection establishments [246, 290], IPv6 sees continuous growth, with more than 219,000 IPv6 prefixes in the global routing table in November 2024 [213], more than 75% of top 1k popular domains being accessible over IPv6 in August 2022 [272], and more than 45% of clients accessing Google via IPv6 in November 2024 [107].

This continuing growth in IPv6 requires a better understanding of the IPv4 and IPv6 interdependencies. One such interdependency is siblings, i.e., a connection between an IPv4 and IPv6 address belonging to the same host. The identification of such connections, i.e., sibling detection has been widely studied in the past [8, 9, 21, 22, 245]. To date, however, siblings have only been analyzed on an IP address level.

In this chapter, we strive to push the boundaries of sibling detection and extend their reach from IP addresses to IP prefixes. We introduce the concept of a *sibling prefix* as an IPv4 and IPv6 prefix pair with similar services (i.e., IPv4 and IPv6 prefixes sharing similar domain names). We present a technique to detect these sibling prefixes by leveraging large-scale DNS resolution results. Identifying sibling prefixes allows us to better understand interdependencies between the IPv4 and IPv6 Internet in several ways and is the first step towards a more comprehensive understanding of differences in network performance, policy, geolocation and security posture of services running concurrently on both IPv4 and IPv6. Sibling prefixes can show shared networking infrastructure, highlight the presence of backup paths (or lack thereof) between IPv4 and IPv6, and shed light on the deployment and management of dual-stack services in different networks. Moreover, identification of sibling prefixes can help network operators to make more informed and inclusive routing decision, by considering both

IPv4 and IPv6 as part of sibling prefixes. The inconsistent treatment of IPv4 and IPv6 traffic, especially in dual-stack environments, might lead to operational inefficiency or security vulnerabilities. Identifying sibling prefixes can thus help network operators apply consistent networking policies across both IP address families. For instance, network operators might want to prioritize, filter, or block traffic/domains of IPv4 prefixes, and identified sibling prefixes allows to do this for the IPv6 counterpart as well. Furthermore, service operators can leverage sibling prefixes to apply knowledge learned about a prefix for one IP version to the other IP version. One example are geolocation database providers using sibling prefixes to transfer geolocation information from IPv4 to IPv6 using sibling prefixes, thus improving geolocation across IP version boundaries.

Our main contributions in this part of dissertation are as follows:

- **Sibling prefix detection methodology:** We present a novel technique to detect sibling prefixes and thoroughly evaluate their suitability. We validate the sibling prefix relationship using the 2200 RIPE Atlas probes, 260 globally distributed dual-stack virtual private servers, and data from port scan results, see §4.2.
- **Tuning sibling prefix sizes:** We develop the sibling prefix tuner (SP-Tuner) algorithm that searches and derives new pairs of sibling prefixes from existing ones with a higher Jaccard similarity, by looking for more suitable CIDR sizes than the ones observed in Internet routing. The SP-Tuner algorithm increases the percentage of sibling prefixes with Jaccard similarity value of 1 (i.e., perfect match siblings) from 52% to 82%, see §5.2.3 and §5.2.4. We publish data and code of our detection methodology at sibling-prefixes.github.io.
- **Large-scale sibling prefix analysis:** We perform large-scale sibling prefix measurements using data from IPv4 and IPv6 DNS resolutions, finding around 46k unique IPv4 and almost 39k unique IPv6 prefixes, resulting in more than 76k sibling prefix pairs, of which more than half share the same organization names for their IPv4 and IPv6 origin ASes. We also conduct longitudinal measurements and find that sibling prefixes are relatively stable over time. While IT organizations have the highest number of sibling prefixes, we further identify sibling prefixes in 24 hypergiants and CDNs. Moreover, we use RPKI data to analyze sibling prefixes' Route Origin Validation status, finding over 60% of sibling prefix pairs having at least one valid origin AS, see §5.3.
- **Impact of sibling prefixes:** We discuss the potential uses and impact of sibling prefixes, highlighting the need for tailored prefix size choices, and drawing attention to a paradigm shift from IP addresses to domains. Finally, we plan to regularly publish a list of sibling prefixes for use by fellow researchers and network operators, see §3.6.

5.1 Datasets

This section provides details on datasets we use for identifying and characterizing sibling prefixes.

5.1.1 DNS Dataset

One of the cornerstones of our methodology to detect sibling prefixes relies on shared domain names between IPv4 and IPv6 prefixes. We leverage domain names from

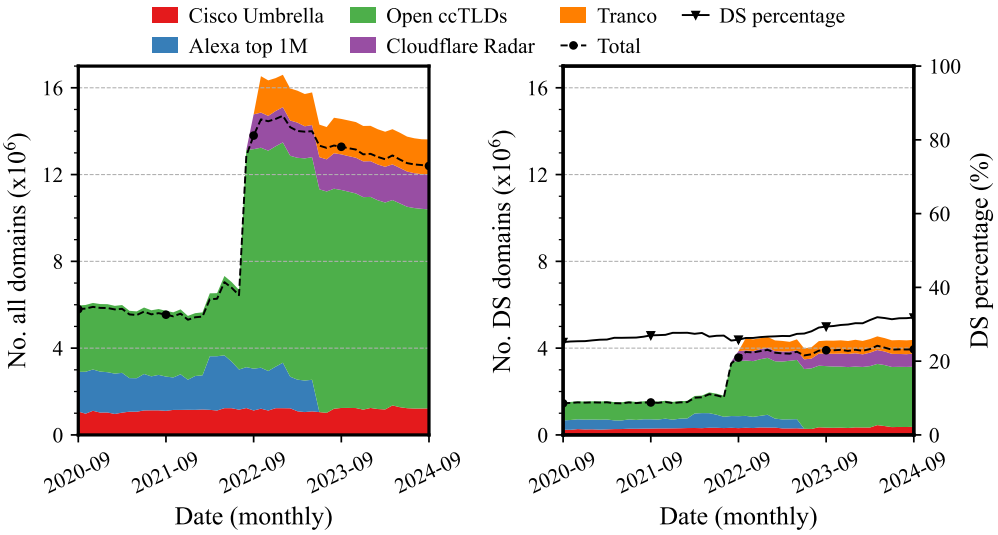

Figure 5.1: Number of all domains (left) and dual-stack (DS) domains (right) over time in the OpenINTEL dataset.

the OpenINTEL DNS dataset [204] to obtain DNS resolution results from large-scale measurements. The OpenINTEL dataset consists of DNS resolution results of the Alexa top 1M [10], Cloudflare Radar [65], Tranco [285], Cisco Umbrella [62], and open ccTLD domains. We collect four years of OpenINTEL data on every second Wednesday of each month from September 2020 to September 2024, resulting in 49 snapshots.

Figure 5.1 shows the longitudinal evolution of the OpenINTEL dataset over time. The left subplot shows the overall number of domains in the dataset, whereas the right subplot focuses on DS domains DS domains are the relevant subset of domains for identifying sibling prefixes, as we rely on shared IPv4-IPv6 domains within sibling pairs.

As can be seen, the overall number of domains as well as DS domains in the dataset has been growing over time. This growth is mainly due to additional datasets being added to OpenINTEL, such as Tranco in September 2022 and Cloudflare Radar October 2022. The largest increase in the total number of domains occurs in August 2022, which corresponds to the inclusion of ".fr" Top-Level Domains (TLDs) 6.35M when it was added to the set of open ccTLDs [268]. On the contrary, the removal of the Alexa top 1M dataset in May 2023, leads to a slight decrease in the number of domains. Furthermore, the percentage of DS domains is also slightly increasing over time from 25.2% in September 2020 to 31.8% in September 2024. This highlights the increasing deployment of dual-stack domains and, therefore, the need to identify these deployments using sibling prefixes. In the remainder of this chapter, we leverage 3.95 million unique DS domains to identify sibling prefixes.

5.1.2 IP to Prefix and AS Dataset

The OpenINTEL dataset provides prefix and AS information for A and AAAA DNS records, but around 1% of records lack prefix or AS information. To address this, we use data from the Routeviews project [240] to identify the missing prefix and AS information. For less than 0.01% of DS domains, we observe private, invalid, or reserved IP addresses, which we discard and do not consider for further analysis. The latest snapshot of the

OpenINTEL dataset (September 11, 2024) contains 271.5k IPv4 addresses and 978.4k IPv6 addresses, which map to 24.1k unique IPv4 prefixes across 7.5k ASes and 12.4k IPv6 prefixes across 7.6k ASes, respectively. We use the extracted IP, prefix, and AS information throughout this chapter.

5.1.3 AS to Organization Mapping Datasets

For all ASes of sibling prefixes, we identify their organization using the AS organization dataset from Chen et al. [56]. The dataset also allows us to identify sibling ASes, i.e., ASes maintained by the same organization. Due to the recency of this dataset, we use CAIDA's AS to organization mapping dataset [41] for the analysis prior to October 2022, and the Chen et al. [56] dataset for the analysis from October 2022 onward. We use the AS to organization datasets throughout our analysis to provide insights on the status of sibling prefixes for the same organization compared to the rest of the data. We provide further details on sibling prefixes from the same organization in §5.3.5.

5.1.4 Hypergiants and CDN Datasets

We use the list of Hypergiant (HG) networks provided by Böttger et al. [30] and Gigis et al. [98], and the list of Content Distribution Networks [53] to classify sibling prefixes as HG, CDN, and non-CDN-HG. Using the HG and CDN datasets, we analyze the similarity status for sibling prefixes of CDN-HG versus non-CDN-HG in §5.4.1.

5.1.5 ASdb Datasets

To infer the business type of the origin ASes for sibling prefixes, we use the ASdb dataset [298]. The dataset classifies ASes into one or more of 17 business categories, such as "Computer and IT", "Government", or "Media". This dataset allows us to analyze the prevalence of different business types among sibling prefixes (see §5.4).

5.1.6 RPKI Dataset

We examine the validity of each pair of sibling prefixes in the Resource Public Key Infrastructure. For this purpose, we download the RPKI data of all five Regional Internet Registries [234] from September 2020 to September 2024 for every month. With the RPKI data, we identify the Route Origin Validation state for sibling prefix pairs. We use the RPKI dataset in §5.4.2.

5.1.7 Port Scan Dataset

We scan a set of 14 well-known ports [19, 83, 244] on all IP addresses of sibling prefixes. We report their responsive status and Jaccard similarity by comparing them to the Jaccard similarity observed in the OpenINTEL data §5.2.7.

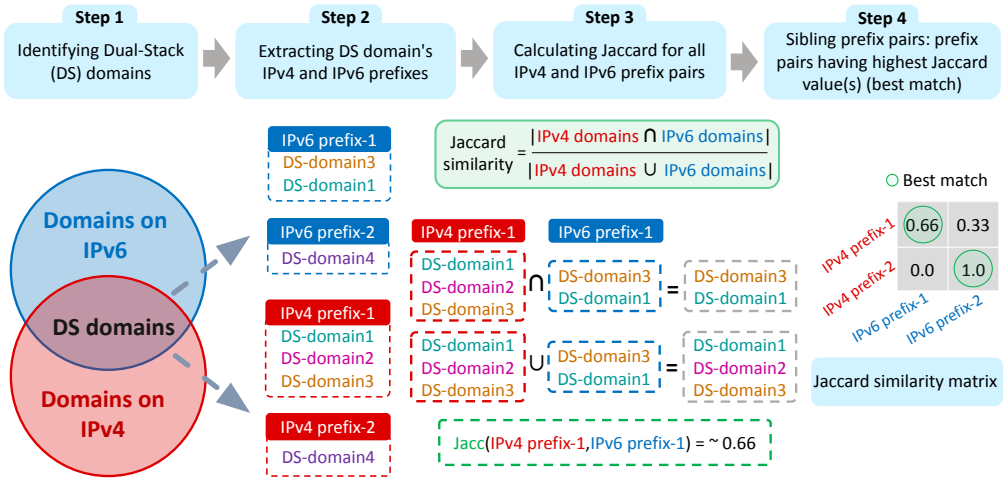

Figure 5.2: Methodology to identify sibling prefixes using Jaccard similarity index.

5.2 Methodology to Detect Sibling Prefixes

This section explains our methodology to identify sibling prefixes. We use the mapping of a domain name to an IPv4 and IPv6 address as an indicator for running services, which can then be used to infer sibling prefixes.

As we are interested in the actual domain that maps to an IP address, we use the domain name provided in the DNS response instead of the queried domain. This helps to overcome cases where the domain name in the response is different from the domain that was queried, especially for DNS queries resulting in CNAME responses. Figure 5.2 provides an overview of our methodology to identify sibling prefixes.

5.2.1 Identifying Sibling Prefixes

As shown in Figure 5.2, identifying sibling prefixes consists of four steps: (1) Identifying dual-stack domains, (2) extracting DS domain's IPv4 and IPv6 prefixes, (3) calculating similarity metrics for all prefix pairs, and (4) selecting sibling prefixes. We detail each of these steps as follow.

Step 1: Identifying dual-stack domains. In the first step, we examine the mapped IP addresses of all domain names in our dataset. If the domain name maps to a CNAME, we follow the CNAME chain until we reach the final IP address in the CNAME chain.

This results in two different sets of domains, all domains resolving to IPv4 addresses and all domains resolving to IPv6 addresses, as illustrated with red and blue circles in Figure 5.2. The intersection of domain names in these two sets are domain names with both IPv4 and IPv6 addresses, i.e., DS domains. We use only these DS domains in our sibling prefix detection technique.

Step 2: Extracting DS domain's IPv4 and IPv6 prefixes. In the second step, we identify the prefixes for all IP addresses of DS domains. We use the prefix information present in the OpenINTEL dataset (see §4.1) for every IP address of each DS domain. We then group the domains by IPv4 and IPv6 prefixes to calculate the similarity between an IPv4 and an IPv6 prefix based on the set of DS domains.

Step 3: Calculating Jaccard value for all IPv4-IPv6 prefix pairs. In the third step, we calculate the Jaccard similarity index [145] for IPv4-IPv6 prefix pairs by comparing the set of domains resolving to each pair. We denote all DS domains of an IPv4 prefix as set A and all DS domains of an IPv6 prefix as set B. Then, the Jaccard similarity index for sets A and B can be calculated as Equation (1):

$$\text{Jaccard}(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

Note that we also investigate alternative similarity metrics, see §5.2.2.

Step 4: Sibling prefix pairs. For each prefix pair, we get a Jaccard similarity value between 0 and 1, with values close to 1 indicating high similarity between two prefixes and values close to 0 indicating low similarity. In the final step, we select the prefix pairs with the highest Jaccard value, i.e., the "best match". Prefix pairs with a similarity value of 0 are discarded, and if multiple prefix pairs share the same highest Jaccard value, we keep all of them. We refer to these best matches as sibling prefix pairs for the corresponding IPv4 and IPv6 prefixes. These are the sibling prefixes we analyze in-depth in the remainder of this chapter.

5.2.2 Examining Possible Similarity Metrics

Selecting a suitable similarity metric is crucial for identifying sibling prefixes. We compare the overlap coefficient [275] and Dice coefficient [269] with the Jaccard similarity index [145].

The overlap coefficient measures the overlap between two finite sets. For two sets A and B, the overlap coefficient can be calculated using the intersection of sets A and B divided by the smaller size of the two sets. To calculate the overlap coefficient (OC) for sets A and B we use Equation (2):

$$OC(A, B) = \frac{|A \cap B|}{\min(|A|, |B|)} \tag{2}$$

The Dice coefficient is used to measure the similarity of two samples. It is often used to compare the similarity of two texts or sequences of words. The Dice coefficient for sets A and B can be calculated as shown in Equation (3):

$$\text{Dice}(A, B) = \frac{2 \times |A \cap B|}{|A| + |B|} \tag{3}$$

Next, we identify sibling prefixes by applying the Jaccard similarity index, overlap coefficient, and Dice coefficient on DS domains and compare the results. Figure 5.3 shows the results of similarity values for the three metrics. With the overlap coefficient more than 90% of sibling prefix pairs have a similarity value of 1.0. If either DS domain of the IPv4 prefix is a subset of the IPv6 prefix, or vice versa, the overlap coefficient value will be by definition 1. This property makes the overlap coefficient unsuitable for our study; as explained, we are interested in similar prefixes, not overlapping prefixes. The Dice and Jaccard lines in Figure 5.3 are relatively similar, having a similarity value of 1 for around 50% of sibling prefix pairs. However, the similarity values below 1 have slightly different similarity values, with Dice being lenient to the right side and having higher similarity values. This is because the Dice coefficient is generally more sensitive

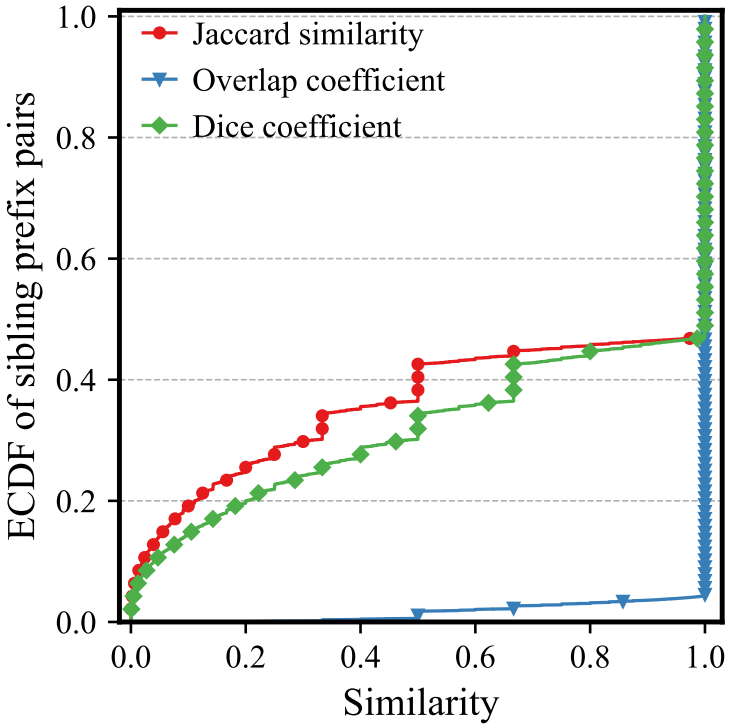

Figure 5.3: Comparison of Jaccard, Dice, and overlap coefficient similarity metrics.

to slight overlaps in sets [269]. On the other hand, the Jaccard index provides less biased and more balanced similarity results for variable set sizes [89]. In our study, as IPv4 and IP6 prefixes tend to have differently sized domain sets (e.g., sometimes a couple of domains in one set compared to dozens in the other set), we choose the Jaccard index as a suitable similarity metric to identify sibling prefixes.

5.2.3 Sibling Prefix Tuner (SP-Tuner) Algorithm

We use IP prefixes from BGP announcements and domains to identify sibling prefixes. The CIDR sizes of these sibling prefix pairs, derived from BGP announcements, can occasionally be too specific. Conversely, achieving better Jaccard similarity may be possible by changing the CIDR size of prefixes to cover more similar sets of domains. We introduce the sibling prefix tuner (SP-Tuner) algorithm to improve the similarity (i.e., Jaccard similarity value) and fine tune the CIDR size of sibling prefixes.

We investigate the effects of reducing and increasing prefix sizes on Jaccard similarity of sibling prefixes. Two possible approaches are to either check for the Jaccard similarity of less specific or more specific of the current sibling prefix pairs. For example, to examine the covering /22 prefix of a BGP-announced /23, or to examine the Jaccard similarity for more specific prefixes, like the more specific /24 prefix of the BGP-announced /23 prefix. We implement and thoroughly analyze both the less specific (see §5.2.5 for more details) and more specific variants of SP-Tuner algorithm. In the remainder of the chapter we leverage SP-Tuner using more specific CIDR sizes as it produces better Jaccard similarity

Algorithm 1 SP-Tuner-MS (More Specific)

Data: DS domains, IPv4_addresses, IPv6_addresses, sibling prefix pairs, Jaccard similarity
Result: Refined sibling prefix pairs with improved Jaccard similarity
Initialization:
tree_v4 ← {*DS_domains*, *IPv4_addresses*}
tree_v6 ← {*DS_domains*, *IPv6_addresses*}
curr_jacc ← *precomputed non-zero value*
prefix_v4_len_thresh ▷ IPv4 prefix length threshold
prefix_v6_len_thresh ▷ IPv6 prefix length threshold

```
for (sibling_prefix_v4, sibling_prefix_v6) ∈ sibling_prefix_pairs do
    tree_v4 ← sibling_prefix_v4
    tree_v6 ← sibling_prefix_v6
    new_jacc ← 0
    while new_jacc ≤ curr_jacc do
        prefix_v4_subprefixes ← GetNextSubprefixes(sibling_prefix_v4)
        prefix_v6_subprefixes ← GetNextSubprefixes(sibling_prefix_v6)
        for prefix_v4 ∈ prefix_v4_subprefixes do
            for prefix_v6 ∈ prefix_v6_subprefixes do
                new_jacc ← max(Jaccard(prefix_v4, prefixv6))
                if HasBranch(prefix_v4, prefix_v6) then
                    sibling_prefix_pairs ← UpdateBranches()
                end if
            end for
        end for
        if (prefix_v4_len ≥ prefix_v4_len_thresh)
      and (prefix_v6_len ≥ prefix_v6_len_thresh) then
            return (prefix_v4, prefix_v6)
        end if
    end while
end for
```

values. A more specific CIDR block, like /28 instead of /24, gives network operators tighter control over which IP addresses are affected by a policy. This level of precision is beneficial when network operators want to block a small set of suspicious IPs without disrupting legitimate traffic or when allowing access only to a narrow group of trusted hosts. Broader prefixes can be too blunt and end up causing unintended side effects. We implement the SP-Tuner algorithm with two PyTricia tree [149] data structures for each IP version and their respective DS domains. PyTricia facilitates efficient storage and retrieval of IP addresses and their associated domains within a tree data structure.

As shown in Algorithm 1, the more-specific variant of SP-Tuner (i.e., SP-Tuner-MS) processes each sibling prefix pair to refine it into more specific subprefixes. The algorithm traverses each tree in the downward direction from the point of prefix insertion to identify more specific prefixes with improved Jaccard values. During this traversal, some domains may fall under alternate branches that are not part of the best-matching subprefix. These subprefixes still contain relevant domains but are excluded from the maximal pair. To prevent domain loss, we track these additional branches and treat them as new candidate sibling prefix pairs, applying the SP-Tuner algorithm to them as well. This approach ensures that no domains are lost during prefix size tuning. Additionally, we employ thresholds at both IPv4 and IPv6 prefix levels to ensure that tree traversal stops at meaningful levels.

5.2.4 Effectiveness of SP-Tuner

To evaluate the effectiveness of the SP-Tuner algorithm, we apply the algorithm on the most recent domain snapshot from Septeber 11, 2024. We configure two CIDR size values, one for IPv4 and one for IPv6, as the threshold of the SP-Tuner algorithm. The SP-Tuner

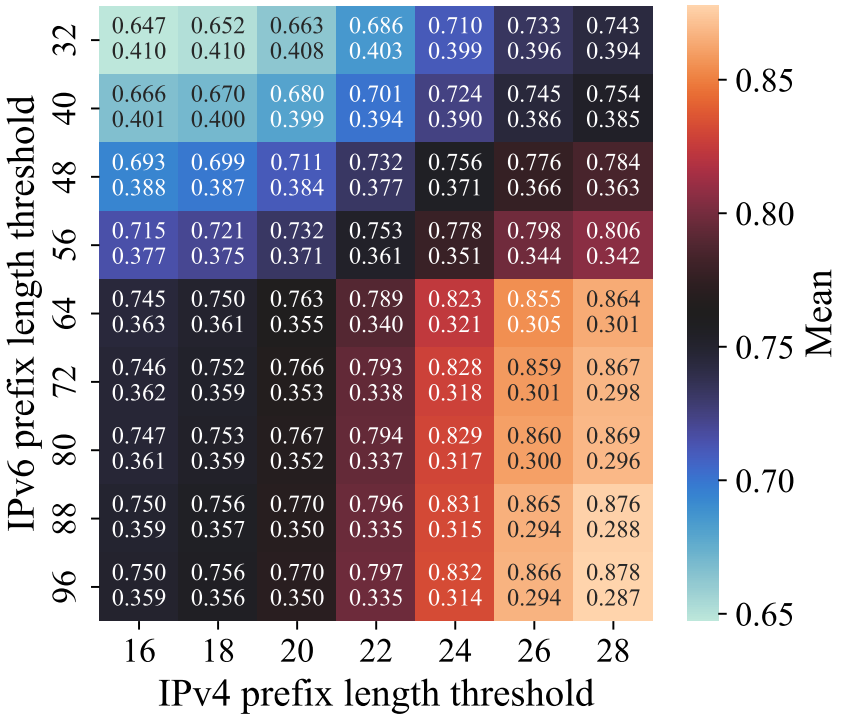

Figure 5.4: Heatmap of SP-Tuner Algorithm Performance: Mean Jaccard Index (top) and Standard Deviation (bottom) across IPv4 (x-axis) and IPv6 (y-axis) CIDR size thresholds.

algorithm calculates the Jaccard value of sibling prefixes by reducing the sibling prefix size up to and including the threshold to check and improve the Jaccard value whenever possible. To assess the impact of different prefix size thresholds, we investigate various IPv4 and IPv6 prefix sizes, starting from /16 up to /31 for IPv4, and from /32 up to /124 for IPv6 sibling prefixes, respectively.

Figure 5.4 shows our results for SP-Tuner thresholds up to /28 IPv4 and /96 IPv6 prefix sizes. We refer the interested reader to Figure 5.6 in §5.2.4.1 for the complete heatmap. The color in Figure 5.4 and the top value within each cell indicate the mean Jaccard value for all sibling prefixes, and the bottom value within each cell shows the standard deviation of the Jaccard values. Moreover, the x-axis and y-axis in Figure 5.4 represents IPv4 and IPv6 prefix length thresholds, respectively, starting from /16 for IPv4 and /32 for IPv6, with subsequent thresholds increasing incrementally.

Examining the mean Jaccard values of Figure 5.4 row-wise for the IPv4 threshold shows that the more specific the IPv4 CIDR size, the higher the mean Jaccard value. The same pattern applies to the column-wise mean Jaccard values for the IPv6 threshold. We find that from the lowest mean Jaccard value of 0.647 for the combination IPv4 threshold of /16 and IPv6 threshold of /32, the SP-Tuner algorithm can improve the mean Jaccard value up to 0.878 for the /28 and /96 threshold values of IPv4 and IPv6 prefixes, respectively. On the same combination, SP-Tuner also reduces the standard deviation from 0.410 to 0.287. The improvements in Jaccard similarity, deriving more similar sibling prefixes for existing prefixes with low standard deviation, show that the SP-Tuner algorithm is able to identify more fine-tuned sibling prefixes with higher

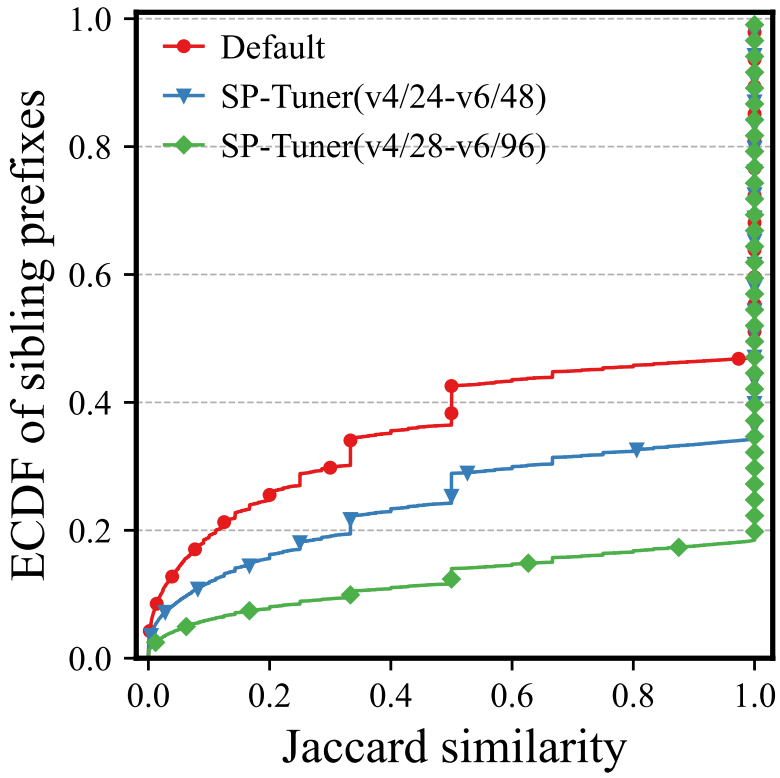

Figure 5.5: CDF of Jaccard similarity for sibling prefixes comparing default values with SP-Tuner algorithm at routable and optimal IPv4/IPv6 thresholds.

Jaccard similarity. Note that the goal is to identify sibling prefixes at the prefix level, not the IP address level. Hence, we pick prefix thresholds of /28 and /96. We leave the choice to users of sibling prefixes to pick suitable CIDR sizes based on their specific use case, be it the default BGP-announced sibling prefixes as seen in the dataset, or /24 IPv4 and /48 IPv6 thresholds for most-specific routable prefixes, or /28 and /96 as in our case. For this study, we provide a detailed analysis based on the /28 and /96 threshold prefixes as the highest Jaccard similarity value with the lowest standard deviation.

We further examine the impact of the SP-Tuner algorithm by applying it on the data from September 11, 2024 and comparing it with the default BGP-announced, as seen in the DNS data, sibling prefixes. Figure 5.5 shows the CDF of Jaccard similarity values for the default BGP-announced sibling prefixes in red, SP-Tuner routable threshold of /24 and /48 for IPv4 and IPv6, respectively, and the best performance SP-Tuner case with an IPv4 threshold of /28 and an IPv6 one of /96. We see for the default case around 52% of sibling prefixes have a Jaccard value of 1, i.e., they are perfect matches. After applying SP-Tuner, this percentage increases to 67% for the routable prefix threshold and to more than 82% for the /28 and /96 thresholds. In summary, the SP-Tuner with the best performance thresholds increases the number of sibling prefixes from 52% in the default case to around 82% of sibling prefix pairs, all having Jaccard similarity index of 1 and more tailored smaller CIDR sizes by excluding irrelevant address spaces of prefixes.

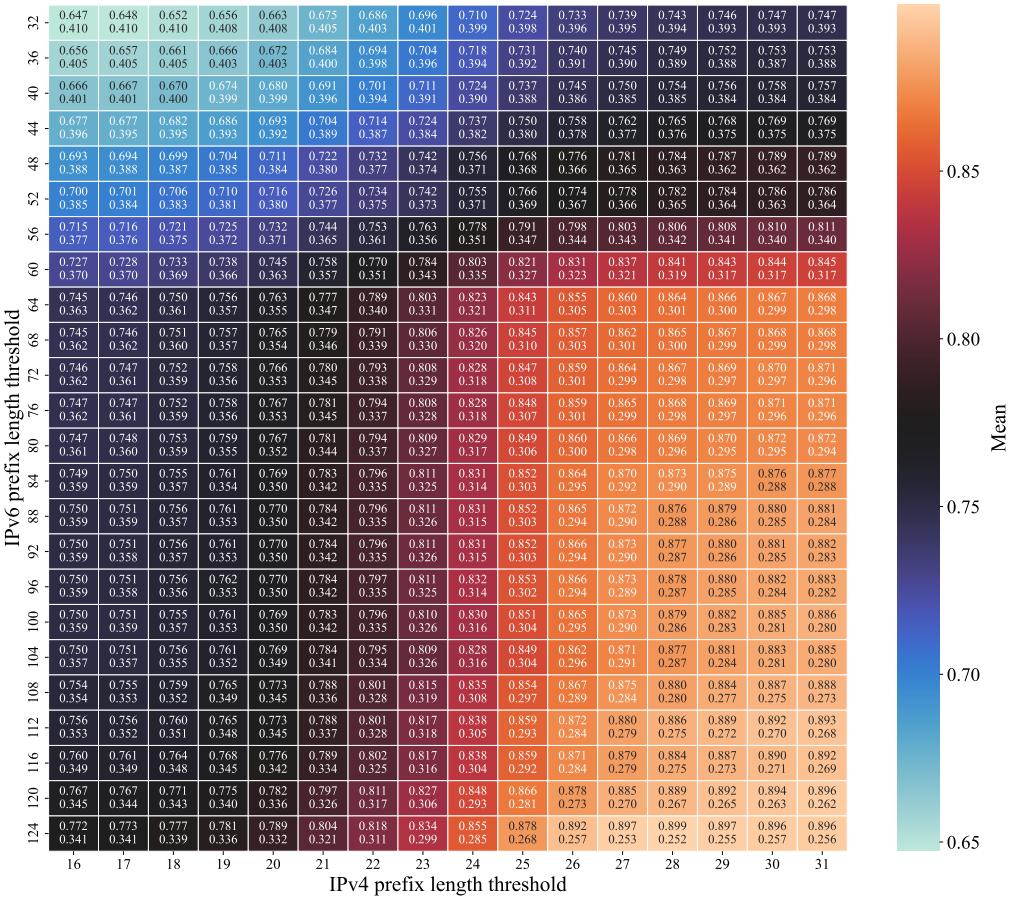

Figure 5.6: Heatmap of SP-Tuner Algorithm Performance: Mean Jaccard Index (top) and Standard Deviation (bottom) across IPv4 (x-axis) and IPv6 (y-axis) CIDR size thresholds.

5.2.4.1 SP-Tuner-MS Sensitivity Analysis

To achieve an improved Jaccard similarity, we conduct a sensitivity analysis to evaluate the behavior of SP-Tuner-MS under various prefix length thresholds. The results of these tests, performed on IPv4 prefixes ranging from /16 to /31 and IPv6 prefixes ranging from /32 to /124, is illustrated in Figure 5.6. In each cell, the top value represents the mean Jaccard similarity across all sibling prefixes, while the bottom value indicates the standard deviation of the Jaccard values.

5.2.5 SP-Tuner-LS (Less Specific)

Algorithm 2 with the name SP-Tuner-LS shows the traversal of each tree in the upward direction from the point of prefix inserts. At every step, it generates the supernet of the current prefix and checks Jaccard similarity until improvement occurs. Another point to check at every step of going up is the AS number; this might happen when generating the supernet of a prefix, causing the originating AS number to change. To address this, leverage the RouteView data for that specific date, ensuring the same date as our input data.

Algorithm 2 SP-Tuner-LS (Less Specific)

Data: DS domains, IPv4_addresses, IPv6_addresses, sibling prefix pairs, Jaccard similarity
Result: Refined sibling prefix pairs with improved Jaccard similarity
Initialization:
  $tree\_v4 \leftarrow \{DS\_domains, IPv4\_addresses\}$
  $tree\_v6 \leftarrow \{DS\_domains, IPv6\_addresses\}$
  $curr\_jacc \leftarrow$ *precomputed non-zero value*
  $prefix\_v4\_len\_thresh$ ▷ IPv4 prefix length threshold: 1 levels up
  $prefix\_v6\_len\_thresh$ ▷ IPv6 prefix length threshold: 4 levels up
for $(sibling\_prefix\_v4, sibling\_prefix\_v6) \in sibling\_prefix\_pairs$ **do**
  $tree\_v4 \leftarrow sibling\_prefix\_v4$
  $tree\_v6 \leftarrow sibling\_prefix\_v6$
  $new\_jacc \leftarrow 0$
  while $new\_Jacc \leq curr\_Jacc$ **do**
    $prefix\_v4\_superprefixes \leftarrow GetNextSuperprefixes(sibling\_prefix\_v4)$
    $prefix\_v6\_superprefixes \leftarrow GetNextSuperprefixes(sibling\_prefix\_v4)$
    for $(prefix\_v4, prefix\_v6) \in \{prefix\_v4\_superprefixes, prefix\_v6\_superprefixes\}$ **do**
      $new\_Jacc \leftarrow \max(Jaccard(prefix\_v4, prefix\_v6))$
      if $(IsASnumChange(prefix\_v4))$ || $(IsASnumChange(prefix\_v6))$ **then**
        return $(prefix\_v4, prefix\_v6)$
      else if $(prefix\_v4\_len \leq v4\_len\_thresh)$
    || $(prefix\_v6\_len \leq v6\_len\_thresh)$ **then**
        return $(prefix\_v4, prefix\_v6)$
      end if
    end for
  end while
end for

The Figure 5.7 shows the results of applying SP-Tuner-LS. Using this approach and increasing the prefix sizes for fine-tuning the sibling prefixes does not improve the Jaccard similarity values, as depicted by the green line in Figure 5.7. However, the blue line indicates that setting a threshold on the prefix sizes for how much they could be increased does not result in a significant improvement in Jaccard similarity.

5.2.6 Ground Truth Evaluation

To compare our identified sibling prefixes with a real-world deployment, we use RIPE Atlas probes [200] as a "ground-truth" dataset. We extract the publicly available IPv4 and IPv6 addresses from dual-stack RIPE Atlas probes, map them to prefixes, and compare these prefixes to our identified sibling prefixes.

From a total of 5174 dual-stack probes, 2200 (42.5%) probes have IPv4 and IPv6 covered by our sibling prefixes, 1663 (32.1%) of probes are partially covered by our sibling prefixes (i.e., either IPv4 or IPv6), and 1310 (25.3 %) probes are not covered by our sibling prefixes at all. This shows that almost half of the dual-stack probes are completely covered by our sibling prefix dataset, which means that we can evaluate the similarity of the RIPE Atlas dual-stack probes with our sibling prefix results. Out of the 2200 completely covered RIPE Atlas probes, 1966 (89.36%) are in our best-match sibling prefixes (i.e., results of step 4 of Figure 5.2). On the other hand, only 234 (10.64%) of the completely covered probes are not best-match siblings. This underlines that our sibling prefixes line up quite well with the RIPE Atlas probes ground-truth dataset, which increases the confidence in our chosen methodology.

Finally, we also check our identified sibling prefixes against 260 globally distributed dual-stacked virtual private servers (VPSes) from IPinfo's probe network [140] hosted by different VPS providers (e.g., Google, Azure, Vultr, AWS). For the VPSes where we get both an IPv4 and IPv6 address match in our siblings, 53 are within our best-match

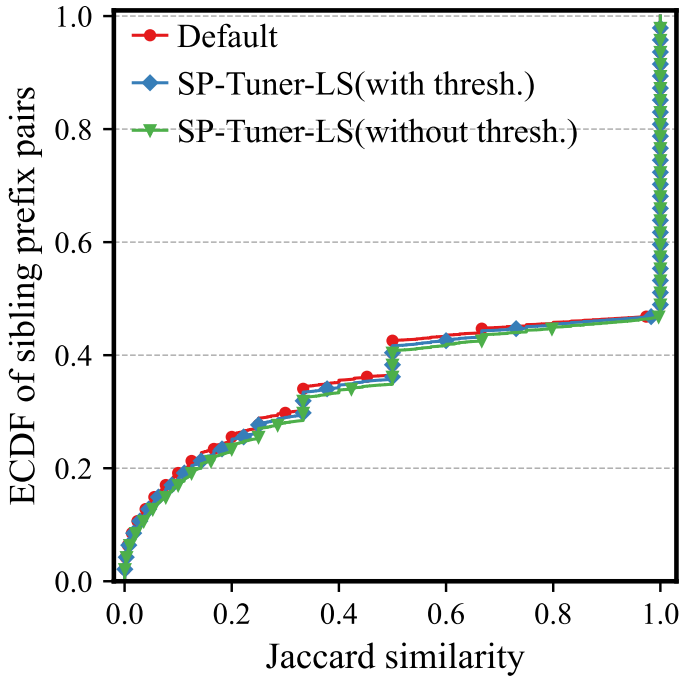

Figure 5.7: Jaccard similarity after applying SP-Tuner-LS on sibling prefix pairs data.

siblings, whereas 13 are mismatches, with the remainder not having IPv4 and IPv6 matches. This again shows, that the majority of our best matches provide accurate siblings matches.

5.2.7 Overlap with Open Ports

To better understand if our identified sibling prefixes based on domain names also align with actual deployed infrastructure, we compare our domain-based approach with an open-port-based approach. We use IP addresses from our latest OpenINTEL dataset on September 11, 2024 as the reference dataset and perform active scanning on September 19, 2024. We then perform port scans using ZMap [84] and ZMapv6 [280] on 14 common open ports [19, 83, 244], i.e., 20, 21, 22, 23, 25, 53, 80, 110, 123, 143, 161, 194, 443, 7547. For 70.9% of sibling prefixes we get responses. Then, we map the addresses from the port scanning results with the prefixes in sibling prefixes to identify the associated prefixes for them. We use the Jaccard similarity index to determine the similarity between these prefix pairs. However, instead of using the domain set, we utilize the responsive ports within IPv4 and IPv6 prefixes.

In Figure 5.8 the heatmaps display the Jaccard similarity values of sibling prefixes based on the port scan on the y-axis, and the Jaccard values for the same pairs of sibling prefixes based on the OpenINTEL dataset on the x-axis. The color and number in each heatmap cell shows the percentage of sibling prefixes based on 28-96 threshold pair of SP-Tuner algorithm. For the top row, the rightmost cell shows a value of 36% as the highest number of sibling prefixes matching ≥ 0.9 Jaccard value on the both x and y axes. This indicates a correlation between the similarity of the OpenINTEL results and port

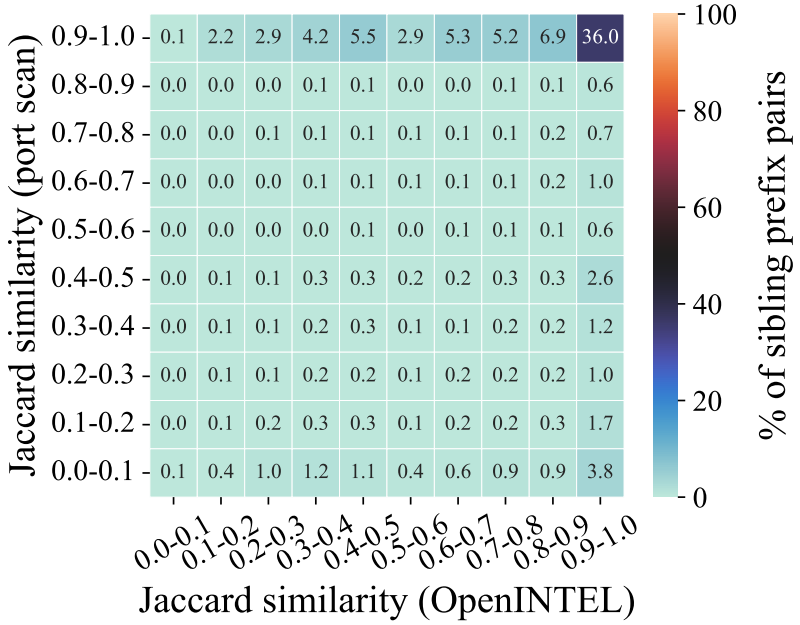

Figure 5.8: Heatmap of open port scanning: Each cell shows the percentage of sibling prefixes, with the x-axes and y-axes being the Jaccard values for DNS (OpenINTEL) and port scan data, respectively.

scan results for sibling prefixes. Moreover, the values in the topmost row of Figure 5.8 reflect that for sibling prefix pairs having high Jaccard value (≥ 0.9) in port scan, the OpenINTEL Jaccard similarity is distributed to a lower range. A similar pattern with a slightly lower percentage is valid by comparing the OpenINTEL Jaccard value with ≥ 0.9 on the rightmost column to the port scan on the y-axis. In summary, 36% of responsive sibling prefix pairs with a high Jaccard similarity of ≥ 0.9 on OpenINTEL dataset are also likely to have a high Jaccard value based on port scan results.

5.2.8 Limitations

Although we try to fine-tune our methodology quite well to the task of sibling prefix detection, there are some limitations to it, which we will elaborate in the following. As our sibling prefix methodology is fundamentally based on domain name data, it is only as good as the used domain name data for sibling identification. In our analysis, we leverage a large-scale DNS resolution dataset (see §5.1.1). Even though the dataset covers more than 13 million unique domains, our findings are limited to the used dataset, which represents only a subset of all DNS domains. Thus, adding even more domain name data as an input to the methodology, might provide better coverage in terms of the number of sibling prefixes and potentially also their quality. Our methodology can also be applied with inputs different than domain names, such as alias datasets or open ports on devices (see §5.2.7). As long as these inputs result in a mapping from a prefix to a set, our technique of using set similarity and then picking the maximum similarity value can still be applied. Despite these limitations, we believe that our methodology is suitable to detect sibling prefixes in the Internet.

5.2.9 Ethical Considerations

In this work, we follow ethical Internet measurement guidelines [80, 84, 209]. First, we use use publicly available datasets (see §4.1). Second, for ground truth evaluation and comparison of DNS data with open ports (see §5.2.6 and §5.2.7), we perform port scans on IP addresses from the OpenINTEL dataset. To limit potential harm for third parties, we use a blocklist, maintain public web pages describing our measurement activity and providing contact information, and we limit our scanning rate to 50 kpps. Third, during our measurements, we did not receive any complaints. Thus, we believe that our work does not raise ethical concerns.

5.3 Sibling Prefixes

In this section, we thoroughly analyze sibling prefixes observed in our dataset. We begin with examining IP and prefix changes of dual-stack domains, followed by exploring their distribution in sibling prefixes. We then continue exploring sibling prefixes by conducting a longitudinal analysis and by assessing the impact of SP-Tuner on historical data. Next we analyze CIDR sizes, origin ASes, and business types of sibling prefixes. Finally, we conclude the analysis by investigating differences of sibling prefix characteristics in hypergiant and CDN networks and examine their ROV status.

Throughout this section, we use the sibling prefix pairs resulting from our SP-Tuner algorithm with a /28 IPv4 and a/96 IPv6 prefix threshold. For some cases where using sibling prefixes as observed in BGP announcements (i.e., the default case) is more relevant to our analysis, we use that notion of sibling prefixes. Analyzing the OpenINTEL dataset, we identify 3.5M dual-stack domains running on IP addresses of 46.3k IPv4 and 39.5k IPv6 prefixes. These prefixes originated from 6.6k IPv4 and 5.9k IPv6 ASes, collectively they result in 76k sibling prefixes as of September 2024.

5.3.1 Address and Prefix Dynamics in Dual-Stack Domains

Before diving into the analysis of sibling prefixes themselves, we first analyze address and prefix dynamics found in DS domains. The goal of this analysis is to understand to what extent addresses and prefixes of DS domains change over time. We analyze address and prefix changes for the DS that are consistently visible in our dataset. To identify these consistent DS domains, we examine how frequently a sibling prefix appears in our dataset over a one year period from September 13, 2023, to September 11, 2024, i.e., a total of thirteen monthly snapshots. We collect data over a 13-month period to ensure year-over-year coverage and make full use of available data.

The left subplot in Figure 5.9 illustrates the cumulative distribution of DS domains on the y-axis and the number of times a DS domain appeared in the data ("visibility frequency") on the x-axis. Around 40% of DS domains are consistently visible in all thirteen data points. About 20% of DS domains appear only once, and the remaining 40% of DS domains have a visibility frequency of two to twelve snapshots. These percentage values indicate that apart from the 40% consistent DS domains, the remaining 60% of DS domains are either not consistently dual-stack (i.e., they switch between dual-stack to IPv4/IPv6-only) or are not always present in our dataset.

Next, we want to analyze and peek into the temporal prefix stability of the 40% consistent DS domains. The center subplot of Figure 5.9 shows the percentage of DS domains having

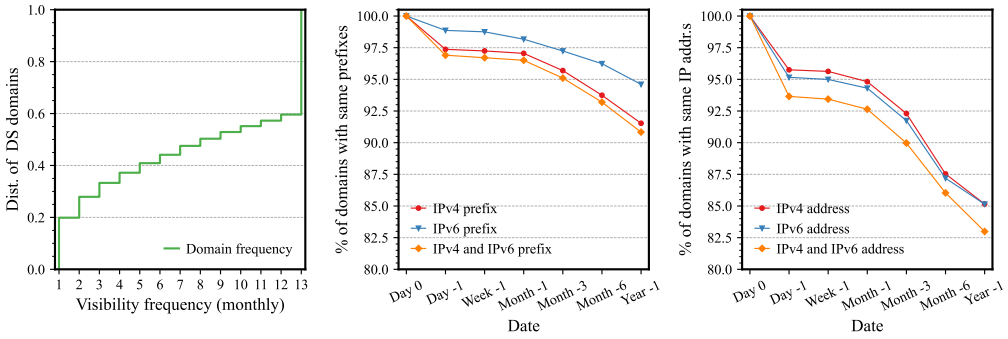

Figure 5.9: Consistently visible DS domains (left), prefix changes (center), and address changes (right).

stable prefixes over time, again with the reference date of September 11, 2024, as "day 0" on the x-axis. We observe that around 95% of consistently visible DS domains are resolved to the same prefix for up to three months. We do not see any prefix change throughout the one-year analysis period for more than 91% of DS domains. While the maximum change for IPv4 prefixes is around 9%, this number is lower at around 6% for IPv6 prefixes. This shows that overall, consistent DS domains exhibit high prefix stability.

Finally, we analyze changes of consistently visible DS domains on the IP address level, shown in the right subplot of Figure 5.9. We find that 83% of DS domains are mapped to the same IP addresses throughout the analysis period. Overall, on the IP address level IPv4 and IPv6 behave much more similarly compared to the prefix level. This indicates that addresses change at a similar rate in both protocols, but prefixes are more stable in IPv6 compared to IPv4.

In summary, We show that 40% of DS domains are consistently visible for thirteen months. Additionally, 83% of these consistent DS domains have stable addresses, and 91% of the consistent DS domains have stable prefixes throughout one year. This shows that consistent DS domains from our DNS dataset are stable and can therefore be used for our sibling prefix identification technique.

5.3.2 Dual-Stack Domains in Sibling Prefixes

Next, we focus on analyzing the number of DS domains within each sibling prefix pair. For this purpose, we first generate sibling prefixes using the SP-Tuner algorithm with a /28 and /96 CIDR threshold. Then, we group sibling prefixes by the number of dual-stack domains within each pair. As shown in Figure 5.10, we bin sibling prefixes regarding the number of associated domains. The cell color and number in the heatmap show the percentage of sibling prefixes for each group of DS domains. Over 55% of sibling prefix pairs contain a single domain, as shown in the bottom left corner of the plot. The sibling prefix pairs with 2 to 5 domains are the second-largest group, making up 21.3% of sibling prefixes. Furthermore, sibling prefixes with more than 100 domains for IPv4 and IPv6 contribute 1.6% to all sibling prefixes. Finally, the diagonal cells in the heatmap contains a relatively high percentage of sibling prefixes compared to neighboring cells, indicating that sibling prefixes are relatively likely to share a similar number of DS domains for IPv4 and IPv6 prefixes.

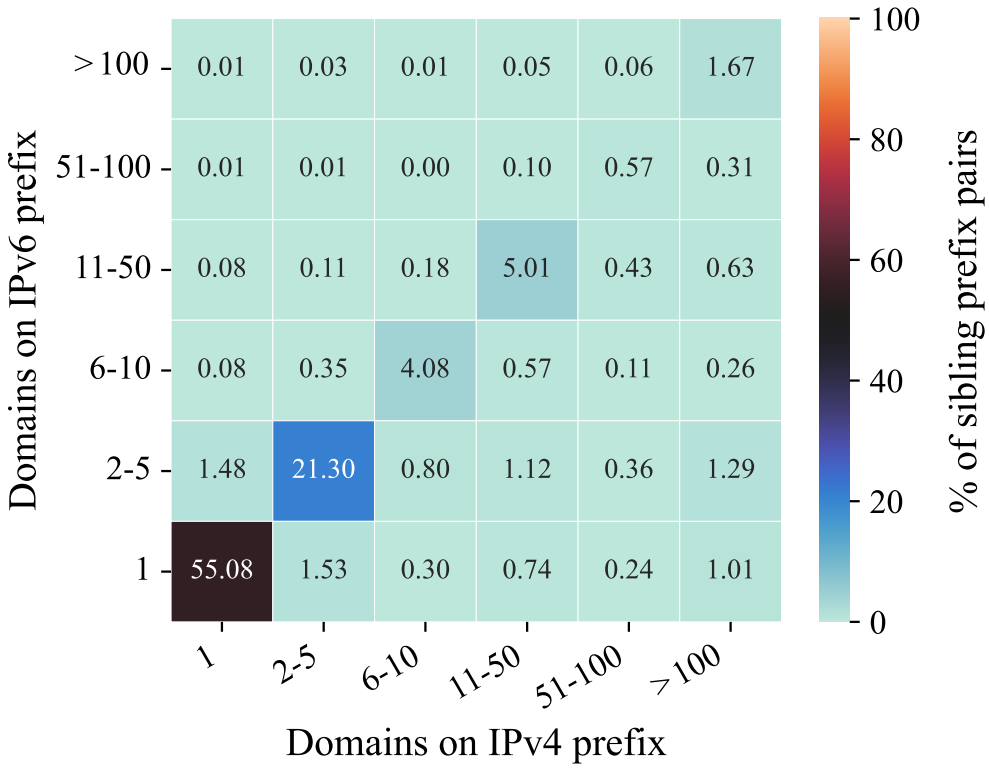

Figure 5.10: Sibling prefix pairs count and classification based on number of dual-stack domains in each IPv4 and IPv6 prefix of sibling prefixes

In summary, The majority of sibling prefixes contain single domains. Moreover, we observe a relatively high number of IPv4 sibling prefixes with more than 100 dual-stack domains. Finally, sibling prefixes tend to have a similar number of domains for IPv4 and IPv6.

5.3.3 Longitudinal Analysis

In this part of the analysis we perform a longitudinal analysis of sibling prefixes over a 4-year period, as shown in Figure 5.11. The x-axis shows different points in time starting from the base date September 11, 2024 denoted as "day 0", and previous points in time denoted by the difference to this base date. Over the past four years, the number of sibling prefixes has doubled from around 36k to more than 76k. This is due to the addition of dual-stack domains in OpenINTEL (see §5.1.1) and to the increased use of IPv6 among previous IPv4-only domains.

To provide insights into temporal effect on sibling prefixes, we perform a longitudinal analysis of Jaccard values. To this end, we compare our most recent snapshot (September 2024) with data from four years ago (September 2020). Note that—as any longitudinal analysis—this analysis is generally affected by changes in the underlying used datasets as well as changes in the Internet as a whole. When investigating the potential impact of changes in the datasets, we can see that the percentage of DS domains remains very stable, even for adding the ".fr" TLD in August 2022 and removing the Alexa toplist from

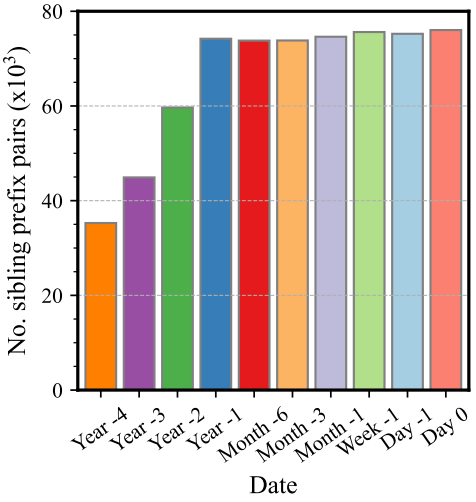

Figure 5.11: Number of sibling prefixes at different points in time. (Reference date: September 11, 2024)

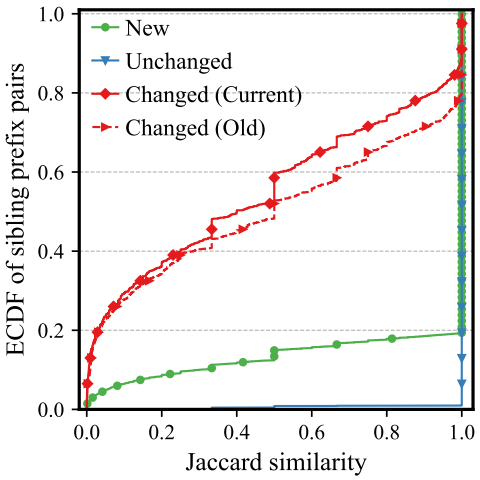

Figure 5.12: Improvement of Jaccard similarity of sibling prefixes over time.

the OpenINTEL data in May 2023 (see Figure 5.1 right). When looking at the number of sibling prefixes in Figure 5.16 we see a slight changes for these two dates. Rather than sudden jumps or drops these changes are more gradual, hinting at multiple different effects being at play in addition to changes in the underlying datasets. Therefore, we believe that the general trends in the longitudinal analaysis still provide valuable insights into sibling prefix changes over time.

Figure 5.12 compares the Jaccard value distribution for sibling prefixes where the Jaccard value has not changed ("unchanged", blue line), with changed sibling prefixes ("changed", red lines), and completely new sibling prefixes ("new", green line). We further differentiate the changed sibling prefixes by plotting their old Jaccard value (dashed line) and new Jaccard value (solid line).

From a total of 76k sibling prefixes, the "new" category makes up 67k (88%), "unchanged" 7.7k (10%), and "changed" 1.2k (2%) of sibling prefixes. We find that unchanged sibling prefixes exhibit the highest similarity, with almost all of them having a Jaccard value of 1. New sibling prefixes are more similar (80% with Jaccard value 1) compared to old, but changed (21% with Jaccard value of 1) sibling prefixes. Finally, the current Jaccard value of changed sibling prefixes is lower compared to their Jaccard value four years ago.

To measure the impact of SP-Tuner on Jaccard similarity, we compare sibling prefixes as observed in our dataset (i.e., with default BGP-announced prefix sizes) in Figure 5.13 with sibling prefixes after applying SP-Tuner in Figure 5.14. For the default case, we see that around 45 to 55% of sibling prefixes have a Jaccard similarity value of 1. However, after applying SP-Tuner in Figure 5.14, the percentage of sibling prefixes with a Jaccard similarity value of 1 is almost doubled to around 80%. This improvement in Jaccard similarity values shows that SP-Tuner is able to identify a high number of fine-tuned sibling prefixes for various points in time.

In summary, The number of sibling prefixes has more than doubled compared to four years ago to around 76k in September 2024. We find that the Jaccard value of sibling prefixes has substantially improved over time, with old prefixes having around 21% and

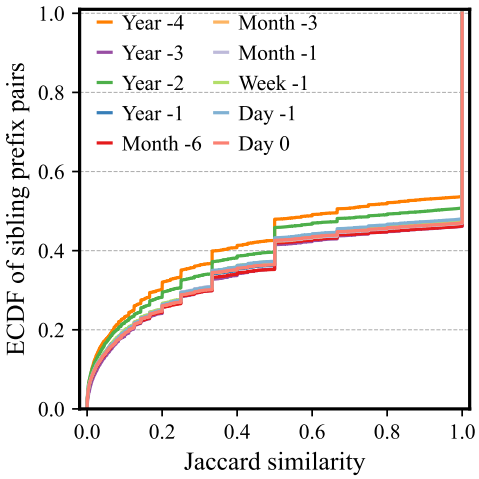

Figure 5.13: Jaccard similarity of sibling prefixes for various data points in time. (Reference date: September 11, 2024)

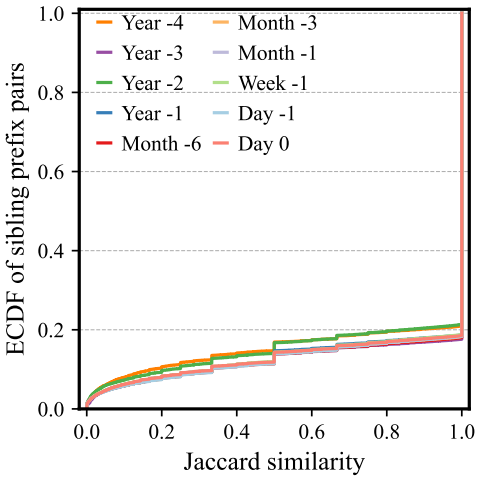

Figure 5.14: CDF of Jaccard similarity values for sibling prefixes on various data snapshots in time after applying SP-tuner algorithm.

new sibling prefixes having 80% of prefix pairs with a Jaccard value of 1. The changed sibling prefixes have the least, 18%, and unchanged sibling prefixes have the highest percentage of 99% sibling prefixes with a Jaccard value of 1. Moreover, fine-tuned sibling prefixes identified by SP-Tuner outperform BGP-announced sibling prefixes by almost doubling the fraction of perfect matches.

5.3.4 CIDR Sizes

We continue our analysis by examining the most common CIDR sizes of IPv4 and IPv6 in sibling prefix pairs. As for SP-Tuner, the vast majority, i.e., 86.95%, of sibling prefixes fall into /28 and /96 IPv4 and IPv6 CIDR sizes. This is due to the /28–/96 threshold applied by SP-Tuner, which maps most prefixes to these prefix length values (see §5.2.4.1). Therefore, we focus on the default case as it shows a more nuanced distribution as can be seen in Figure 5.15. To aid readability, we group sibling prefixes into different CIDR size groups, separating out commonly used CIDR sizes. We observe that /24 is the most prevalent CIDR size in IPv4, while /48 is most prominent in IPv6. Consequently, the /24-/48-combination makes up the largest share of sibling prefixes with 23.41% of pairs. Notably, we see relatively high percentages for cells in the region between /17–/24 in IPv4 and /32–/48 in IPv6, which together makes up more than 88% of all sibling prefixes. We observe a small fraction of sibling prefix pairs formed by less specific prefix CIDR sizes smaller than /17 for IPv4 and less specific than /32 in IPv6. Moreover, non-globally-routable prefixes [254] more specific than /24 in IPv4 and /48 in IPv6 are very rare among sibling prefixes.

In summary, the majority of SP-Tuner sibling prefixes fall into the /28-/96 group, whereas BGP-announced sibling prefixes are most commonly /24 in IPv4 and /48 in IPv6. Moreover, CIDR size ranges from /17 to /24 in IPv4 and /32 to /48 in IPv6, make up the vast majority of sibling prefixes on the Internet.

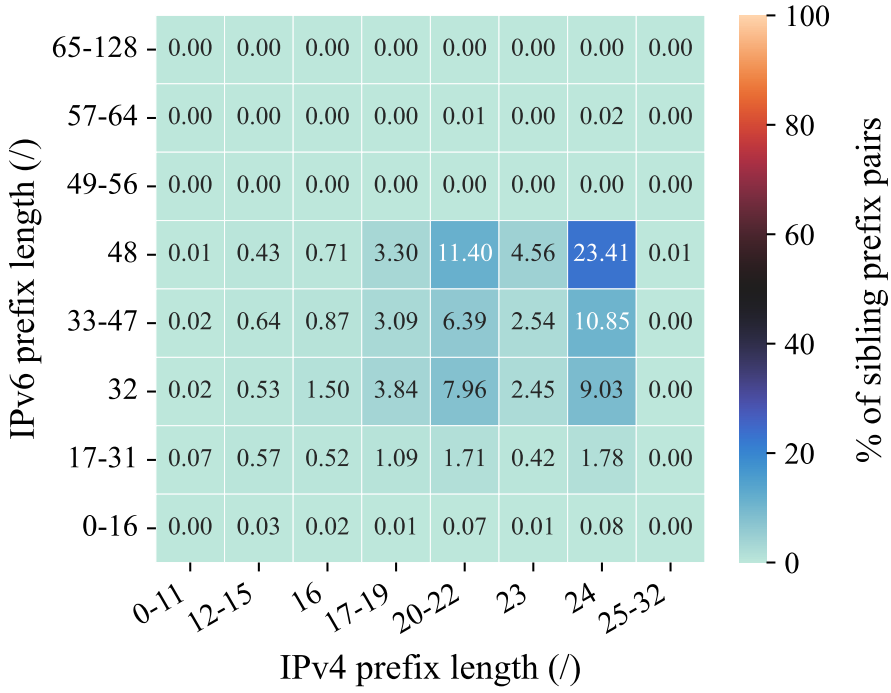

Figure 5.15: Distribution of CIDR sizes in sibling prefixes.

5.3.5 Origin ASes

In this section, we analyze the difference of sibling prefixes being originated by the same or different organizations. We merge sibling ASes in order to obtain sibling prefixes originated by the same or different organizations [294]. If the origin ASes of IPv4 and IPv6 sibling prefixes in our dataset either share the same AS number or, if different, are registered to the same organization name, we classify them as "same organization"; the remaining sibling prefixes are classified as "different organization". Figure 5.16 shows a longitudinal view of the number of sibling prefixes originated by the same or different organization category. We find that of the nearly 76k sibling prefixes in September 2024, more than half of them (around 41k) belong to the same organization category, indicating that their IPv4 and IPv6 origin ASes share the same organization name. Note that this does not mean that all of these prefixes belong to a single organization, but rather to multiple organizations whose IPv4 and IPv6 ASes are registered under the same name. Sibling prefix pairs from the different organizations category contribute around 35k pairs in September 2024. We also notice a dip in sibling prefixes originated by different organization ASes in May 2023 and earlier snapshots. Investigating the case, we find a unique domain (*site24x7.enduserexp.com*) missing in the OpenINTEL data on May 2023 and in earlier snapshots. Site24x7 is an IT infrastructure monitoring company [260], hosting probes in around 376 IPv4 and 55 IPv6 prefixes, all being originated by different origin ASes and with that single domain, resulting in around 20.5k sibling prefixes mostly with a Jaccard value of 1. In our dataset, *site24x7.enduserexp.com* is missing on May 2023 and on multiple dates in 2022 and 2021 (see dips of "diff. org." bars), resulting in fewer sibling prefixes for the different organization case.

Moreover, we show the number of *unique* IPv4 and IPv6 prefixes (red and blue lines),

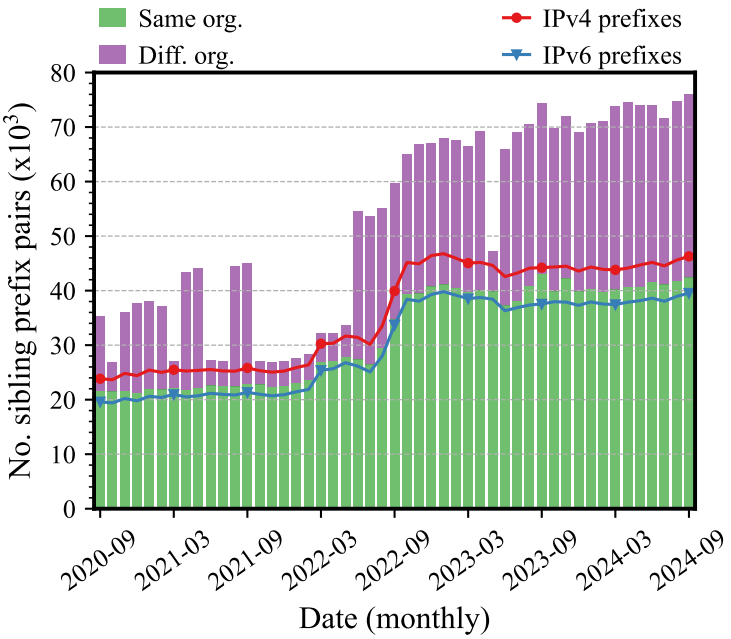

Figure 5.16: Sibling prefix pairs count for the same and different organizations over time. Lines represent the number of unique IPv4 and IPv6 prefixes in sibling prefixes.

which are generally quite stable over time. The slight increase in prefixes over time is due to improvements in OpenINTEL's domain coverage. Overall, we identify around 6.8k more IPv4 prefixes (46.3k) compared to IPv6 (39.5k) as observed in September 2024. The high ratio of IPv4 to IPv6 is due to the larger prefix size in IPv6 compared to IPv4, which in turn is able to cover more IP addresses and domains.

To shed more light on sibling prefixes depending on the origin AS organization, Figure 5.17 shows the median Jaccard value for sibling prefixes for same and different organizations over time. The green line shows the median Jaccard value for sibling prefixes of the same organization and has remained relatively stable at 1.0 over the past four years. The purple line, the median Jaccard value for sibling prefixes originated by different organizations, also has a Jaccard value of almost one with few dips to around 0.7 and 0.9 in 2021. The purple line shows an evident influence of the *site24x7.enduserexp.com* domain on median Jaccard values for different organizations. As such, when that domain is present, it causes many sibling prefixes pairs having a Jaccard value of 1, and its absence is causing the median Jaccard value to drop. This indicates that, if we exclude special cases such as Site24x7, sibling prefixes originated by the different organization generally exhibit a high similarity as sibling prefixes originated by the same organization case. Moreverove, other sites like Catchpoint [52], services that use multi-CDN [151, 238], alongside, some authoritative DNS that return more than one address to the queries might be potential reasons for different organizations category.

In summary, we identify around 46k unique IPv4 and almost 39k unique IPv6 prefixes, resulting in more than 76k sibling prefix pairs in the dataset as of September 2024. More than half of these sibling prefixes, both the IPv4 and IPv6 prefixes are originated by the same organization, which has a median Jaccard value of 1. The median Jaccard value for different organizations is sensitive to special case(s) involving a single domain.

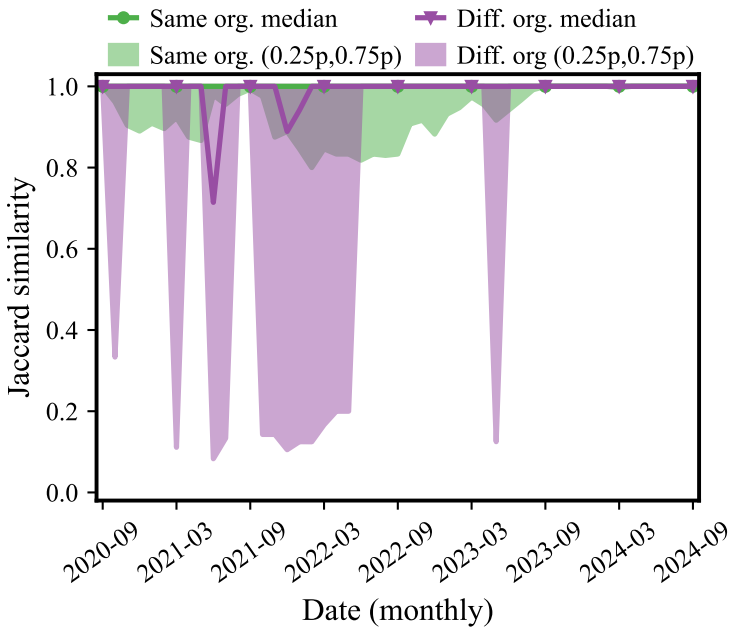

Figure 5.17: Median Jaccard similarity values of sibling prefix pairs for same and different organizations over time.

5.4 Business Type of Sibling Prefixes

To better understand the users of sibling prefixes, we analyze the business type of sibling prefix organizations. For this purpose, we use the most recent ASdb dataset [298] from January 2024 to identify the business type of each sibling prefix origin AS in our January 2024 sibling prefix snapshot as observed in our dataset. For this particular analysis, we use only those origin ASes of sibling prefixes that match a single business type, making up around 80% of all the prefixes. As we want to focus our attention on business relationships of sibling prefixes originated by different ASes, we therefore exclude sibling prefix pairs where both IPv4 and IPv6 prefixes have the same origin AS. The same AS number map to the same business type(s).

In Figure 5.18 we show the business type for origin ASes of IPv4 prefixes on the x-axis and for IPv6 prefixes on the y-axis, with the cell color indicating the number of sibling prefixes within this business combination. We notice that IT organizations (light yellow cell) form the largest fraction of sibling prefix pairs with more than 10k pairs for IPv4 and IPv6 origin ASes. Origin ASes in the education category (black cell in the top left) form the second largest fraction of around 150 sibling prefix pairs. The vertical and horizontal color density of the cells for IT organizations indicates the usage patterns of sibling prefixes for different ASes, where at least one of the origin ASes belongs to an IT organization. Showing that organizations mostly use IT companies, in addition to their own infrastructure for their services (i.e., websites). We also observe a small number of sibling prefix pairs with both origin ASes in the government, media, and retail categories.

To further deepen our knowledge of the business types of sibling prefixes, we consider the following two cases:

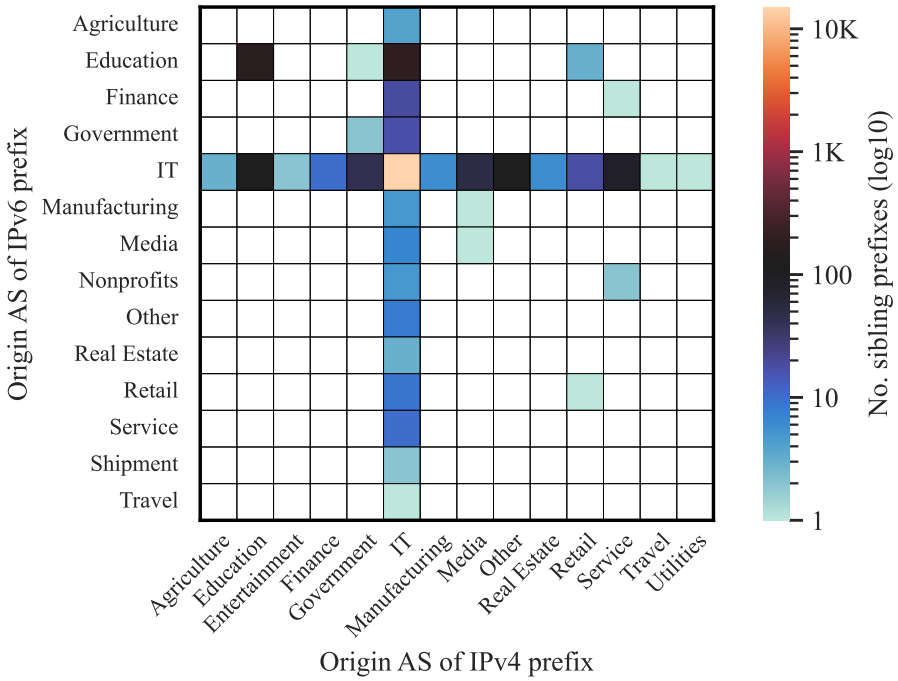

Figure 5.18: Business types heatmap for origin ASes of sibling prefixes, with IPv4 on x-axis and IPv6 on y-axis, indicating the count of sibling pairs in each cell.

a. Analysis of of Business Type for Origin AS of Sibling Prefixes

In this case, instead of counting the number of sibling prefixes, we count the number of unique origin AS pairs of the sibling prefixes. This analysis helps overcome the possible influence of a few origin ASes that might contribute to many sibling prefixes. While we still have both the ASdb and sibling prefixes data from January 2024, we exclude cases where the IPv4 and IPv6 prefixes of a sibling prefix pair have the exact origin AS number. Figure 5.19 illustrates the number of origin AS pairs for various business types of the origin ASes. The x-axis represents the business type of the IPv4 prefix origin AS, and the y-axis represents the business type of the IPv6 prefix origin AS.

The overall heatmap pattern for the count of origin AS pairs of sibling prefixes is very similar to the Figure 5.18 in §5.4 explained for the number of sibling prefixes. As it can be seen in Figure 5.19, the largest pair of ASes, more than one thousand pairs shown in light yellow color, are the cases where both origin ASes are It organizations. The origin AS pairs having an Education business type are the second largest group or origin AS pair for sibling prefixes. The row and column of cells for the IT business are colored, while the majority of other cells are empty, indicating that at least one of the origin ASes for the sibling prefixes is an IT organization. We can conclude that even if we consider a pair of unique origin ASes for the sibling prefixes in the business type analysis, the overall takeaway of the business type explained in §5.4 still holds.

b. Unfiltered Sibling Prefixes for Business Type Analysis Finally, we do not filter the cases where both prefixes in a sibling prefix pair have the same origin AS number. The idea is to provide a more inclusive picture of the sibling prefixes even if both origin AS have the same AS number. We want to understand the business types of those origin

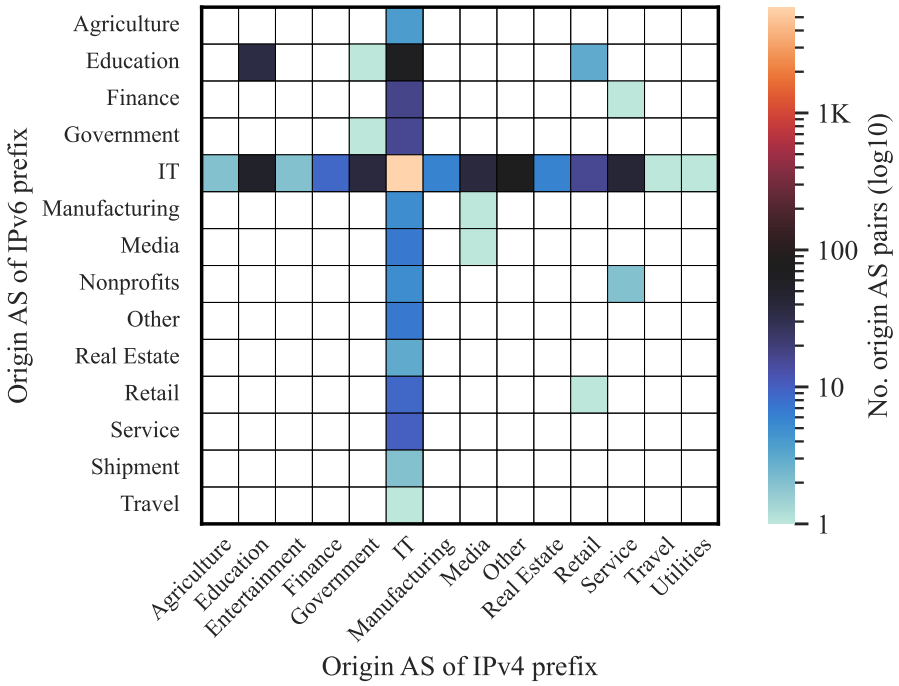

Figure 5.19: Business types heatmap for origin ASes of sibling prefixes, with IPv4 on x-axis and IPv6 on y-axis, indicating the count of origin AS pairs in each cell.

ASes and how much this affects the analysis we explained §5.4.

Figure 5.20 shows the business type analysis for all the sibling prefixes as observed in January 2024. We notice that the highest number in the color bar on the right side of the plot is increased from the range of one thousand, shown as 1K, to more than ten thousand, shown as 10K. The high number of sibling prefix pairs is because many sibling prefixes with the exact origin AS number are included in the data and as explained in §5.3.5. Another significant difference in Figure 5.20 comparing to Figure 5.19 and Figure 5.18 in §5.4 is the diagonal line of colorful cells. The sibling prefix pairs having the exact origin AS fall into the diagonal line, as both origin AS numbers fall into the same type of business. As a result, we see the diagonal line of the colorful cell in the heatmap in Figure 5.20. However, the cell having the highest number of sibling prefixes, more than ten thousand sibling prefix pairs, shown in light yellow color, still belongs to the IT organizations, which are high in number but similar to the previous analysis of the business type of the origin ASes for sibling prefixes. The IT organization has an overall higher number of sibling prefixes on the corresponding row and column, indicating that one of two origin ASes of sibling prefixes falls into the IT business type. The result is similar to all previous analyses concerning the business type of the origin ASes for sibling prefixes.

In summary, we observe that 80% of sibling prefixes with different origin ASes can be mapped to a single business type. IT companies contribute the highest share of around 10k sibling prefixes, with educational organizations following second. For most of the remaining sibling prefixes, at least one origin AS belongs to an IT organization.

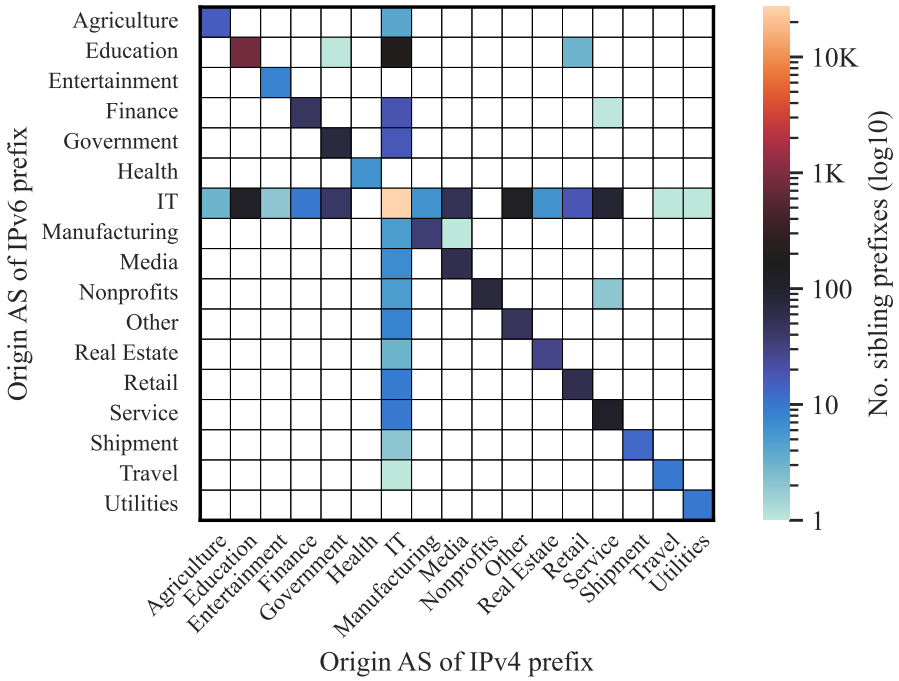

Figure 5.20: Business types heatmap for origin ASes of sibling prefixes, with IPv4 on x-axis and IPv6 on y-axis, indicating the count of sibling prefix pairs in each cell.

5.4.1 Hypergiants and CDNs

Next, we analyze to what extent sibling prefixes are used by hypergiants and CDNs. We analyze similarity values for sibling prefixes for known HGs and CDNs and compare them with other networks using our most recent data snapshot of September 11, 2024. For every prefix pair, we identify their organization using the dataset from Chen et al. [56] and use the list of publicly known HGs [30] and CDNs [53]. Note, we only consider SP-Tuner output sibling prefixes where IPv4 and IPv6 prefixes belong to the same organization of HG or CDN.

Figure 5.21 shows the distribution of Jaccard similarity values in HGs, CDNs, and other networks. The y-axis shows the organization names of one of the 24 HGs or CDNs, with the number in parentheses indicating the count of sibling prefixes for this specific HG or CDN. We group all the HGs and CDNs that contribute less than 50 sibling prefix pairs as "other-HG-CDN". Moreover, sibling prefixes that are neither originated by HGs nor CDNs are shown as "non-CDN-HG" in the bottom row.

The x-axis shows the Jaccard similarity values in ten cells, ranging from the smallest Jaccard similarity value of 0.0–0.1 to a very high similarity of 0.9–1.0. The color and number in each heatmap cell indicate the percentage of sibling prefix pairs for each HG or CDN with specific Jaccard similarity value. Thus, each row shows the distribution of sibling prefixes for a specific HG or CDN, summing up to 100%.

Examining the number of sibling prefixes, we identify the the highest number of sibling prefixes for Amazon, with 4564 pairs, followed by Microsoft, Akamai, and Google with 1125, 1056, and 1046 sibling prefix pairs, respectively. The right-most (0.9–1.0) column,

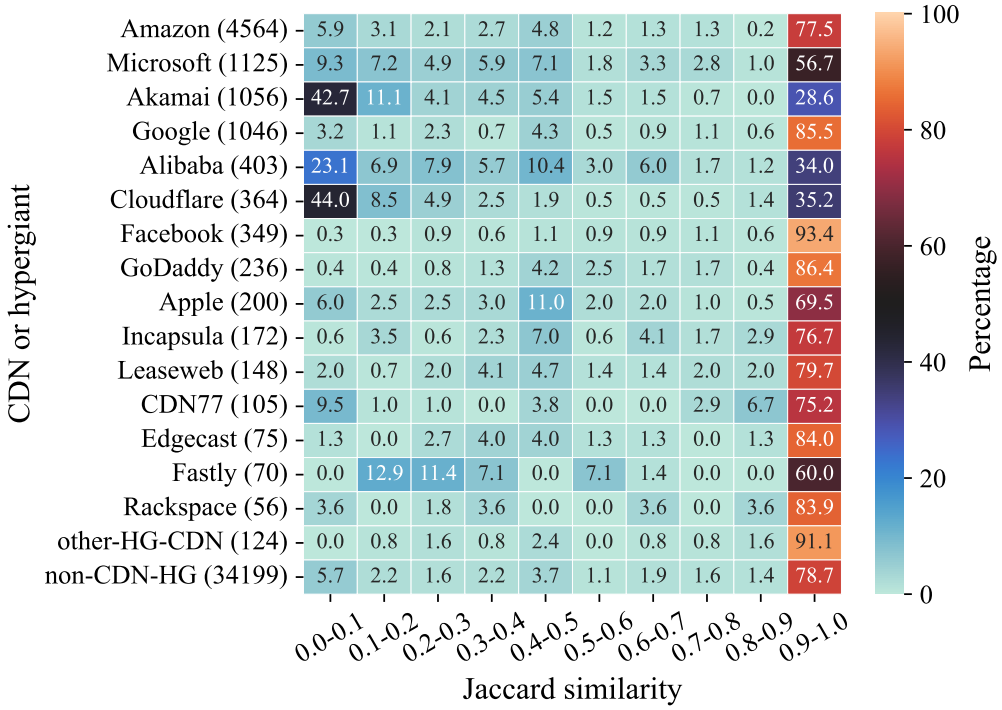

Figure 5.21: Heatmap of Jaccard similarity values (%) for sibling prefixes of Hypergiant and CDN networks, and non-CDN-Hypergiant ASes.

with a few exceptions, contains the highest shares of sibling prefixes. This shows that many sibling prefixes in an HG and CDN are very similar in containing perfect matching domains. Cloudflare and Akamai have the highest share of sibling prefixes in the worst matching category, i.e., with Jaccard values of 0.0–0.1. This aligns with certain CDNs moving semantics from IP addresses to domain names [87, 167, 182]. For non-CDN-HG ASes, 78% of sibling prefixes have a high Jaccard similarity value of 0.9–1.0. We observe the highest matching category for 93% of Facebook sibling prefixes and 91% of other-HG-CDN category followed by GoDaddy having 86% as the top three. Overall, we see that CDNs and HGs exhibit high similarity for sibling prefixes.

Figure 5.22 shows a detaild information of distribution of Jaccard similarity values for sibling prefixes in HGs, CDNs, using the /28-/96 thresholds of SP-Tuner algorithm.

In summary, we identify 24 hypergiant or CDNs using sibling prefixes. Amazon, Microsoft, and Akamai are the top three HGs with the highest number of sibling prefixes. The vast majority of sibling prefixes in HGs and CDNs are either very similar, with few HGs being exceptions with low similarity scores.

5.4.2 RPKI Validity

The RPKI can be used to check the legitimacy of a prefix announcement in BGP. Thus, we use RPKI data [234] to analyze the legitimacy of sibling prefix pairs as observed in our dataset. For this analysis we use the BGP-announced prefix sizes instead of the SP-Tuner output, as those align better for this BGP-specific analysis.

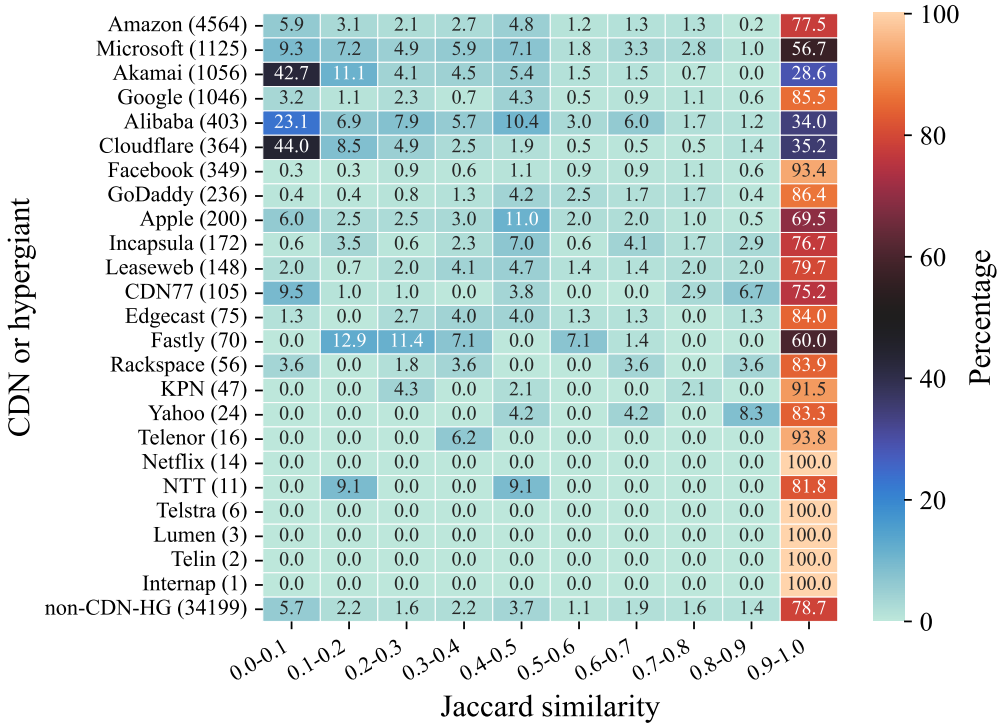

Figure 5.22: Heatmap of Jaccard similarity for the CDN, HGs and non-CDN-HG for the SP-Tuner thresholds of /28-/96.

Figure 5.23 illustrates the Route Origin Validation status for sibling prefixes from monthly snapshots between September 2020 and September 2024. While the x and y axes show the date and percentage, the share for each ROV status is displayed as the area over time. The red color indicates that both prefixes in a sibling prefix pair have a valid ROV status. The blue indicates that at least one of two prefixes has a valid ROV status, and the other was not found in RPKI. We find that for nearly 50% of sibling prefixes in 2020, to around 65% of sibling prefixes in September 2024, either one (i.e., the IPv4 or IPv6 prefix) or both prefixes of a sibling pair have a valid ROV status. For a small fraction of sibling prefixes (2–8%), we observe a conflicting ROV status, i.e., one prefix has a valid and the other an invalid ROV status (green). The share of both prefixes not being in the RPKI (purple) decreases from 40% in September 2020 to almost 20% in September 2024. For around 10% of sibling prefixes, we observe an ROV status of either one prefix being invalid and another not found or both prefixes having an invalid ROV status, as shown in orange and light green colors, respectively.

These findings are in line with the general trend of increased RPKI deployment in the Internet [195] and can help protect sibling prefixes from routing incidents such as BGP hijacks [24, 38, 217]. For all sibling prefix pairs, where one prefix is valid, and the other is not found, it is crucial to add the second prefix to the RPKI by creating a valid Route Origin Authorization object. It is even more important for cases where entire sibling prefix pair have invalid, a combination of (valid, invalid) or (invalid, not found) ROV state. In such cases, traffic might not be routed to those prefixes over both IP protocol versions, leading to a lack of resilience.

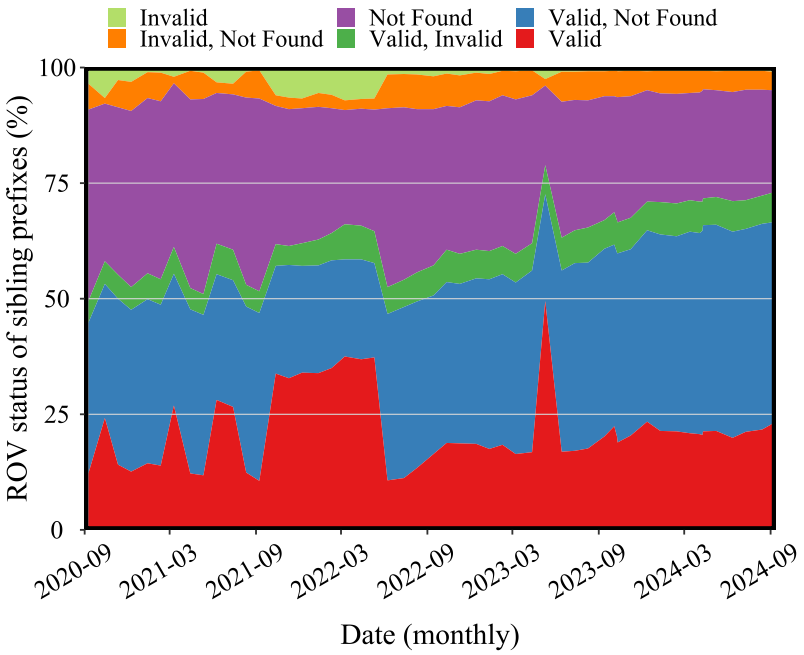

Figure 5.23: Route Origin Validation status of sibling prefix pairs in the RPKI data over time.

In summary, for around 65% of sibling prefixes, at least one sibling prefix has a valid ROV status. Still, for nearly 10% of sibling prefixes, at least one prefix has an invalid ROV status.

The decrease in number of not-found ROV cases over time, is in line with the overall increase in RPKI deployment. Creating valid ROV status for both pairs of sibling prefixes is essential to protect sibling prefixes and services running on those prefixes from routing incidents like BGP hijacks.

5.5 Discussion

Sibling prefix input dataset: In this research, we introduce a technique to identify sibling prefixes in the Internet. While our analysis uses domain similarity as input to identify sibling prefixes, we argue that we can identify sibling prefixes using other services, such as DNS MX records, rDNS names, or aliased hosts. We cross-check our findings with real world deployments of RIPE Atlas probes as a "ground-truth" dataset, finding a high overlap. Additionally, we perform port scans and find correlations between sibling prefixes identified using DNS and port scan datasets. An even larger and more comprehensive input dataset might allow to further increase the number of identified sibling prefixes in the Internet.

Domains instead of addresses: Moreover, some network architectures have shifted away from owning dedicated IP addresses for their services, instead sharing the same IP prefixes for their services and across multiple data centers [132, 182]. Therefore, identifying sibling prefixes in the Internet will help network operators around the globe to make more informed decisions, such as applying similar routing policies for a set of domains running on IPv4 and IPv6 as sibling prefix pairs. Sibling prefixes enable to allow, block, filter, or otherwise differentiate a group of domains collectively based on their prefixes. One example application of sibling prefixes is the adaption of IPv4 spam blocklists to IPv6, which closes the backdoor for spammers to switch to IPv6 if they are blocked on IPv4. Furthermore, a list of sibling prefixes with high Jaccard values could help the research community to more easily adapt their IPv4 analysis technique to IPv6. Therefore, we plan to regularly publish a list of sibling prefixes to be used by network operators and researchers at sibling-prefixes.github.io.

5.6 Related Work

Previous studies explored various aspects of siblings, alias resolution, fingerprinting techniques, IP addresses, and device detection on the Internet, focusing on the IP address level rather than on the prefix level.

Server siblings: In 2013, Berger et al. [21] studied the associations between Internet DNS client resolver IPv4 and IPv6 addresses. The study uses passive and active techniques to identify 674k associated address pairs. In 2015, Beverly and Berger [22] identified pairs of IPv4 and IPv6 server addresses potentially assigned to the same physical machine as siblings using TCP-reachable devices. In 2016, Czyz et al. [71] explored potential policy discrepancies between IPv4 and IPv6 dual-stack hosts and found that a ports are nearly always more open in IPv6 compared to IPv4. In 2017, Scheitle et al. [245] identified pairs of server IPv4 and IPv6 addresses as siblings using TCP timestamps. They used manually crafted algorithms and machine-learned decision trees to classify pairs of IPv4 and IPv6 server addresses as siblings, i.e., running on the same machine. In 2021 and 2023, Albakour et al. [8, 9] explored Simple Network Management Protocol (SNMP) v3, Secure Shell (SSH), and BGP for remotely fingerprinting network infrastructure in the wild. By sending unsolicited and unauthenticated SNMPv3 requests, the authors identified the status of network devices, including device vendor and uptime.

Alias resolution and fingerprinting: In 2012, Keys et al. [152] introduced MIDAR, a tool leveraging the IP ID for alias resolution. In 2013, Luckie et al. [173] proposed Speedtrap, a technique that induces fragmented IPv6 responses from router interfaces in a particular temporal pattern to for router fingerprinting. In 2015, Padnamabhan et

al. [207] introduced UAv6, a new alias resolution technique for IPv6. They probed the unused addresses using ICMPv6 Address Unreachable responses for IPv6 alias resolution. In 2016, Beverly and Berger [23], focused on discovering router-level IPv6 topologies. They used fingerprinting-based IPv6 alias resolution techniques to induce fragmented responses from IPv6 router interfaces. In 2020, Vermeulen et al. [283] presented a tool called "Limited Ltd." which exploits ICMP rate limiting for alias resolution.

Address assignment: In 2020, Padnamabhan et al. [208] looked into the IPv6 address assignments in various networks and how they relate to IPv4 dynamics. They found IPv6 assignments having longer durations than IPv4, often remaining stable for months. In 2023, Rye and Levin [242] leveraged NTP pools to gather 7.9 billion IPv6 addresses from NTP clients, examining the potential benefits and harm of larger IPv6 hitlists and the possibility of measurement and analysis. In the same year, Hsu et al. [125] identified the most common delegation prefix length by RIRs being /32, and the most prevalent BGP prefix length in IPv6 being a /48.

Prefix lists: The work most closely related to ours is by Naab et al. [194], who introduced the concept of prefix top lists in 2019. After aggregating domain-based top lists into network prefixes, they leveraged a Zipf distribution to assign weights to each prefix. They found that different domain-based top lists provide differentiated views on the Internet prefixes with minimal weight change over time.

To the best of our knowledge, our work is the first to investigate sibling relationships between IPv4 and IPv6 on a prefix-level.

5.7 Summary

In this chapter, we provided a thorough analysis of sibling prefixes in the Internet by applying the Jaccard similarity index on dual-stack domains. We introduced the SP-Tuner algorithm that identifies and fine-tunes IP prefix sizes resulting in tailored sibling prefixes having higher Jaccard values and smaller CIDR sizes. With SP-Tuner we were able to improve the share of perfect match sibling prefixes from 52% to 82%. We found that more than half of sibling prefixes belong to the same organizations, with the majority being associated with IT companies. We found sibling prefixes to be relatively stable over time and to be prevalent in 24 hypergiant and CDN networks. Sibling prefixes showed similar RPKI adoption rates as other prefixes, we did however find instances with inconsistent ROV status between IPv4 and IPv6 prefixes of sibling pairs. Finally, we plan to regularly publish sibling prefix lists for researchers and operators.

Chapter 6
RPKI Synchronization Delay

In this chapter, we study the causes of slowness in the Resource Public Key Infrastructure synchronization, including the impact of using a single prefix (as recommended) and multiple prefixes per ROA object. We identify the bottlenecks in RPKI synchronization and propose techniques to optimize RPKI synchronization delay. The Resource Public Key Infrastructure [36, 168, 179] is designed to improve Border Gateway Protocol security by providing a cryptographically verifiable association between the IP prefixes and Autonomous Systems that are authorized to advertise these prefixes. These associations are recorded in ROA objects and are signed by a Certificate Authority (CA). Each of the five Regional Internet Registries operates a Trust Anchor (TA) and a Publication Point (PP) to host the RPKI objects.

Network operators use Relying Party software to download the RPKI data from a globally distributed set of Publication Points or repositories, cryptographically verify the ROAs, and provide the Validated ROA Payloads (VRPs). We refer to all these steps as the RPKI or Relying Party synchronization process. The updated set of VRPs from the RPKI synchronization process is then provided to the RPKI-enabled BGP-speaking routers to help them filter out the illegitimate routes originated by rogue ASes from incoming BGP announcements.

Network operators have documented operational issues such as long delays for changes in RPKI to take effect [48–50]. The time it takes for an announcement to propagate globally in BGP is fast [95, 133, 296]. However, changes in RPKI are slow and might lead to scenarios where BGP accepts outdated or invalid announcements, potentially leading to service disruptions.

RPKI delay has three main components: (1) the *publication delay*, which is the time required for a change in RPKI data to be publicly visible in Publication Points, (2) the *synchronization delay*, which is the time Relying Party software will take to fetch, validate and update its list of Validated ROA Payloads, and (3) the *BGP refresh interval*, which is the time interval at which RPKI-enabled routers refresh their list of VRPs perform Route Origin Validation and update their Forwarding Information Base [36].

Fontugne et al. [90] analyzed the end-to-end delay by observing the reaction time of BGP after changing ROA information at the five RIRs. More precisely, they show that the

highest cause of delay between RPKI and BGP comes from the synchronization of RP software but they provide little details on why this step is slow.

The RP synchronization delay is composed of three main elements: (1) RPKI data downloading time, (2) RPKI data validation time, and (3) the time it takes to produce the list of VRPs. The first part requires communication between RP software and Publication Points.

As RPKI is gaining importance, and its adoption is growing [196], the download and process time of RPs is also expected to increase over time due to an increasing number of CAs, ROAs, and PPs incurring more delays. The increasing fetch time for retrieving a full snapshot of the RPKI tree may hamper further deployment. In this chapter, we study the dynamics of fetching the RPKI tree from the PP hierarchy, examine the cryptographic verification of ROAs, and propose optimization techniques to enhance the RP synchronization delay. Our contributions are as follows:
In-depth analysis of RPKI synchronization delay: We analyze each TA's structure of publishing ROAs and the impact of mode of operation (hosted vs delegated) in RPKI data publication. Our results show that consolidating several prefixes of an AS into a single ROA can reduce the cryptographic verification time by a factor of up to three. While the fetching and validation of ROAs are the main delay factors in RP synchronization, we further identify that 7% of VRPs in delegated PPs accounts for up to 90% synchronization delays for one of the TAs (see §6.1).

Large-scale latency measurement: We perform large-scale active measurements using more than 700 RIPE Atlas anchors from 91 countries around the globe to identify the accessing delay to RPKI PPs. Most PPs are accessible within 50 milliseconds of RTT from their primary serving regions but require hundreds of milliseconds from other regions. Accessing RPKI PPs using IPv4 or IPv6 has a similar delay, but the RPKI Resource Delta Protocol (RRDP) deployment of PPs is accessible faster than PPs using the rsync protocol. We show that for every additional RTT of 100 milliseconds, RPKI synchronization increases up to 25 seconds (see §6.2).
We identify the root causes of RP synchronization delay and discuss possible optimization techniques for PP maintainers, RP software developers, and RP synchronization modes of operation.

6.1 Measuring RP Synchronization Delay

RP synchronization delay can occur in several parts of the process, including network delay of downloading data from RPKI Publication Points, cryptographic verification at the client by following the certificate chain, and production of VRPs. In this section, we thoroughly analyze delay factors that impact the RPKI validation process and try to identify potential delay areas for optimization.

6.1.1 Experimental Setup

To measure the Relying Party software synchronization delay, we use dedicated hardware located in Germany, running Debian 11 with an AMD 8-core processor, maximum speed 2.1GHz each, 16 threads, 31 GiB of RAM, and 2.0 GiB of Swap memory. We connect the machine directly to the Internet with no firewall in between via one Gbps link. We performed a similar analysis from a second vantage point in Japan using a dedicated

Virtual Machine (VM) and obtained similar results. In this chapter, we present results of dedicated hardware only, which is less biased by the system's underlying infrastructure compared to the VM.

Relying Party Software: We use *rpki-client* version 9.3 [85], and *Routinator* version 0.14.0 [163] as the actively maintained, popular in the RPKI community, and commonly used RP software in practice [158]. Using two different RP implementations, we identify implementation-specific delays and separate them from RPKI infrastructure-caused delays. Our goal is to understand the bottlenecks in provisioning RPKI objects and their impact on delay.

We use the default configuration for both RP software to produce a list of VRPs and always run only one instance of one software at a time during the experiment to avoid any competition in terms of network and CPU resources.

In practice, network operators start by pulling and validating the entire RPKI tree and then perform synchronization of local RPKI cached data by fetching updates only. We refer to the first case as the Cold cache and the second case as the Warm cache. To thoroughly analyze the RPKI synchronization delay, we also introduce a third case we call Hot cache RPKI synchronization. We define the Cold, Warm, and Hot modes for RPKI cached data as follows:

Cold cache: We manually delete all the RPKI cached objects from our system and then start the RPKI synchronization with an empty cache. Clearing the cache before running the RP software allows us to measure the entire time RPs take to fetch data, process it, and produce the list of VRPs. For RP operations in Cold mode, we select the one-hour interval aligned with rpki-client man pages [216].

Warm cache: RPs reuse the previously downloaded RPKI data. However, they still have to synchronize with RPKI repositories to update to the latest data, run validation, and produce the VRPs. The cached RPKI data in our experiment is around 10 minutes old, aligned with Routinator's default policy [164], for each Warm mode RP synchronization.

Hot cache: RPs process the cached data only and do not download any data from RPKI repositories. While Hot mode is not used for operational settings, we use this mode to exclude the network delay and provide the exact delay required to perform cryptographic verification or validation of RPKI data and produce the final output set of VRPs.

As we report on timing values, we consider the impact of CPU cache on our analysis. A full or partial existence of RPKI data in RAM, swap/virtual memory, or buffer memory, which we refer to collectively as the CPU cache memory, may influence our analysis. To handle the CPU cache memory impact, for each of the RP operation modes, we consider two states of CPU cache retained or True where a '(T)' is appended to the name of RP software like *Routinator (T)* or *rpki-client (T)*, and the CPU cache cleaned or False with '(F)' appended to RP name, for example, *Routinator (F)* and *rpki-client (F)* in the rest of this chapter. For each mode of RPKI cache in Cold, Warm, and Hot, and every status of CPU cache being True or False, we perform 100 iterations of each RP software synchronization for 15 consecutive days. We run the same configuration for the two RPs as close in time as possible, but only one instance at a time. We collect each experiment's detailed logs and output, constituting this section's main dataset.

6.1.2 Synchronization Delay of Relying Party Software

We define the RP synchronization delay as the time from the start to the end of one RPKI validation process where RP software contacts RPKI Publication Points, downloads

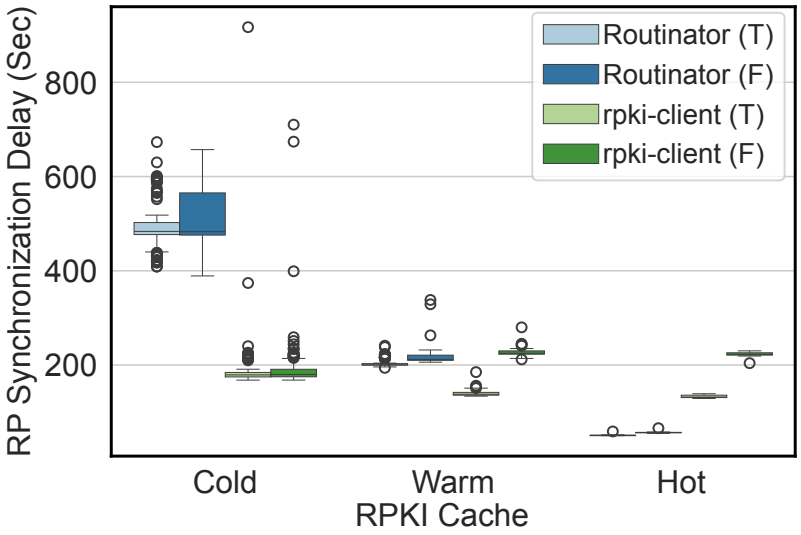

Figure 6.1: Boxes show the RPs synchronization delay for Cold, Warm, and Hot RPKI cache modes. The order of the boxes matches the legend.

data, performs cryptographic verification of data, and produces a set of VRPs (excluding synchronization with the router via RTR). The Hot RPKI cache mode does not have any network involvement, and the RP synchronization delay reflects only the time required to cryptographically verify the RPKI data from the system's local cache and produce VRPs. We use the terms RP synchronization and RPKI validation interchangeably throughout this chapter.

In Figure 6.1, Routinator has the highest synchronization delay in Cold mode with a median value of around 490 seconds for both cases of CPU cache True and False. For rpki-client (T) and (F) in Cold mode, the median delay is around 190 seconds. Using Routinator in Cold mode has a major page fault rate of around 455k. When RP software accesses a memory page that is not in physical RAM, a major page fault occurs, and the operating system then fetches the page from the disk. Because accessing disk storage is slower than RAM, major page faults create delays for Routinator. For rpki-client, we identify zero to 20 major page faults. In Cold mode, neither RP is sensitive to CPU cache status as all the data should be downloaded from the Internet.

For the Warm mode, Routinator (T) has a median RP synchronization delay of around 200 seconds and 210 seconds when the CPU cache is False. rpki-client (T) and (F) have a median value of around 140 seconds and 240 seconds, respectively. We believe the increase in synchronization delay for rpki-client (F) is due to high read from the file system, showing rpki-client is more sensitive to the CPU cache than Routinator. As expected, the Warm mode has a lower RP synchronization delay compared to the Cold mode because less data has to be downloaded.

We see the highest outlier of RP synchronization delay for rpki-client in Cold mode because it has a default timeout of 900 seconds to switch from RRDP to rsync protocol [216]. We identify the *rpki.cnnic.cn* repository containing 1,218 VRPs (1,117 IPv4, 101 IPv6) registered for 173 ASes as the problematic PP causing the highest outlier. To avoid delays caused by outliers, skipping the problematic PPs, especially when the local copy exists, might be an option. Stalling the RPKI synchronization process by a PP is a vulnerability

also reported in other studies [282] [92].

Routinator (T) and (F) have the lowest median delay of 40 and 50 seconds, respectively, in Hot mode. The rpki-client (T) has a higher median delay of more than 100 seconds and more than 200 seconds when the CPU cache is False or rpki-client (F). The Hot mode indicates that Routinator has a faster cryptographic verification of the RPKI data than rpki-client. However, the rpki-client has a better synchronization time when the network is involved in the RP synchronization process. Cold mode is write-intensive, with nearly 4 million write operations to the file system. Inversely, the Warm and Hot modes are read-intensive, with more than 4 million read operations for Routinator (F) and rpki-client (F) only when the data is unavailable in the CPU cache.

We run the same set of experiments using a memory-mapped file system for the RPKI data storage to understand the potential impact of read and write to the disk file system on RP synchronization. Using a memory-mapped file system mitigates the impact of the system's CPU cache and further reduces the RP synchronization delay by up to 8%. This is because the read/write from and to memory-mapped file systems are faster than the disk-mapped file system.

For Routinator, the significant delay difference between Hot mode and the other modes indicates that fetching RPKI data from the Internet is the main contributor to synchronization delay. In contrast, the rpki-client exhibits a different behavior of taking more time to process the cryptographic verification of ROAs during synchronization. The use of RPKI cache is more efficient for Routinator, reducing synchronization delay to nearly 200 seconds compared to not using cached data, specifically in settings where a refresh occurs every 10 minutes. However, the RPKI cache has a few seconds of improvement for the rpki-client only when the CPU cache is retained. The rsync protocol implementations with rpki-client using openrsync and Routinator using the system-provided rsync might also contribute to variations in synchronization delay in Cold mode [85, 163].

Routinator has the fastest cryptographic verification of RPKI data and is more efficient in handling empty CPU cache status, but suffers from a high number of page faults, causing the highest RP synchronization for the software in Cold mode. In contrast, rpki-client takes nearly twice as much time as Routinator for the cryptographic verification of the data, but has efficient network handling and lower synchronization delay than Routinator in Cold mode with almost no page fault. It is also more prone to the system's CPU cache state. We think combining Routinator's fast cryptographic verification and robust handling of CPU cache with rpki-client's efficient network handling in an RP can significantly reduce RP synchronization, and running them on a memory-mapped file system can even further optimize the RPKI synchronization process.

Typically, Routinator runs as a daemon, and rpki-client runs by cron, in Warm mode, and stores its cache state on disk; this means these RPs will only run in Cold mode once after first installation on a system, or after an explicit wipe of the on-disk RPKI cache by an operator. In normal operations, and because RPs typically run in Warm mode, rpki-client and Routinator will have a very similar synchronization delay in day-to-day performance.

In summary, we identify network delay being the major delay contributor for Routinator and the cryptographic verification of RPKI data as the major delay contributor for rpki-client software. Routinator has the fastest cryptographic verification of RPKI data and is more efficient in handling empty CPU cache status, but suffers from a high number of page faults, around 455k, causing the highest RP synchronization for the software in Cold mode. In contrast, rpki-client takes nearly twice as much time as Routinator for the cryptographic verification of the data but has efficient network handling and

lower synchronization delay than Routinator in Cold mode with almost no page fault. It is also more prone to the system's CPU cache state. We think combining Routinator's fast cryptographic verification and robust handling of CPU cache with rpki-client's efficient network handling in an RP can significantly reduce RP synchronization, and running them on a memory-mapped file system can even further optimize the RPKI synchronization process.

6.1.3 Impact of ROA Composition

We want to understand whether the delay for cryptographic verification of ROAs for all five RIRs is the same or different. Measuring each RIR's RPKI tree individually provides us with a clearer picture of the performance bottlenecks at each RIR. For this purpose, we measure the RP synchronization delay in Hot cache mode for the entire RPKI tree of each RIR (i.e., the TA and all its delegated repositories). The Hot mode is not biased by our machine's location, as only the RPKI cached data is analyzed, and network delay is not involved. Since each TA's RPKI tree maintains a vastly different number of ROAs and VRPs, we normalize the measured delay per VRP and the average number of VRPs per ROA.

Figure 6.2 (a) illustrates a scatter plot of the average number of VRPs produced per second on the x-axis and the average number of VRPs per ROA object on the y-axis for each RIR using Routinator and rpki-client. The results show that the RPKI tree of RIPE and APNIC has the fastest processing time per VRP, more than 10k and 9k VRPs per second, respectively, using Routinator or more than 3k using rpki-client. The rest of the TAs have less than 3k VRPs per second processing time using rpki-client and less than 6k using Routinator. AFRINIC has the lowest number of 2.3k VRPs per second using Routinator, and LACNIC has the lowest number of 1.8K VRPs using rpki-client. Cryptographic verification of ROAs for the same amount of VRPs from ARIN, LACNIC, and AFRINIC RPKI tree requires at least twice as much time as APNIC and RIPE using Routinator. The VRP production rate, number of VRPs per second, and the difference between Routinator and rpki-client are due to Routinator being fast in Hot mode and even faster when the data amount is higher, as shown for AFRINIC and ARIN. This difference between RIRs is mainly explained by the bundling of multiple prefixes in a single ROA object, a common practice for RIPE and APNIC but not the other RIRs. ROAs from ARIN, AFRINIC, and LACNIC RPKI trees produce, on average, around one VRP. In contrast, the RIPE and APNIC RPKI tree ROAs contain, on average, 3.6 and 5.6 VRPs. Consolidating several prefixes for one authorized AS into a single ROA object reduces the number of cryptographic operations required per VRP and, hence, significantly reduces the delay observed for processing RPKI data.

Figure 6.2 (b) displays the distribution of prefixes per AS extracted from VRPs for each RIR. Around 60% to 85% of ASes have more than one IP prefix per AS in each TA's RPKI tree. In the range of 10% to 18% of ASes have more than ten prefixes per AS, with Amazon AS16509 having the highest number of 10,879 IP prefixes in RPKI data.

All ASes with more than one prefix registered in RPKI are candidates for bundling their multiple prefixes into a single ROA object. We discuss the potential side-effects of bundling in §3.6.

In summary, bundling several prefixes of an AS into a single ROA produces up to three-fold higher number of VRPs in the RPKI validation process for RIPE and APNIC, compared to other RIRs RPKI that compose one ROA for each IP prefix. More than 60% of ASes, register more than one IP prefix in RPKI, indicating the potential of bundling all

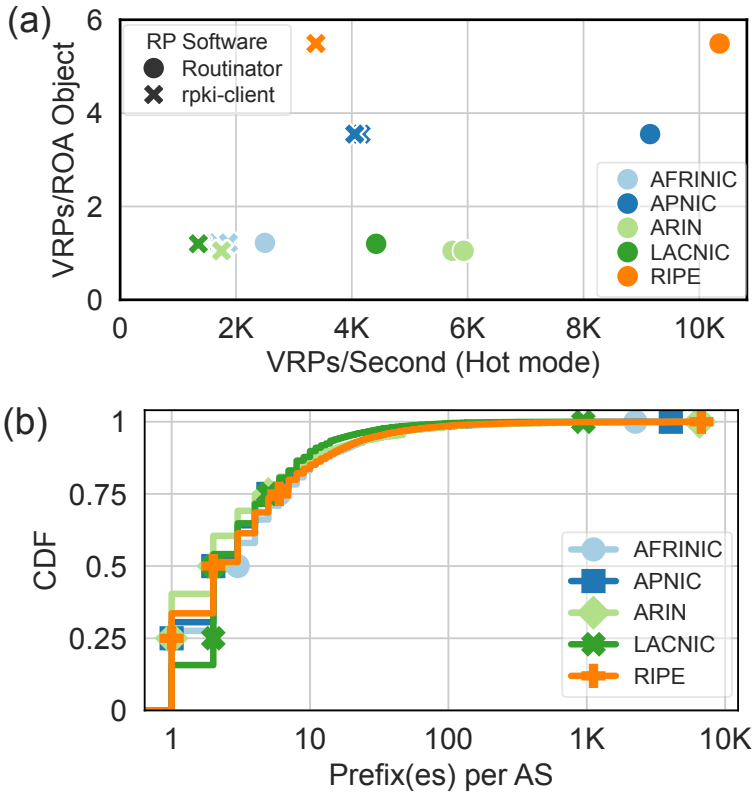

Figure 6.2: For each RIR's RPKI tree (a) shows the average rate of RPKI data validation and (b) shows the distribution of IP prefix per AS.

prefixes of an AS into a single ROA object to improve the validation time.

6.1.4 Impact of Certificate Chain Depth

In the RPKI, certificates are organized in a hierarchical mode. Figure 6.4 illustrates certificate chain structure for RPKI. The topmost certificates in RPKI are the RIRs' Trust Anchors (root certificates) (Depth 1). For different operational reasons, RIRs can create intermediate child certificates to manage their CA, AFRINIC and LACNIC use depth of 2, and the rest of the RIRs use a depth of 3 for this purpose. They also create certificates (member certificates) for or to National Internet Registries or Local Internet Registries for organizations to which resources are assigned (Depth 4). LIRs and NIRs can issue End-Entity certificates (Depth 5), which are used to sign ROAs. LIRs and NIRs can sub-allocate resources, this adds another level to the depth of the certificate.

RP synchronization needs a new TCP connection to be established for each Publication Point. This process involves numerous handshakes, making it crucial to ensure fast downloads of RPKI snapshots or updates from these delegated CAs. We study the impact of the depth of the certificate chain or delegated Certificate Authorities on RP synchronization delay. For this particular measurement, we use only Routinator as it is not possible to measure CA depth using rpki-client. Routinator has a configurable chain depth to mitigate certain denial-of-service attacks [282] with a default CA depth value

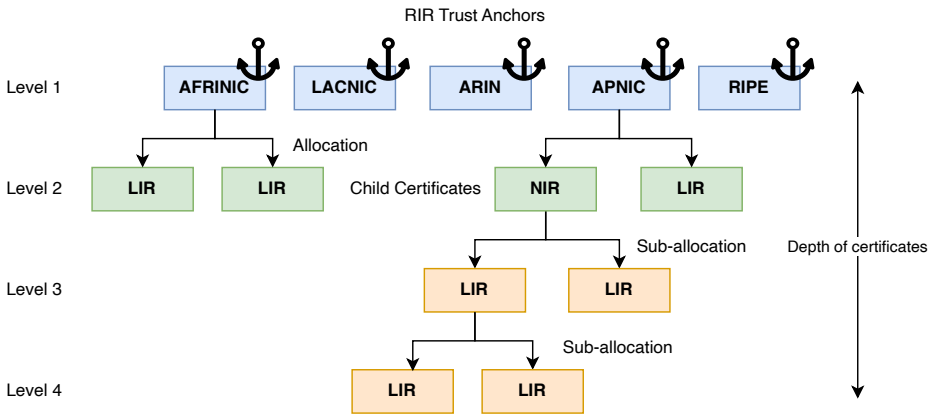

Figure 6.3: Illustration of RPKI certificate hierarchy and depth of certificates.

32. We use this feature Routinator to perform RPKI synchronization in Cold mode with CA depths of 2 to 6 for each RIR's RPKI tree, as other depths contain no ROAs and 50 iterations at each depth and for every RIR.

In Figure 6.4 the x-axis shows the maximum depth of the certificate chain of trust (i.e., a maximum depth of 4 includes all levels up to 4). The y-axis represents the delay in seconds (top subplot) and the number of VRPs (bottom subplot). We show the RP synchronization delay for each CA depth for APNIC, ARIN, and RIPE in the top subplot of Figure 6.4 and the corresponding number of VRPs at each depth in the bottom subplot. We exclude AFRINIC and LACNIC from this analysis because they take less than 100 milliseconds, not affecting the total delay of RP synchronization. Moreover, AFRINIC has no CA depth, and LACNIC has only one additional depth than TA, and only for the mentioned RIRs, we observe data at a CA depth of 2. For APNIC, the number of VRPs are increased from 119k at a CA depth of 3 to 137k at a CA depth of 5, shown in the bottom subplot of Figure 6.4. However, the CA depth impact on RP synchronization delay is minimal for APNIC, with all depths having less than 100 seconds of delay. The *rpkica.twnic.tw* PP at CA depths of 3 and 4, and *rpki.sub.apnic.net* PP causing outliers with increased RP synchronization delay from 200 to nearly 400 seconds. Delegated CAs of APNIC contribute 13% additional VRPs and add only around 2% extra delay, which means they have a positive impact and help RP software validate more data in relatively less time.

The maximum certificate chain depth clearly impacts RP synchronization delays for ARIN. We notice a median of 184 seconds RP synchronization delay at a CA depth of 3, which increased to a median of 360 seconds at a depth of 4 and is slightly higher at the CA depth of 5. Processing 7% of VRPs from ARIN's delegated CAs cause more than 90% delay for the entire ARIN RPKI resources in Cold mode.

Moving deep from CA depth of 3 to 4 for RIPE CA depth, the RP synchronization delay has medians of 304 and 305 seconds, which is in line with the slight increase of VRPs from 268k to 270k, as shown in the bottom subplot. However, the delegated CAs at a depth of 5 that contributes nearly 320 VRPs increase the RP synchronization from 304 to 330 seconds. This means around 1% of delegated RPKI data causes up to 10% additional delay for RIPE.

These delays reflect the fact that a small number of ROAs from delegated CAs may inflate

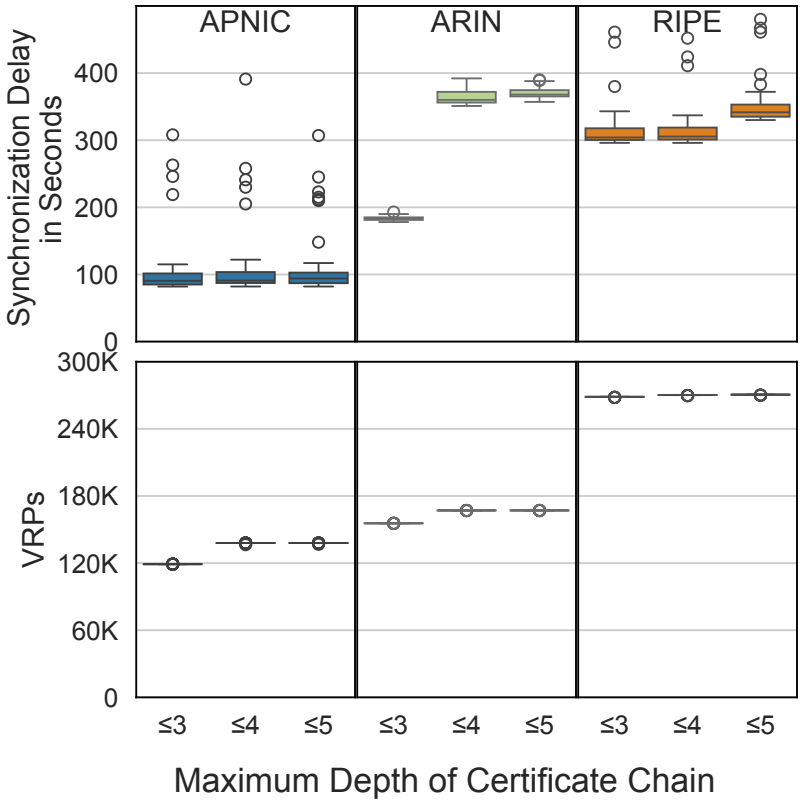

Figure 6.4: Impact of CA depth on TA's RP synchronization delay (top plot) and their contribution of VRPs (bottom plot) using Routinator in Cold mode.

the delay observed for the whole RPKI tree of a TA. While delegated CAs hosted on cloud infrastructure positively impact the RP synchronization for APNIC, a few specific delegated CAs cannot always serve RPKI data using RRDP and cause RPs to hang for hundreds of seconds on them, which can slow down the entire RP synchronization process. Analyzing the RP synchronization logs, we find that sometimes repositories cannot serve the RPKI data using RRDP, they time out, and then fall back to rsync. This timeout and fallback to rsync is the common issue for those repositories creating outliers with high RP synchronization.

RPs download and process delegated CA objects after the initial delay incurred in fetching and processing TA data at the top of the certificate chain. Any slowdown in the certificate chain's depth for delegated CAs can significantly prolong the entire RPKI synchronization process. Therefore, the quick and efficient operation of delegated CAs is even more important. Maintainers of particular delegated CAs may take action to ensure smooth and fast serving of RPKI data to avoid hanging of RP synchronization process on their repositories. Fast operation of delegated CAs is crucial in RPKI synchronization, and even more in the future as their number may increase.

Figure 6.5 illustrates the RPKI data validation rate or the number of VRPs validated per second at each CA depth. For APNIC, delegated CAs positively impact RP synchronization and increase the VRP validation rate from 1300 VRPs to almost 1500 VRPs per second. For ARIN, the delegated CAs reduce the RPKI validation by almost half,

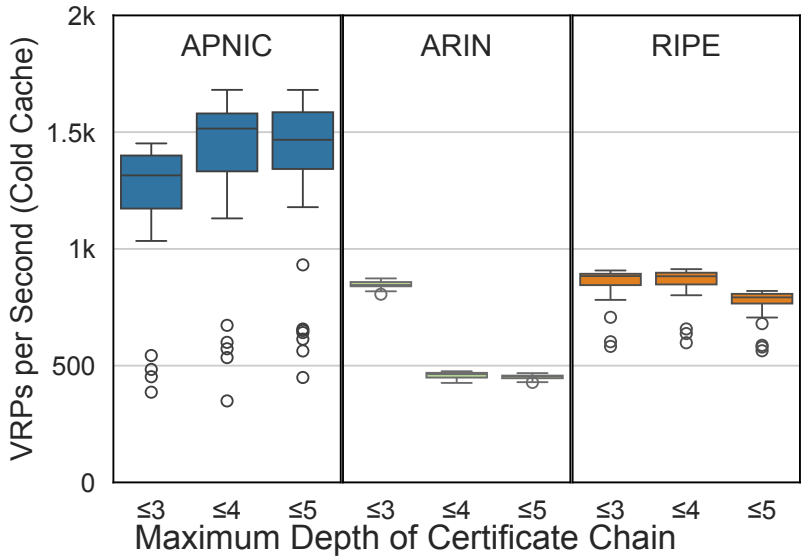

Figure 6.5: RP synchronization rate of VRPs per second at various Certificate chain depths of TAs.

from 800 VRPS per second at a CA depth of 3 to around 450 VRPs at depths of 4 and 5. The RIPE delegated CA at depth reduces the validation rate of VRPs from 800 to nearly 750 VRPs per second. The impact of outliers, as explained in Figure 6.4, reducing the validation rate of RPKI data is visible here as well.

In summary, processing 7% of VRPs from ARIN's delegated CAs cause more than 90% delay for the entire ARIN RPKI resources in Cold mode. Almost 1% of VRPs hosted at CA depth of 5 cause nearly 10% extra delay for the RIPE RPKI resources. These delays reflect the fact that a small number of ROAs from delegated CAs may inflate the delay observed for the whole RPKI tree of a TA. While delegated CAs positively impact the RP synchronization for APNIC, we notice a few specific delegated CAs that cannot always serve RPKI data using RRDP and cause RP to hang for hundreds of seconds on them, which can slow down the entire RP synchronization process. Fast operation of delegated CAs is crucial in RPKI synchronization, and even more in the future as their number may increase.

6.2 RPKI Delay Measurement using RTT

RPKI data contains nearly 560k small objects stored in distributed repositories or Publication Points worldwide. Downloading RPKI data requires exchanging around 140k packets between the RP software and RPKI Publication Points. Therefore, understanding the Round Trip Time to access RPKI resources or PPs from around the globe is crucial to the RP synchronization delay.

6.2.1 Measurement Design and Data

We perform an active measurement using all the RIPE Atlas anchors by running a series of TCP-based traceroutes toward all the RPKI Publication Points for two weeks.

We use around 700 RIPE Atlas anchors located in 91 countries for our experiment because they are usually hosted in data centers or in well-provisioned locations like universities. Hence, the delay results observed from RIPE Atlas anchors from a particular network are arguably a good approximation of the delay experienced by Relying Party software in that network for the RPKI synchronization process.

We use a list of 69 PP hostnames for both RRDP and rsync PPs extracted from the Authority Information Access (AIA) field of all RPKI certificates of each TA. We mimic the two protocols by running TCP-based traceroute (both rsync and RRDP are using TCP), on port 443 for RRDP and port 873 for rsync measurements. We could use ping, but ICMP might be handled/prioritized differently by network operators than real traffic.

For traceroute measurements, we use both IP versions to check if there is a difference in that regard. Traceroute packets are 48 bytes in size (i.e. size of the data part of the packet and excluding any IP, ICMP, UDP or TCP headers), Time To Live (TTL) value of 32 (default), and a response time out value of 10000 milliseconds.

Moreover, we anticipated that path-level information might become useful for extended analyses or follow-up studies. The PP hostnames are resolved using the anchor's local DNS. We probe each PP hostname via IPv4 and IPv6 (three times per address) every five hours for two weeks—ensuring coverage of all hours and weekdays. We use data of traceroutes reaching the correct destination IP. These include 3,893,594 traceroutes to RRDP PP hostnames and 3,149,005 to rsync PP hostnames.

6.2.2 Measuring Delay of RPKI Publication Points

The violin plots in Figure 6.6 show the distribution of RTT delay for accessing RPKI rsync and RRDP PPs using IPv4 and IPv6 from each region globally. The x-axis shows the continent, the y-axis shows the RTT delay in milliseconds in log scale, the width of the violin reflects the relative density of data, and the dotted horizontal lines, within each violin, represent the data's quartiles.

We observe a similar distribution of RTT delay for both IP versions, with IPv6 having a slightly lower delay toward RRDP Publication Points for some regions. Overall, the violin plots show a significant peak above 100 milliseconds for most of the continent, except for skewed distribution data points, which are lower than 100 (10^2) milliseconds for Europe and North America, indicating a high concentration of data points. Generally, RRDP PPs are accessible faster than the rsync PPs globally.

Nearly 75% of data points from Africa and Oceania observe an RTT delay of over 300 milliseconds accessing RRDP and rsync PPs, and this number is around 50% for accessing RRDP PPs. RIR's PPs have the lowest RTT delay in their primary serving region, like AFRINIC PP being fast in African countries only and LACNIC being fast in South America and users accessing them from other regions experiencing a relatively high RTT of more than 300 milliseconds on average.

Europe and North America generally have the fastest access to RPKI resources, possibly due to most PPs being hosted in these regions. Africa, Oceania, and South American countries observe the highest RTT delay in accessing RPKI resources.

Eight of the RRDP PPs, including *rrdp.apnic.net*, *rrdp.ripe.net*, all using Content Distribution Networks such as Cloudflare, Akamai, Amazon, and Google, have lower than 10 (10^1) millisecond RTT delay for all regions.

Our findings reveal regional disparities in accessing RPKI resources. Understanding

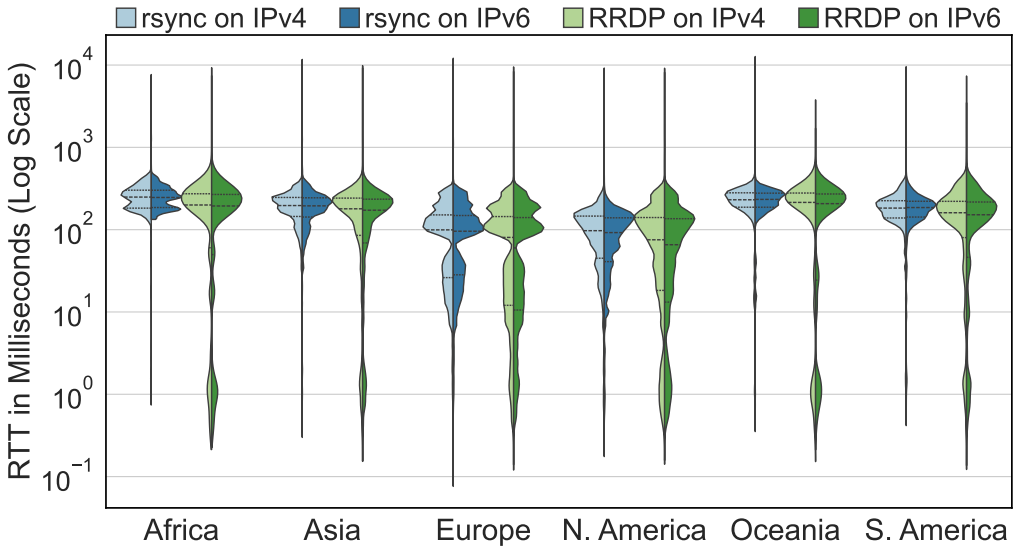

Figure 6.6: Violin plots showing the distribution of IPv4 and IPv6 RTT delay (in milliseconds, log scale) to RPKI rsync and RRDP Publication Point hostnames.

these disparities can inform strategic decisions in infrastructure deployment and resource allocation to ensure equitable access to RPKI services worldwide.

In summary, RRDP Publication Points are accessible faster than the rsync Publication Points globally. Europe and North America generally have the fastest access to RPKI resources, possibly due to most PPs being hosted in these regions. Africa, Oceania, and South American countries observe the highest RTT delay in accessing RPKI resources.

6.2.3 RTT Delay to RIRs' RPKI Resources

We use IPv4-only data and select RRDP-only repositories because they are the primary PPs for RP synchronization. If RRDP fails, the RP switches to the rsync protocol. The RTT delay in accessing RPKI PPs from IPv4 and IPv6 are similar. Europe and North America have faster access to RPKI resources, and RRDP PPs have lower delays than the rsync PPs (see §6.2.2 for details).

We measure the RTT delay in accessing each RIR's TA Publication Point and its delegated Publication Points from every continent. The violin plots in Figure 6.7 exhibit the RTT delay distribution for accessing RRDP PPs of each RIR's RPKI tree globally.

The shapes of violin plots reflect that for most of the cases (with few exceptions), each RIR's Publication Points, including the delegated ones, have a relatively lower accessing RTT in their primary serving continent and a high RTT delay from other regions. For example, the AFRINIC, ARIN, LACNIC, and RIPE have, on average, an RTT delay range of less than one to nearly 90 milliseconds for their primary serving continents, respectively, and a 100 to nearly 700 milliseconds RTT delay for other regions. The prime reason for this difference is that the PPs are hosted in those regions. We identify that a few of APNIC's delegated PPs use international cloud infrastructure. For example, *rpki.owl.net*, originally from Vanuatu islands of Australia, and *rrdp.rp.ki* using xTom [292] and Misaka Network [191] cloud infrastructures, respectively, that has data centers in

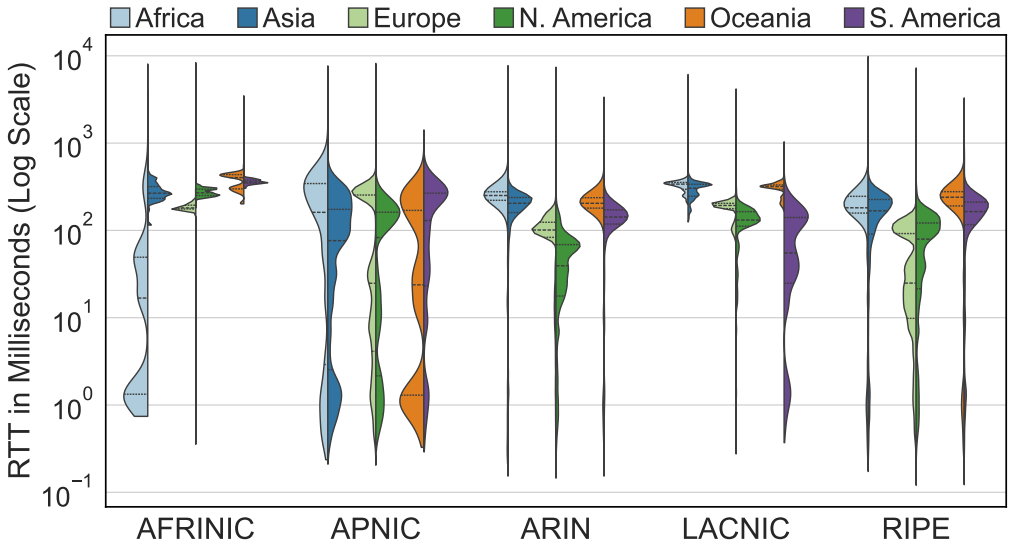

Figure 6.7: Violin plots showing the distribution of RTT (in milliseconds, log scale, IPv4) in accessing RPKI RRDP Publication Point hostnames of each RIR's RPKI tree.

Oceania, North America, and Europe causing the lower bump of high-density RTT smaller than 10 milliseconds for these regions.

AFRINIC PP has more than 200 milliseconds RTT delay for other continents, except Africa, with Europe being slightly lower than the rest. Europe also has the lowest RTT delay in accessing APNIC RPKI tree Publication Points, with North America and Asia being the second and third with the lowest RTT. After North America, which has the lowest RTT delay to ARIN's RPKI tree PPs, Europe and South America are second and third with the lowest RTT delay. Africa observes a high RTT of 300 to 800 milliseconds when accessing ARIN's PPs. While Asia, Africa, and Oceania observe a high RTT delay in the range of 400 milliseconds to 700 milliseconds, the vast majority of probes from Europe and North America have RTT delay in the range of 100 milliseconds to 500 milliseconds RTT toward LACNIC PPs. The RIPE PPs have the second lowest RTT, the first being Europe, for North America, with nearly 50% RTT delay being below 100 milliseconds. Asia, Africa, and South America have a similar pattern of RTT delay, with more than 100 milliseconds on average for nearly 75% of the data. Oceania has the highest range of RTT delay of above 200 milliseconds towards RIPE PPs for more than 50% of the data.

For both APNIC and RIPE RPKI PPs, we observe densities of data with RTT of less than 10 milliseconds for other regions apart from their primary serving zone. Using the IP2ASN [223] and BGPtools [279], we identify CDN service providers that serve the fastest PPs in RPKI.

Help of CDN: To quantify the added benefits of serving PPs from a CDN, we compare IPv4 RTTs for the two largest RIR PPs: RIPE's CDN-hosted PP *rrdp.ripe.net* and ARIN's *rrdp.arin.net* self-hosted PP. Here, we consider only RRDP hostnames since rsync cannot be served by CDNs. Figure 6.8 illustrates the continent-level distribution RTT delay for accessing RIPE and ARIN's PPs. ARIN's PP has, on average, a high RTT delay of more than 100 milliseconds, peaking at 300 milliseconds, for more than 90% of the data from all continents except North America. Accessing RIPE PP has a much lower RTT, peaking at lower than 10 milliseconds, with some around 1 milliseconds, for all the continents. A

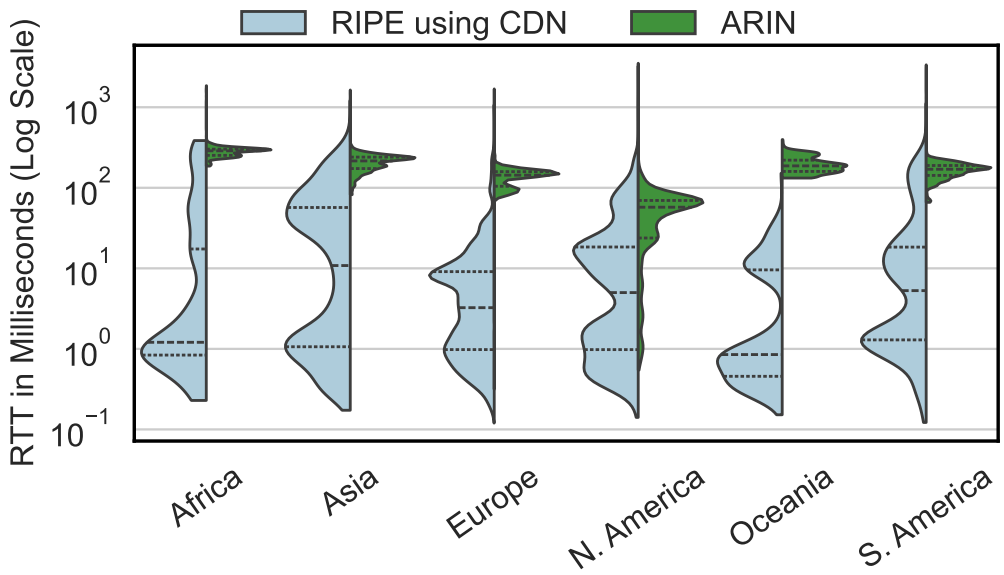

Figure 6.8: Violin plot showing the distribution of RTT (in milliseconds, log scale) to RPKI Publication Point hosted on CDN vs. None-CDN, from continents.

few networks with high RTT delays shown on the upper part of the violin plot might be due to the slow local infrastructure at a particular time or the impact of routing policies leading traffic to a CDN location with a high RTT. a high RTT. More than 75% of the RTT data on accessing ARIN and RIPE PPs from regions around the world, apart from their primary serving regions, show 100 milliseconds RTT delay for ARIN PP and less than 10 milliseconds RTT delay for RIPE PP. This indicates RIPE PP is accessible ten times faster for 75% of the cases.

Artificial Delay: To measure the impact of RTT delay on RP synchronization, we artificially add RTT delay for all the egress packets from the measurement machine in the network path and re-run the RP software to synchronize with RPKI. For Cold mode, every additional 100 milliseconds of RTT adds around 25 seconds and 15 seconds extra delay, respectively, to the RP synchronization process of Routinator and rpki-client. For the Warm mode, every 100 seconds of extra RTT delay adds around 5 seconds for Routinator and 3 seconds for rpki-client.

In summary, Publication Points of each RIR's RPKI tree are accessible faster in their primary serving regions than in other regions. PPs running on well connected CDN infrastructure have the lowest RTT delay of less than 10 milliseconds for users accessing them from other continents. AFRINIC and LACNIC PPs have relatively high RTT delays from other regions apart from their primary serving region. APNIC, ARIN, and RIPE PPs have a similar pattern with lower RTT delay compared to AFRINIC and LACNIC from other regions. Two of the APNICs delegated PPs that use cloud infrastructure are accessible faster in Europe and North America than in Asia because those Publication Points use cloud infrastructure that has data centers in Europe and Oceania. Our findings show that TA Publication Point of RIPE using CDN is ten times faster than ARIN for 75% of the users worldwide. Learning from PPs using CDN with les than 10 milliseconds, we believe there is room for reducing the RTT delay of all PPs by taking a similar approach. Every additional 100 milliseconds of RTT adds on average around 20 seconds extra delay

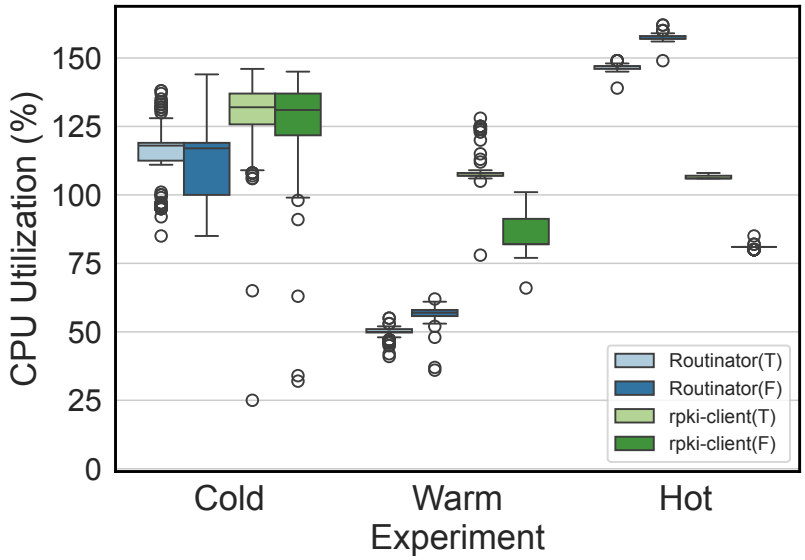

Figure 6.9: Box plot showing the median number of CPU utilization for cold, warm, and hot RP synchronization modes. The box sequence aligns with the legend.

to RPKI synchronization in Cold mode.

6.3 Resource Utilization

In this section we examine details of the resource utilization during RPKI synchronization for each of the RP software and each mode.

6.3.1 CPU Utilization

We collect Central Processing Unit (CPU) utilization for each second and all the 16 CPU cores of our measurement system during RP synchronization of Cold, Warm, and Hot modes. Box plots in Figure 6.9 depict the CPU's median utilization percentage for Routinator and rpki-client. We see that Routinator has the highest load of CPU utilization above 150% in Hot mode and the lowest median percentage in Warm mode. The high CPU utilization in Hot mode reflects Routinator's efficient use of CPU in Hot mode, performing the cryptographic data verification faster or doing more in less time. rpki-client has the highest CPU utilization of above 125%, being more efficient in Cold mode and relatively lower CPU utilization in Warm and Hot modes of RPKI synchronization. The impact of empty CPU cache causing high CPU utilization is visible in Warm and Hot modes for both RPs.

We deepen our study by examining where the CPU time is consumed during RP synchronization. Extracting the details of CPU utilization details of Figure 6.1 (a), the Figure 6.10 (a) depicts the CPU time for the input-output waiting time as 'iowait', software interruption handling time as 'soft', executing system or kernel codes time as 'sys', and executing user level code as 'usr'. The x-axis shows RPs with CPU cache status of True or False, and the y-axis represents the CPU time in seconds. For every RPKI cache mode

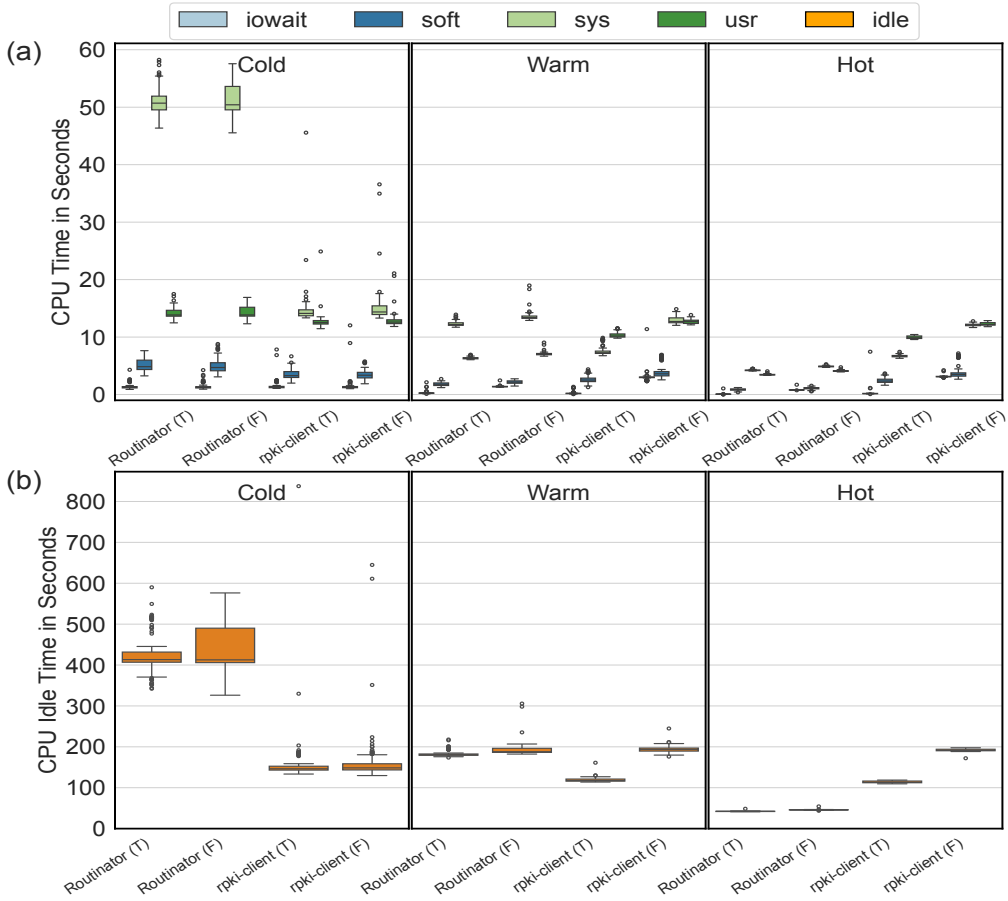

Figure 6.10: (a) Details of CPU time with (b) showing the idle time of CPU during RP synchronization. The order of the boxes matches the legend.

of Cold, Warm, and Hot in Figure 6.10 (a), we show their corresponding CPU 'idle' time in Figure 6.10 (b). Focusing on Cold mode in Figure 6.10 (a) and (b), Routinator has a high system time of 51 seconds with more than 400 seconds of CPU idle time. We identify more than 455k page fault incidents for nearly 4 million write operations, with the inefficient handling of network activities leading to high system and idle time for Routinator in Cold mode. While around 15 seconds of CPU time is spent on the user process, the software interruption handling and the iowait time, respectively, are not a bottleneck and consume around 2 seconds to 5 seconds of CPU time. The rpki-client software has less than half the idle time of CPU than Routinator in Cold mode. The gap between system time and user time is small, falling in the range of 13 to 15 seconds, with the software interruption handling having a median of 4 seconds. The low CPU idle time by rpki-client indicates a more efficient use of network process handling and CPU resource utilization, ultimately leading to lower RP synchronization than Routinator.

The Warm mode in Figure 6.10 (a) depicts Routinator's median system time drastically reduced from around 50 seconds in Cold to nearly 13 seconds. The CPU idle time is less than 200 seconds shown in Figure 6.10 (b), with CPU cache False being slightly higher. In Warm mode, we have less network delay, as only the updates are fetched; hence, the

lower synchronization delay of Routinator further indicates that network handling is the main contributor to delay for Routinator. For rpki-client, the median CPU time for user-level code execution is higher, 10 seconds, than the system time of 7 when the CPU cache is True and is almost the same, 12 seconds when the CPU cache is False. For the idle time, rpki-client has a median value of around 120 seconds for the CPU cache True, and this median idle time jumps to around 195 seconds when the CPU cache is False. The CPU cache is causing around 75 seconds of extra idle time for the rpki-client in Warm mode, increasing the system time. Comparing the CPU idle time for rpki-client in Cold mode to rpki-client (F) in Warm and Hot modes, we find that the latter two modes with higher idle time, indicating rpki-client's poor utilization of the CPU when it has to read everything from the file system and no data is available in the CPU cache.

Figure 6.10 (a) depicts that all the median time values are less than 7 seconds for Routinator in Hot mode. The median values follow the same sequence of system, user, iowait, and soft times from high to low as other modes. Routinator has an idle time of 40 seconds, the lowest among all the cases. For rpki-client(T), the user time has the highest median value of around 10, followed by the system being 8 seconds, soft and iowait time having less than 5 seconds as the median. For rpki-client(F), the median user value jumps to around 13 seconds, with system time being almost the same as user time. The iowait and soft times are also slightly higher, with close to the same median of 4 seconds. For the rpki-client idle CPU time, we notice a similar median as the Warm mode, showing that as the CPU cache is False, the idle time median increases from around 120 to nearly 195 seconds.

6.3.2 RAM Utilization

We collect and analyze the statistics of Random Access Memory (RAM) utilization during RPKI synchronization. Figure 6.11 illustrates the maximum RAM utilization for both RPs in Cold, Warm, and Hot modes and for the CPU cache status of True and False. We see that Routinator requires around 520 MB RAM, the highest amount for the software in Cold mode, and almost a steady amount of nearly 380 Megabyte (MB) in Warm and Hot modes. During RP synchronization, the outlier in Warm mode also causes a higher RAM utilization as RP has to keep the state data and network session. For rpki-client, we observe a relatively high amount of nearly 580 MB RAM utilization, which is almost steady in Cold, Warm, and Hot modes of RPKI synchronization. It seems that rpki-client allocates a fixed amount of RAM, which might be pre-calculated, for its operation and performance despite being used in Cold, Warm, or Hot modes. We do not see any impact of CPU cache on maximum RAM utilization for RP software. Measuring the maximum RAM utilization for RPs using a memory-mapped file system further confirms that the maximum RAM utilization finding is stable and accurate. The amount of RAM allocated for the file system is not measured as the maximum RAM, as that is used more as storage for the RPKI cache.

6.3.3 Read and Write to File System

For each mode of Cold, Warm, and Hot, we collect the number of read and write operations performed on the file system. Figure 6.12 illustrates the number of write-to-file system operations for Routinator and rpki-client software with CPU cache status of True and False. For the Cold mode, we observe around 3.7 million write operations for Routinator and around 4 million for rpki-client. The high number of write operations is

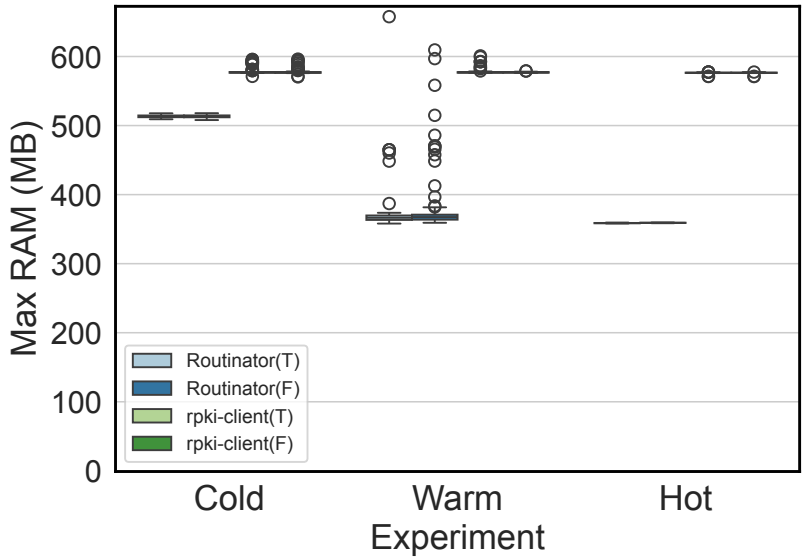

Figure 6.11: Box plot showing the median number for maximum RAM utilization during cold, warm, and hot RP synchronization modes. The box sequence aligns with the legend.

due to an empty cache, and the RPs write all the RPKI data fetched from the network to the file system, including the validated output produced after RP synchronization. While the Cold mode has the highest number of write operations, other modes do not observe even a million write operations, with most being lower than 100k write operations. The Warm mode, again, because of fetching partial updates from the network, has around a few thousand write operations, except when the CPU caches are unavailable, rpki-client's write operations reach nearly 600k sometimes. We discussed that the rpki-client is more sensitive. Hence, an empty CPU cache causes a high number of writes. The Hot mode has the lowest number of write operations, and nothing has to be fetched from the network. In contrast, the Hot and Warm modes require more than 4 million read operations when the CPU cache does not exist, as illustrated in Figure 6.13. When the data is unavailable in the CPU cache, RAM, swap, and buffer memory, the RP software needs to read it from the file system, causing many read operations in Warm and Hot modes. However, we observe little to no read operation for the Cold mode, even Warm mode with CPU cache available. This small number of reads is potentially due to the fact that when retrieving data from the network, a copy is also stored in RAM or swap memory, and RP does not need to perform read operations from the file system again.

6.3.4 Page Faults

When a computer program, in our study Relying Party software, tries to access a page of memory not currently available in RAM, a page fault occurs. The computer's operating system then loads the page from the file system to make it accessible for the RP software. The unavailability of a memory page in RAM and loading it from the file system delay RP synchronization performance. Figure 6.14 depicts the median number of page faults occurring in Cold, Warm, and Hot modes RPKI synchronization for Routinator and RPKI client. We observe only a high number of around 455k page faults for the Routinator in Cold mode. This high number of page faults is one of the key reasons for Routinator's

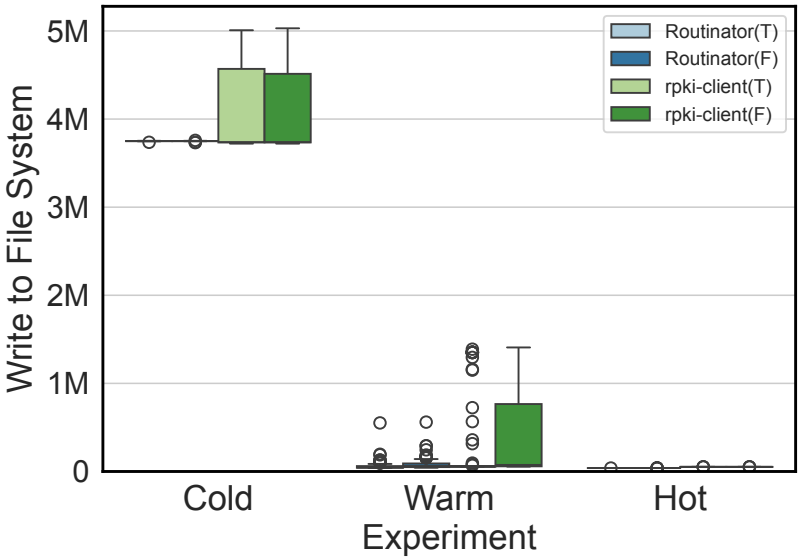

Figure 6.12: Box plot showing the median number of write operations to disk file system for cold, warm, and hot RP synchronization modes. The box sequence aligns with the legend.

high synchronization delay in Cold mode. Except for rpki-client(F), where we see around 10k page faults, mainly because of an empty CPU cache, we do not observe any number of page faults higher than 5k for other cases. Examining the number of page faults when storing RPKI cache on a memory-mapped file system, we identify a minimal number of less than 200-page faults for all the cases. This low number is due to the RPKI cache being available in memory. We can see that cleaning the RAM, dentries, and inodes for the cases of empty CPU cache causes page faults for both RPs, with Routinator having 170-page faults as the highest in Cold mode and more than 100-page faults in Cache mode.

6.4 Discussion

The reason for the RPKI synchronization delay varies based on the selection of RP software. Routinator has the faster cryptographic verification of RPKI objects, and rpki-client has efficient network handling in RPKI synchronization. This difference between the implementation of RP software indicates a potential room for optimization for Routinator to handle the network activities efficiently and for the rpki-client to perform the cryptographic verification process faster. As RPKI synchronization requires a high number of read and write operations, using a memory-mapped file system can further optimize the delay.

RIPE and APNIC consolidate multiple prefixes into a single ROA object, which is cheaper regarding resource consumption (CPU load, disk space) and more beneficial during the RP synchronization process than one prefix per ROA. Conversely, ARIN, AFRINIC, and LACNIC have a simpler management of ROA objects of more or less one prefix per ROA. However, using a single ROA for multiple prefixes complicates ROA management, for example the expiration or change for a single prefix requires to reissue the ROA for all prefixes.

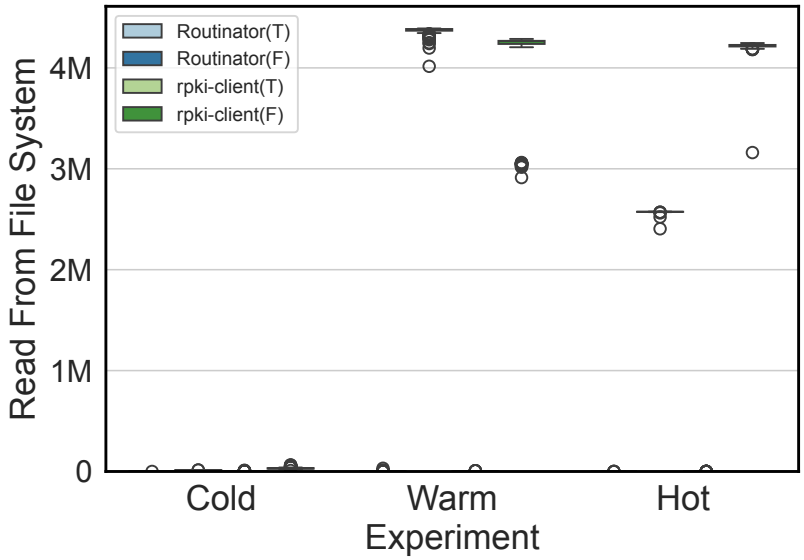

Figure 6.13: Box plot showing the median number of read from file system operation for cold, warm, and hot RP synchronization modes. The box sequence aligns with the legend.

In terms of management, using ROA with a single IP prefix is simpler and thus recommended [165, 293]. One of the RIRs acknowledged aggregating multiple prefixes for the same ASN on a single ROA, which needs a complex system that has to carefully track data changes and reissue ROA objects proactively without risking IP prefix invalidation. Adding a new prefix to RPKI means that the CA should maintain a state for the existing ROAs, revoke the ROA that should be updated, and create a new one with the new prefix, a time-consuming task for CAs but beneficial for RP synchronization.

The historical trend of registering prefixes to the RPKI [195], reported growth of IPv6 prefixes in global routing table [211], usage of small CIDR sizes or hyperspecific prefixes [254], with some prefixes used by multiple origin ASes [252] results in a higher number of VRPs coming in the RPKI ecosystem. Delegated RPKI permits a distributed and flexible architecture but may introduce complexity and potential delays in RPKI synchronization. Despite the current and relatively low numbers of delegated repositories, our findings reveal that RPs hang on specific delegated Publication Points for hundreds of seconds, making the entire RPKI synchronization process slow. The two studied RPs implement timeout mechanisms to skip unresponsive Publication Points, yet as the RPKI infrastructure and the number of RPKI objects (e.g., ROAs) is growing, we expect a larger number of delegated CAs; hence, quick synchronization will become even more challenging.

The current Publication Points deployment of RPKI data lacks a worldwide quick-to-access setup, with most of the PPs being accessible fast only in their primary serving regions. Serving the RPKI data from a worldwide, easy-to-access infrastructure like a CDN will help reduce the network delay in the RPKI synchronization process. We observe that RRDP PPs with cachable data on CDN are quicker to access globally than the PP used for rsync, which is not cachable on CDN. Ideally, serving all the RPKI data using RRDP might be faster. However, in practice, we encounter cases where a Publication Point can not serve data using RRDP, and they switch to the rsync protocol, which has known vulnerabilities [66] and slows the RP synchronization.

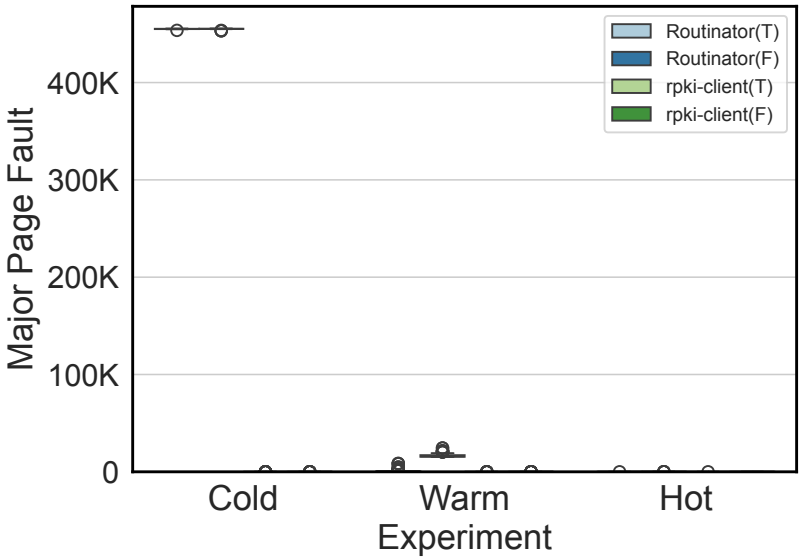

Figure 6.14: Box plot showing the median number of major page fault for cold, warm, and hot RP synchronization modes. The box sequence aligns with the legend.

The rapid changes in RPKI software and RP implementations make RPKI a challenging and moving target for this study. We found two Publication Points with no IPv6 addresses, resulting in a disparate number of ROAs when validating from an IPv6 RP; however, the problem was resolved within a few months. Introduction of new object types like AS Provider Authorization, Autonomous System Provider Authorization (ASPA) [18], Signed Prefix List [265], improved validation procedure [263], are examples of the rapid changes in the RPKI landscape. Thus, we believe RPKI synchronization delay may change over time and needs constant attention from the networking community.

We acknowledge that conducting RP synchronization from diverse locations with additional RP software would provide us with broader coverage of how networks experience RP synchronization delay. Despite limitations, our approach is suitable, and our analysis identifies the delay factors in the RP synchronization process and provides valuable insights on how to improve them.

6.5 Related Work

Osterweil et al. [205] simulated a "fully deployed" RPKI and showed that it would take between 15 to 30 days to synchronize the local caches at RPs around the world. The number of objects was calculated based on RIRs' allocation trends with a one-to-one mapping between allocation and a ROA. However, since 2012, there have been a few improvements to RPKI, namely the standardization of the RPKI Delta Protocol [32], the use of Content Delivery Networks for repository hosting [259], bundling ROAs and other network optimizations that helped reduce the fetching time.

Measuring RPs in the wild, Kristoff et al. [158] looked at the *timeliness* of RPs which is based on their configured refresh time intervals that would determine how many times an RP would synchronize with the PP. Routinator, a RP software, has a refresh interval of

10min [164]. For rpki-client, another RP software, the manual page states an hourly run using crontab [216]. To date, there are no defined standards on the refresh time interval and each RP comes with its own default time settings.

Hlavacek et al. [120] analyzed the retrieval interval, the cache management algorithms and the frequency at which the different RPs refresh the cache by contacting the publication server. They also looked at the reachability of PPs by varying network configurations such as enabling/disabling Domain Name System Security Extensions (DNSSEC). Hlavacek et al. [121] proposed an attack scenario whereby an attacker can inject packet loss at specific time intervals and stall RPs by creating very long delegation chains. This would prevent RPs from fetching ROAs and other RPKI data from PPs, forcing the expiry of cached objects. While their attack scenario is hypothetical, increasing the depth of certificates impacts the synchronization time, as we see in §6.1.4. The possibility for malicious repositories that could stall the RP synchronization process was studied by Hlavacek et al. [122], and the sort-and-limit algorithm is proposed to prevent attacks and repository failures. van Hove et al. [282] studied a threat model where the attacker controls a CA and PP, and discovered that attackers might have the potential to disrupt RP software.

RPKI MIRO [226] was introduced to monitor and inspect RPKI objects. Researchers investigated RPKI usage by assessing the deployment of RPKI and ROV [60, 169, 225, 237, 277] and explored the RP software vulnerabilities, security concerns, and RFC compliance [189, 190, 249].

To reduce RP synchronization time Li et al. [170] proposed a bit-map encoding scheme to compress the total size of ROA payloads in RRDP by 26.6%. Using a 10 Mbps link, they showed a reduction of 41.3% for the currently used max-length approach and 50.4% for the minimal ROA approach [99]. While the compression algorithm proposed can reduce validation time, it remains theoretical as it requires adoption by all RPs, which is impractical.

The closest study to ours is by Fontugne et al. [90] which looked at the end-to-end delay between the creation or removal of a ROA and the corresponding changes observed in BGP. They announced prefixes assigned from all five RIRs and created/deleted ROAs to change the RPKI status of the announcements. Delay metrics were recorded at several steps of the RPKI lifecycle, namely at user creation, ROA signing, publication, and BGP update. They found that RIRs usually publish new RPKI information within five minutes, except APNIC, which is ten minutes slower.

However, their work does not investigate the detailed causes for RP's synchronization delay. We address this gap by providing an in-depth analysis of the delay factors in the RP synchronization process.

6.6 Summary

Using two RP software, we thoroughly analyze the elements that induce RP synchronization delay, including the structure of ROAs, the impact of delegation in RPKI resource hosting, and the latency to Publication Points. Network delay and the cryptographic verification delay of RPKI data are the two main elements of delay in RPKI synchronization. Routinator is fast in cryptographic verification and robust in handling empty CPU cache, but suffers from many page faults during data retrieval from the Internet. Therefore, it has the highest delay of nearly 500 seconds in Cold mode and the lowest delay of 40 seconds in Hot mode. rpki-client efficiently uses the network with almost no page faults,

but it takes double the time of Routinator for cryptographic verification of ROAs and is more prone to the system's CPU cache state. Consequently, it has around 190 seconds, less than half of the Routinator delay in Cold mode, and more than double of Routinator in Hot mode, and a similar delay in Warm mode. In normal operations, RPs typically run in Warm mode, so rpki-client and Routinator will have a very similar synchronization delay in day-to-day performance.

Bundling several prefixes of an AS into a single ROA can reduce the synchronization time by a factor of three, and is possible for more than 60% of ASes in RPKI. Delegated CAs hosted on a fast-to-access infrastructure positively impact the RPKI validation rate, but for ARIN, 7% of VRPs from delegated CAs cause more than 90% delay.

We highlight regional disparities in accessing RPKI PPs, with Europe and North America experiencing lower latencies and other regions significantly higher latencies in accessing RPKI resources. The current PP deployment delays reveal that each RIR's RPKI tree is accessible relatively faster in their primary serving regions and slower from other regions. Our results show that CDN usage can reduce the PP accessing time by up to 10 times in some regions.

We quantify that every additional 100 milliseconds of extra delay in the network path will increase the RP synchronization by 25 seconds for Routinator and 10 seconds for rpki-client in Cold cache mode, and almost half of the values in Warm mode.

Chapter 7
Conclusion

In this dissertation, we examined the Internet routing ecosystem to detect and classify unexpected routes. Using active and passive Internet measurement techniques along with various supplementary datasets, we investigated the causes of unexpected routes, their users, and potential function. Our findings reveal that routes such as hyper-specific prefixes and MOAS prefixes exist due to accidental leakage, mergers and acquisitions of companies, or network operators use them for specific routing services such as fine-grained traffic engineering. Our findings of unexpected routes sharpens the interpretation of HSPs, clarifies long-lived MOAS cases helping network operators and researchers separate normal prefixes from routing inconsistencies. Applying the Jaccard similarity index on DNS records, we identify IPv4 and IPv6 prefix pairs whose IPs contain identical or closely related DNS records, indicating semantic alignment across IP versions. Moreover, analyzing the RPKI synchronization delay, we find that bundling several prefixes of an AS, though not recommended, improves cryptographic verification of RPKI data by a factor of three.

7.1 Summary

In this dissertation, our goal is to understand the role of unexpected routes and analyze their prevalence in Internet routing. Assessing the operational, security, and potential functional implications of unexpected routes in Internet routing is another objective in this thesis. Our overarching research goal is to develop and apply methodologies for characterizing the unexpected routes in the Internet's complex and dynamic routing ecosystem underlined by the following research question:

Q: What is the prevalence of unexpected routes in Internet routing?

To answer this question, we examine the IP prefix size, number of origin ASes, the similarity between IPv4 and IPv6 prefix pairs, and the RPKI ecosystem, each separately

and in detail with its sub-research question. We began our analysis by examining routes with a prefix size that was too specific, as stated in the first sub-research question.

Q1: How prevalent are Hyper-specific prefixes in Internet routing?

In Chapter 3, we thoroughly analyze the prevalence of hyper-specific prefixes on the Internet. In particular, we utilize eleven years' worth of BGP-based routing information collected by three route collector projects, real-world experimentation, and various supplemental data sets to provide insight into the prevalence, propagation, persistence, and intended use cases, and whether operators use HSPs intentionally or accidentally.

Our observations show that HSPs are in use on the Internet, especially between connected neighbors; however, due to their limited propagation across AS hops on the Internet, it is hard to quantify the exact number of available HSPs. We find that HSP usage increased over time. At their peak, HSPs formed 18 % and 23 % of all the IP prefixes announced on the Internet, for IPv4 and IPv6, respectively, with an average of 10% of announcements being HSPs in 2020. Up to 3,000 ASes announce HSPs at various CIDR sizes. Even though their announced HSPs represent approximately 10 % of all visible prefixes, they only propagate to five route collector peers, on average.

The propagation of HSP via the limited number of networks and blockage by many other network operators indicates no common mechanism among operators for handling HSPs. While we do not observe the majority of HSPs to be associated with blackholing or more specific prefix hijacks, we find that they are filtered rigorously among settlement-free peers and only propagate between customers and providers.

While their total number increases over time, most HSPs can only be seen for a few minutes by route collector peers. Nonetheless, some HSPs can be seen constantly throughout an entire year and propagate widely. Most HSPs represent (internal) routes to peering infrastructure or are related to address block relocations or blackholing. While hundreds of operators intentionally add HSPs to well-known routing databases, we observe that many HSPs are possibly accidentally leaked routes.

By conducting a BGP experiment, we show that HSPs can traverse 2 % of Internet routes. Moreover, we use historical active measurement data to identify services overrepresented in HSPs compared to the general Internet. Most of those protocols are used for infrastructure services, while others, such as CPE management protocols, are heavily underrepresented in HSPs. We also identify HSPs which have been atomically aggregated into /24 and /48 prefixes mostly at their origin AS, giving a more complete picture of the HSP landscape.

After hyper-specific prefix analysis, we focus on the origin AS part of the IP prefixes with the following sub-research question.

Q2: What is the prevalence of long-lived MOAS prefixes in Internet routing?

As explained in Chapter 4, we analyzed long-lived MOAS prefixes over a period of six years. We identify around 24k long-lived MOAS prefixes in IPv4 and 1.4k in IPv6 being announced in January 2023. By analyzing the RPKI status, we find that more than 40% of MOAS prefixes have all origins registered correctly, with only a minority of MOAS having invalid origins. Moreover, the most prominent CIDR size of MOAS prefixes is /24 for IPv4 and /48 for IPv6, suggesting their use for fine-grained traffic steering.

We attribute a considerable number of MOAS prefixes to mergers and acquisitions of companies. Additionally, more than 90% of MOAS prefixes originate from two origin ASes, with the majority of detected origin AS relations being customer-provider. We identify that the majority of MOAS users are IT companies, and just 0.9% of IPv4 MOAS and 6.3% of IPv6 MOAS prefixes are used for anycast.

We introduced the concept of sibling prefixes, extending the notion of IPv4–IPv6 sibling relationships to pairs of IP prefixes—specifically, one IPv4 prefix and its corresponding IPv6 prefix. Our research question was:

Q3: How prevalent are sibling prefixes in the Internet routing?

We provide a detailed analysis of sibling prefixes using large-scale DNS resolution data and various supplemental datasets in Chapter 5. We identified over 76,000 such sibling prefix pairs and found them to be relatively stable over time. To improve the precision of these sibling relationships, we proposed the SP-Tuner algorithm, which fine-tunes the CIDR sizes of prefixes to increase their similarity. This tuning significantly improved the share of perfect match sibling prefixes from 52% to 82%, as measured by the Jaccard similarity index applied to dual-stack domains. Our analysis revealed that in over half of the sibling pairs, the IPv4 and IPv6 origin ASes are operated by the same organization—most often large IT companies. Furthermore, 60% of sibling prefixes had at least one prefix with a valid Route Origin Validation status in RPKI. However, we also observed a few cases with inconsistent ROV states between IPv4 and IPv6 counterparts. Sibling prefixes were notably common in major hypergiants and content delivery networks, highlighting their operational significance in today's dual-stack Internet.

Finally, we study the RPKI synchronization delay reasons and the RFC-recommended ROA structure adoption under the fourth sub-research question as follows:

Q4: How recommended ROA structure impacts RPKI synchronization delay?

Our findings, as explained in Chapter 6, show that using a single prefix per ROA object, as the RFC recommends, can take up to three times more time in the cryptographic verification process of RPKI data than bundling multiple prefixes into a single ROA. RPKI synchronization efficiency heavily depends on the delay of the Publication Point infrastructure. Delegated Certificate Authorities can significantly reduce or inflate delays by as much as 90%, depending on how and where their data is hosted. Using two popular Relying Party software implementations—Routinator and rpki-client—we dissected the interplay between network delays, cryptographic validation, CPU caching behavior, and memory access efficiency. Routinator excels in cryptographic performance and remains stable under cold CPU cache conditions, but suffers from frequent page faults during data retrieval, resulting in the highest observed synchronization delay of around 500 seconds in Cold mode. In contrast, rpki-client demonstrates superior network efficiency and minimal page faults but is more sensitive to CPU cache state and slower in cryptographic processing, leading to varied delays across modes—approximately 190 seconds in Cold mode and over double the delay of Routinator in Hot mode. Both tools show comparable performance in typical day-to-day operation (Warm mode). We further highlight global disparities in accessing RPKI PPs: while Europe and North America benefit from low latency, other regions face significantly higher delays, sometimes in the hundreds of milliseconds. Each additional 100 milliseconds of RTT can increase synchronization

delays by up to 25 seconds in Routinator and 10 seconds in rpki-client under Cold mode. Leveraging CDN infrastructure for hosting PPs could dramatically reduce these access times by as much as a factor of 10 in certain regions.

7.2 Future Work

In this section, we explain the potential future direction of our study. As shown throughout this dissertation, the number of unexpected routes is increasing, complicating the Internet routing ecosystem. Therefore, regular monitoring of interdomain routing and adherence of network operators to recommended best current practices are essential for the proper operation of internet routing. Recent proposed IETF draft [97] on publishing end-site prefix lengths complements our study by providing a standardized mechanism to describe and authenticate the expected sub-prefix structure of address allocations. Related to our study, we propose future directions to deepen our understanding of unexpected routes further.

Filtering and propagation policies for HSPs Based on the longest prefix match behavior of the router, the HSPs steer the traffic from any other least specific or covering prefixes; hence, having ROA records for HSPs is important to distinguish between valid HSPs and possible traffic hijack by invalid ROA HSPs. Based on our results, we argue that the scientific and operational communities should rethink their handling of hyper-specific prefixes in the future. Considering the shortage of IPv4 and possible usage of a wide range of available IPv6 address space, network operators may use HSPs to a large extent in the near future, which will lead to a large routing table size. Hence, a common consensus on the borderline between which CIDR size to accept and further propagate the prefix is important for the future of routing on the Internet. The ingress and egress filters are becoming more and more important. Since HSPs cause additional BGP updates, link, and resource utilization to recalculate the BGP routes and increase the routing table size, it is important for the network operators to clearly define when and from which network to accept HSPs and under which conditions to transit HSPs to other ASes.

Accidental or intended MOAS prefixes

Adopting MOAS prefixes can benefit large companies, including improved network reliability, performance, and control, mainly when the network is geographically dispersed. For example, MOAS prefixes can improve network resilience by providing the ability to route traffic through multiple paths if one origin AS becomes unreachable. Additionally, MOAS prefixes can enhance network performance by creating multiple traffic paths and allowing external organizations to control traffic routing. Throughout our analysis, we have identified industries such as Google, Amazon, Facebook/Meta, Alibaba, Netflix, Cloudflare, and Limelight Networks as users of MOAS prefixes in their network infrastructure. However, distinguishing between intended MOAS prefixes and MOAS due to the merger of companies is a challenging task. Exploring supplementary datasets such as RPKI and IRR may help infer the actual usage from accidental leakage of MOAS prefixes to some extent. The exact classification of MOAS prefixes to one of the aforementioned categories and the exact business correlation between the origin ASes of MOAS prefixes are potential directions to be explored as future work.

Choosing the right prefix size for sibling prefixes Our study of sibling prefixes lays the groundwork for further analysis of similarity between the two IP versions. With SP-Tuner we present an algorithm to fine-tune the CIDR size of sibling prefixes to achieve even higher Jaccard similarity values. Although we mostly use the /28 and /96 pairs as

the SP-Tuner thresholds in this study, options in selecting any pair of thresholds make SP-Tuner flexible, and allow to choose appropriate prefix size thresholds based on the need to further fine-tune the resulting sibling prefixes. While this is a good first step in finding well suited prefix sizes, it is not a silver bullet. IPv4 address exhaustion [214, 217] leads to smaller and smaller prefixes being allocated by RIRs, which in turn then leads to more and more fragmentation of the IPv4 address space. Furthermore, identifying the right prefix size remains a challenge, especially in IPv6 [96]. These factors make it exceedingly challenging to identify sibling prefixes. Therefore, it might be useful to look into sibling prefix *set* pairs, i.e., a set of IPv4 prefixes which are siblings of a set of IPv6 prefixes. This could alleviate challenges such as address space fragmentation by pairing different IPv4 fragments with their IPv6 counterpart. We leave the analysis of sibling prefix set pairs to future work.

Optimizing RPKI synchronization Our results highlight the potential for improvement in RPKI synchronization through more efficient ROA structuring, better geographic distribution of Publication Points, and further software optimizations. The ROA structure is not the same across the five RIRs' RPKI tree, with some using a single prefix per ROA object while others use multiple prefixes of an AS in a single ROA. Measuring the impact of ROA structure on daily RPKI operation, ROA changes, and network delay could help understand the benefits of each ROA structure. While using the cloud for repository hosting generally provides users with fast access to RPKI resources, the drawback of this approach remains an underexplored topic. Profiling the RP software to understand which functions or processes take a long time and optimizing them can improve the RPKI synchronization process. Routinator is fast in cryptographic verification of RPKI, and rpki-client has efficient network handling in RPKI synchronization. Therefore, the developers of both software have the opportunity to improve their software's RPKI synchronization further.

List of Abbreviations

AIA Authority Information Access 101

APNIC Asia Pacific Network Information Centre 7

AS Autonmous System 1, 11

ASdb Autonomous System Database 15

ASN Autonmous System Number 1

ASNs Autonmous System Numbers 11

AWS Amazon Web Services 14

BCP Best Current Practice 2

BGP Border Gateway Protocol 1

BH blackholing 29

CA Certificate Authority 92

CAIDA Center for Applied Internet Data Analysis 15

CAs Certificate Authorities 7

CDN Content Delivery Network 6

CDNs Content Delivery Networks 1

CIDR Classless Interdomain Routing 2

CPE Customer Premises Equipment 29

CPU Central Processing Unit 105

CRLs Certificate Revocation Lists 17

CWMP CPE WAN Management Protocol 28

DNS Domain Name System 2

DPS DDoS Protection Service 32

DS dual-stack 63

eBGP external Border Gateway Protocol 12

FIB Forwarding Information Base 5, 12

GCP Google Cloud Platform 14

HG Hypergiant 64

IaaS Infrastructure as a Service 14

IANA Internet Assigned Numbers Authority 10

iBGP internal Border Gateway Protocol 12

IETF Internet Engineering Task Force 1

IoT Internet of Things 1

IP Internet Protocol 1

IQR interquartile range 21

IRR Internet Routing Registry 15

ISPs Internet Service Providers 1

IXPs Internet Exchange Points 1

LIRs Local Internet Registries 10

MANRS Mutually Agreed Norms for Routing Security 7

MED Multi-Exit Discriminator 12

MOAS Multiple Origin AS 2

MSPs More-Specific Prefixes 24

NAT Network Address Translation 11

NIRs National Internet Registries 10

ORCHID Overlay Routable Cryptographic Hash IDentifiers 11

OSI Open Systems Interconnection 10

PaaS Publication as a Service 16

PaaS Platform as a Service 14

PO prefix-origin 50

PP Publication Point 92

RAM Random Access Memory 107

RCs Route Collectors 15

RDAP Registration Data Access Protocol 16

RES Restrict Route Propagation 29

RFC Request for Comments 11

RIB Routing Information Base v

RIPE Réseaux IP Européens 1

RIPE NCC Réseaux IP Européens Network Coordination Centre 18

RIRs Regional Internet Registries 3

RIS Routing Information System 1

ROA Route Origin Authorization v

ROV Route Origin Validation 2

RP Relying Party 3

RPKI Resource Public Key Infrastructure v

RRDP RPKI Resource Delta Protocol 93

RTT Round Trip Time 7

SaaS Software as a Service 14

SEE South East Europe 7

SMTP Simple Mail Transfer Protocol 28

TA Trust Anchor 92

TALs Trust Anchor Locators 18

TCP Transmission Control Protocol 12

TMA Network Traffic Measurement and Analysis 7

TTL Time To Live 101

URI Uniform Resource Identifier 18

VM Virtual Machine 94

VPSes virtual private servers 72

VRPs Validated ROA Payloads 92

List of Figures

List of Tables

Bibliography

[1] Routing security summit 2023. https://manrs.org/event/routing-security-summit-2023/, 2023. Virtual event held 17–20 July 2023; Accessed on September 06, 2025.

[2] Awards – tma conference 2025. https://tma.ifip.org/2025/awards/, 2025. Accessed on September 06, 2025.

[3] Conference and phd school awards – tma conference 2023. https://tma.ifip.org/2023/awards/, 2025. Accessed on September 06, 2025.

[4] Leila Abboud and Julien Toyer. Orange strikes 3.4 billion euro deal to buy Spain's Jazztel. https://www.reuters.com/article/us-orange-jazztel-idUSKBN0HA1SM20140915, Sep 2014.

[5] Emile Aben and Colin Petrie. Propagation of Longer-than-/24 IPv4 Prefixes, 2014.

[6] Emile Aben and Colin Petrie. Has the Routability of Longer-than-/24 Prefixes Changed?, 2015.

[7] Bernhard Ager, Nikos Chatzis, Anja Feldmann, Nadi Sarrar, Steve Uhlig, and Walter Willinger. Anatomy of a large european ixp. In *Proceedings of the ACM SIGCOMM Conference*, pages 163–174, 2012.

[8] Taha Albakour, Oliver Gasser, Robert Beverly, and Georgios Smaragdakis. Third time's not a charm: Exploiting SNMPv3 for router fingerprinting. In *Proceedings of the 21st ACM Internet Measurement Conference*, pages 150–164, 2021.

[9] Taha Albakour, Oliver Gasser, and Georgios Smaragdakis. Pushing Alias Resolution to the Limit. In *Proceedings of the 2023 ACM on Internet Measurement Conference*, pages 584–590, 2023.

[10] Amazon.com. Amazon Alexa. https://www.alexa.com/topsites, 2023. Accessed on April 30, 2023.

[11] American Registry for Internet Numbers (ARIN). Requesting autonomous system numbers (asns). https://www.arin.net/resources/guide/request/#autonomous-system-numbers-asns, 2024. Accessed on May 19, 2025.

[12] IoT Analytics. Iot 2019 in review, December 2019. Accessed on February 11, 2025.

[13] APNIC Labs. IPv6 Capability Measurements. https://stats.labs.apnic.net/ipv6. Accessed on April 28, 2025.

[14] ARIN. Internet routing registry (irr). https://www.arin.net/resources/manage/irr/, n.d. Accessed on July 02, 2025.

[15] J. Arkko, M. Cotton, and L. Vegoda. IPv4 Address Blocks Reserved for Documentation. RFC 5737, IETF, Jan 2010.

[16] Todd Arnold, Jia He, Weifan Jiang, Matt Calder, Italo Cunha, Vasileios Giotsas, and Ethan Katz-Bassett. Cloud provider connectivity in the flat internet. In *Proceedings of the ACM Internet Measurement Conference*, pages 230–246, 2020.

[17] Augusto Arturi, Esteban Carisimo, and Fabián E Bustamante. as2org+: Enriching AS-to-Organization Mappings with PeeringDB. In *Passive and Active Measurement: 24th International Conference, PAM 2023, Virtual Event, March 21–23, 2023, Proceedings*, pages 400–428. Springer, 2023.

[18] Alexander Azimov, Eugene Bogomazov, Randy Bush, Keyur Patel, Job Snijders, and Kotikalapudi Sriram. BGP AS_PATH Verification Based on Autonomous System Provider Authorization (ASPA) Objects. Internet-Draft draft-ietf-sidrops-aspa-verification-19, Internet Engineering Task Force, September 2024. Work in Progress.

[19] Shehar Bano, Philipp Richter, Mobin Javed, Srikanth Sundaresan, Zakir Durumeric, Steven J. Murdoch, Richard Mortier, and Vern Paxson. Scanning the internet for liveness. *SIGCOMM Comput. Commun. Rev.*, 48(2):2—-9, 2018.

[20] R. Barnes, M. Lepinski, A. Cooper, J. Morris, H. Tschofenig, and H. Schulzrinne. An Architecture for Location and Location Privacy in Internet Applications. RFC 6280, IETF, Jul 2011.

[21] Arthur Berger, Nicholas Weaver, Robert Beverly, and Larry Campbell. Internet nameserver IPv4 and IPv6 address relationships. In *Proceedings of the 2013 conference on Internet measurement conference*, pages 91–104, 2013.

[22] Robert Beverly and Arthur Berger. Server siblings: Identifying shared IPv4/IPv6 infrastructure via active fingerprinting. In *International Conference on Passive and Active Network Measurement*, pages 149–161. Springer, 2015.

[23] Robert Beverly, William Brinkmeyer, Matthew Luckie, and Justin P Rohrer. IPv6 alias resolution via induced fragmentation. In *Passive and Active Measurement: 14th International Conference, PAM 2013, Hong Kong, China, March 18-19, 2013. Proceedings 14*, pages 155–165. Springer, 2013.

[24] BGPmon. BGPmon Hijacking events. `https://www.bgpmon.net/category/hijack/`, 2019.

[25] BGP.tools. Anycast Prefixes list on GitHub. `https://github.com/bgptools/anycast-prefixes`, 2022. anycast detector.

[26] Ernst Biersack, Quentin Jacquemart, Fabian Fischer, Johannes Fuchs, Olivier Thonnard, Georgios Theodoridis, Dimitrios Tzovaras, and Pierre-Antoine Vervier. Visual analytics for BGP monitoring and prefix hijacking identification. *IEEE Network*, 26(6):33–39, 2012.

[27] M. Blanchet. Special-Use IPv6 Addresses. RFC 5156, IETF, Apr 2008.

[28] Uli Bornhauser, Jan Luehr, Michael Schaefer, Thomas Trimborn, and Zia Ul Huda. Automatic Analysis and Classification of Multiple Origin AS (MOAS) Conflicts. In *36th IEEE Conference on Local Computer Networks (LCN)*, 2011.

[29] Timm Böttger, Gianni Antichi, Eder L Fernandes, Roberto di Lallo, Marc Bruyere, Steve Uhlig, and Ignacio Castro. The elusive internet flattening: 10 years of ixp growth. *CoRR*, 2018.

[30] Timm Böttger, Felix Cuadrado, and Steve Uhlig. Looking for Hypergiants in PeeringDB. *ACM SIGCOMM Computer Communication Review*, 48(3):13–19, 2018.

[31] R. Braden. Requirements for Internet Hosts - Communication Layers. RFC 1122, IETF, Oct 1989.

[32] T Bruijnzeels, O Muravskiy, B Weber, and R Austein. RFC 8182: The RPKI Repository Delta Protocol (RRDP), 2017.

[33] T. Bruijnzeels, O. Muravskiy, B. Weber, and R. Austein. The RPKI Repository Delta Protocol (RRDP). RFC 8182, IETF, Jul 2017.

[34] Tian Bu, Lixin Gao, and Don Towsley. On characterizing bgp routing table growth. In *Global Telecommunications Conference, 2002. GLOBECOM'02. IEEE*, volume 3, pages 2185–2189. IEEE, 2002.

[35] R. Bush and R. Austein. The Resource Public Key Infrastructure (RPKI) to Router Protocol. RFC 6810, IETF, Jan 2013.

[36] R. Bush and R. Austein. The Resource Public Key Infrastructure (RPKI) to Router Protocol, Version 1. RFC 8210, IETF, Sep 2017.

[37] CAIDA. AS Relationships Dataset. `https://www.caida.org/catalog/datasets/as-relationships/`. Covers data from January 2017 to July 2022.

[38] CAIDA. BGP Hijacks Observatory. `https://dev.hicube.caida.org/feeds/hijacks/events`.

[39] CAIDA. Inferred AS to Organization Mapping Dataset. `https://www.caida.org/catalog/datasets/as-organizations/`. Covers data from January 2017 to July 2022.

[40] CAIDA. Mapping Autonomous Systems to Organizations: CAIDA's Inference Methodology. `https://www.caida.org/archive/as2org/`. Covers data from January 2017 to July 2022.

[41] CAIDA. The caida as to organization mapping dataset. `https://catalog.caida.org/dataset/as_organizations/`, 2012. Accessed on October 15, 2024.

[42] CAIDA. As classification dataset - [2019/01/01-2021/04/01], 2021.

[43] CAIDA. As relationships dataset - [2010/01/01-2021/04/01], 2021.

[44] CAIDA. The ipv4 routed /24 topology dataset - [2021/05/17-23], 2021.

[45] CAIDA. The ipv6 routed /48 topology dataset - [2021/05/17-23], 2021.

[46] CAIDA. As rank and bgp data analysis. `https://asrank.caida.org`, 2024. Accessed on April 24, 2025.

[47] Matt Calder, Ashley Flavel, Ethan Katz-Bassett, Ratul Mahajan, Jitendra Padhye, and Vyas Rao. Mapping the expansion of google's serving infrastructure. In *Proceedings of the ACM Internet Measurement Conference (IMC)*, pages 313–326, New York, NY, USA, 2015. Association for Computing Machinery.

[48] Massimo Candela. NTT's RPKI Deployment Update. RIPE 82, 2021.

[49] Massimo Candela. RPKI Automation and Monitoring. LACNIC 35, 2021.

[50] Massimo Candela. A one-year review of RPKI operations. RIPE 84, 2022.

[51] B. Carpenter and K. Moore. Connection of IPv6 Domains via IPv4 Clouds. RFC 3056, IETF, Feb 2001.

[52] Catchpoint Systems, Inc. Catchpoint - internet performance monitoring, 2025. Accessed on May 2, 2025.

[53] CDN Planet. Content delivery network. `https://www.cdnplanet.com/cdns/`, 2024.

[54] R. Chandra, P. Traina, and T. Li. BGP Communities Attribute. RFC 1997, IETF, Aug 1996.

[55] Rocky KC Chang and Michael Lo. Inbound traffic engineering for multihomed ass using as path prepending. *IEEE network*, 19(2):18–25, 2005.

[56] Zhiyi Chen, Zachary S Bischof, Cecilia Testart, and Alberto Dainotti. Improving the Inference of Sibling Autonomous Systems. In *Passive and Active Measurement: 24th International Conference, PAM 2023, Virtual Event, March 21–23, 2023, Proceedings*, pages 345–372. Springer, 2023.

[57] Kwan-Wu Chin. On the characteristics of BGP multiple origin AS conflicts. In *2007 Australasian Telecommunication Networks and Applications Conference*, pages 157–162. IEEE, 2007.

[58] Shinyoung Cho, Romain Fontugne, Kenjiro Cho, Alberto Dainotti, and Phillipa Gill. BGP hijacking classification. In *2019 Network Traffic Measurement and Analysis Conference (TMA)*, pages 25–32. IEEE, 2019.

[59] David R. Choffnes and Fabian E. Bustamante. Crowdsourcing isp characterization with netalyzr. In *Proceedings of the ACM Internet Measurement Conference (IMC)*, pages 43–56, New York, NY, USA, 2010. Association for Computing Machinery.

[60] Taejoong Chung, Emile Aben, Tim Bruijnzeels, Balakrishnan Chandrasekaran, David Choffnes, Dave Levin, Bruce M. Maggs, Alan Mislove, Roland van Rijswijk-Deij, John Rula, et al. Rpki is coming of age: A longitudinal study of rpki deployment and invalid route origins. In *Proceedings of the Internet Measurement Conference*, pages 406–419, New York, NY, USA, 2019. Association for Computing Machinery (ACM).

[61] Cisco. Removing private as numbers from the as path in bgp, 2021.

[62] Cisco. Umbrella Top 1M Lists. `https://umbrella.cisco.com/`, 2024. Accessed on April 10, 2024.

[63] Luca Cittadini, Wolfgang Mühlbauer, Steve Uhlig, Randy Bush, Pierre Francois, and Olaf Maennel. Evolution of internet address space deaggregation: Myths and reality. *IEEE Journal on Selected Areas in Communications*, 28(8):1238–1249, 2010.

[64] Cloudflare. 10,000 networks and beyond. `https://blog.cloudflare.com/10000-networks-and-beyond/`, 2021. Accessed on April 28, 2025.

[65] Cloudflare. Cloudflare Radar Top List. `https://radar.cloudflare.com/`, 2024. Accessed on April 10, 2024.

[66] Computer Incident Response Center Luxembourg (CIRCL). Vulnerability bundle: d938dc28-6877-40db-ad5f-25f3051288e6, 2025. Accessed on January 16, 2025.

[67] Wikipedia contributors. As 7007 incident, 2025. Accessed on February 11, 2025, Wikipedia, The Free Encyclopedia.

[68] Wikipedia contributors. Internet service provider, 2025. Wikipedia, The Free Encyclopedia.

[69] M. Cotton, L. Vegoda, and D. Meyer. IANA Guidelines for IPv4 Multicast Address Assignments. RFC 5771, IETF, Mar 2010.

[70] Jakub Czyz, Mark Allman, Jing Zhang, Scott Iekel-Johnson, Eric Osterweil, and Michael Bailey. Measuring IPv6 Adoption. In *Proceedings of the 2014 ACM conference on SIGCOMM*, pages 87–98, 2014.

[71] Jakub Czyz, Matthew J. Luckie, Mark Allman, and Michael Bailey. Don't forget to lock the back door! a characterization of ipv6 network security policy. In *23rd Annual Network and Distributed System Security Symposium, NDSS*, 2016.

[72] Daily Sabah. Türk Telekom swallows up Avea and TTNET. `https://www.dailysabah.com/business/2016/01/27/turk-telekom-swallows-up-avea-and-ttnet`. Accessed on March, 2023.

[73] DE-CIX. Bgp announcement filtering, 2021.

[74] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. RFC 1883, IETF, Dec 1995.

[75] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. RFC 8200, IETF, Jul 2017.

[76] Amogh Dhamdhere and Constantine Dovrolis. The internet is flat: Modeling the transition from a transit hierarchy to a peering mesh. In *Proceedings of the 6th International COnference*, pages 21–28, 2010.

[77] Amogh Dhamdhere et al. The state of internet interconnection. Technical report, CAIDA, UC San Diego, 2018. Accessed on April 28, 2025.

[78] Christoph Dietzel, Anja Feldmann, and Thomas King. Blackholing at ixps: On the effectiveness of ddos mitigation in the wild. In *International Conference on Passive and Active Network Measurement*, pages 319–332. Springer, 2016.

[79] Christoph Dietzel, Matthias Wichtlhuber, Georgios Smaragdakis, and Anja Feldmann. Stellar: network attack mitigation using advanced blackholing. In *Proceedings of the 14th International Conference on emerging Networking EXperiments and Technologies*, pages 152–164, 2018.

[80] David Dittrich and Erin Kenneally. The Menlo Report: Ethical Principles Guiding Information and Communication Technology Research. *SSRN Electronic Journal*, 08 2012.

[81] Gert Döring. IPv6 BGP filter recommendations, 2013.

[82] J. Durand, I. Pepelnjak, and G. Doering. BGP Operations and Security. RFC 7454, IETF, Feb 2015.

[83] Zakir Durumeric, David Adrian, Phillip Stephens, Eric Wustrow, and J. Alex Halderman. Ten years of ZMap. In *Proceedings of the ACM Internet Measurement Conference (IMC)*, IMC '24, pages 139—-148, New York, NY, USA, 2024. Association for Computing Machinery.

[84] Zakir Durumeric, Eric Wustrow, and J Alex Halderman. ZMap: Fast internet-wide scanning and its security applications. In *22nd USENIX Security Symposium*, 2013.

[85] Kristaps Dzonsons, Claudio Jeker, Job Snijders, Theo de Raadt, Sebastian Benoit, and Theo Buehler. rpki-client. Available at `https://www.rpki-client.org`, 2023.

[86] W. Eddy. Transmission Control Protocol (TCP). RFC 9293, IETF, Aug 2022.

[87] Marwan Fayed, Lorenz Bauer, Vasileios Giotsas, Sami Kerola, Marek Majkowski, Pavel Odintsov, Jakub Sitnicki, Taejoong Chung, Dave Levin, Alan Mislove, Christopher A. Wood, and Nick Sullivan. The ties that un-bind: decoupling IP from web services and sockets for robust addressing agility at CDN-scale. In *ACM SIGCOMM Conference*, 2021.

[88] R. Fink and R. Hinden. 6bone (IPv6 Testing Address Allocation) Phaseout. RFC 3701, IETF, Mar 2004.

[89] Sam Fletcher, Md Zahidul Islam, et al. Comparing sets of patterns with the jaccard index. *Australasian Journal of Information Systems*, 22, 2018.

[90] Romain Fontugne, Amreesh Phokeer, Cristel Pelsser, Kevin Vermeulen, and Randy Bush. RPKI Time-of-Flight: Tracking Delays in the Management, Control, and Data Planes. In *Passive and Active Measurement: 24th International Conference, PAM 2023, Virtual Event, March 21–23, 2023, Proceedings*, pages 429–457, Virtual Event, 2023. Springer, Springer.

[91] Sheera Frenkel and Raymond Zhong. Facebook, whatsapp and instagram suffer outage. *The New York Times*, October 2021. Accessed on February 11, 2025.

[92] Jens Frieß, Donika Mirdita, Haya Schulmann, and Michael Waidner. Byzantine-secure relying party for resilient rpki. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, CCS '24, pages 49—-63, New York, NY, USA, 2024. Association for Computing Machinery.

[93] V. Fuller and T. Li. Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan. RFC 4632, IETF, Aug 2006.

[94] Lixin Gao. On Inferring Autonomous System Relationships in the Internet. *IEEE/ACM Transactions on Networking*, 9(6):733–745, 2001.

[95] Alberto García-Martínez and Marcelo Bagnulo. Measuring BGP route propagation times. *IEEE Communications Letters*, 23(12):2432–2436, 2019.

[96] Oliver Gasser, Randy Bush, Massimo Candela, and Russ Housley. Publishing End-Site Prefix Lengths. `https://www.ietf.org/archive/id/draft-gasser-opsawg-prefix-lengths-02.html`, 2024.

[97] Oliver Gasser, Randy Bush, Massimo Candela, and Russ Housley. Publishing end-site prefix lengths. Internet-Draft draft-ietf-opsawg-prefix-lengths-07, Internet Engineering Task Force (IETF), Sep 2025. Expires 22 March 2026.

[98] Petros Gigis, Matt Calder, Lefteris Manassakis, George Nomikos, Vasileios Kotronis, Xenofontas Dimitropoulos, Ethan Katz-Bassett, and Georgios Smaragdakis. Seven Years in the Life of Hypergiants' Off-Nets. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, pages 516–533, 2021.

[99] Yossi Gilad, Omar Sagga, and Sharon Goldberg. Maxlength considered harmful to the rpki. In *Proceedings of the 13th International Conference on Emerging Networking EXperiments and Technologies (CoNEXT)*, pages 101–107, Incheon, Republic of Korea, 2017. Association for Computing Machinery.

[100] Phillipa Gill, Michael Schapira, and Sharon Goldberg. Understanding the evolution of the internet as-level ecosystem. In *Proceedings of the ACM SIGCOMM*, pages 313–324, 2008.

[101] Phillipa Gill, Michael Schapira, and Sharon Goldberg. A survey of interdomain routing policies. *ACM SIGCOMM Computer Communication Review*, 44(1):28–34, 2013.

[102] Vasileios Giotsas, Matthew Luckie, Bradley Huffaker, and KC Claffy. Inferring complex AS relationships. In *Proceedings of the 2014 Conference on Internet Measurement Conference*, pages 23–30, 2014.

[103] Vasileios Giotsas, Georgios Smaragdakis, Christoph Dietzel, Philipp Richter, Anja Feldmann, and Arthur Berger. Inferring BGP blackholing activity in the internet. In *Proceedings of the ACM Internet Measurement Conference (IMC)*, pages 1–14, New York, NY, USA, 2017. Association for Computing Machinery.

[104] Sharon Goldberg. Why is it taking so long to secure internet routing? *Communications of the ACM*, 57(10):56–63, 2014.

[105] Dan Goodin. How 3 hours of inaction from Amazon cost cryptocurrency holders $235,000. `https://arstechnica.com/information-technology/2022/09/how-3-hours-of-inaction-from-amazon-cost-cryptocurrency-holders-235000/`, Sep 2022.

[106] Google. IPv6 Adoption. `https://www.google.com/intl/en/ipv6/statistics.html`. Accessed on April 28, 2025.

[107] Google. Google IPv6 Statistics. `https://www.google.com/intl/en/ipv6/statistics.html`, 2024.

[108] Google Cloud. Infrastructure overview: Secure, global, high-performance. `https://cloud.google.com/infrastructure`, 2023. Accessed on July 02, 2025.

[109] Caitlin Gray, Clemens Mosig, Randy Bush, Cristel Pelsser, Matthew Roughan, Thomas C Schmidt, and Matthias Wählisch. Bgp beacons, network tomography, and bayesian computation to locate route flap damping. In *Proceedings of the ACM Internet Measurement Conference (IMC)*, pages 492–505, New York, NY, USA, 2020. Association for Computing Machinery.

[110] NLNOG (Dutch Network Operators Group). Filtering small prefixes, 2021.

[111] GTT. GTT Completes Acquisition of KPN International. `https://www.gtt.net/us-en/resources/news/gtt-completes-acquisition-of-kpn-international`, Dec 2019.

[112] Bassam Halabi and Danny McPherson. *Internet Routing Architectures*. Cisco Press, 2 edition, 2000.

[113] Hodjat Hamidi and Maryam Jahanshahifard. The role of the internet of things in the improvement and expansion of business. *Journal of Organizational and End User Computing (JOEUC)*, 30(3):24–44, 2018.

[114] J. Hawkinson and T. Bates. Guidelines for creation, selection, and registration of an Autonomous System (AS). RFC 1930, IETF, Mar 1996.

[115] Bryton Herdes, Mingwei Zhang, and Tanner Ryan. Cloudflare 1.1.1.1 incident on june 27, 2024, July 2024. Accessed on February 11, 2025.

[116] R. Hinden and S. Deering. IP Version 6 Addressing Architecture. RFC 4291, IETF, Feb 2006.

[117] R. Hinden, R. Fink, and J. Postel. IPv6 Testing Address Allocation. RFC 2471, IETF, Dec 1998.

[118] R. Hinden and B. Haberman. Unique Local IPv6 Unicast Addresses. RFC 4193, IETF, Oct 2005.

[119] R. Hinden and J. Postel. IPv6 Testing Address Allocation. RFC 1897, IETF, Jan 1996.

[120] Tomas Hlavacek, Philipp Jeitner, Donika Mirdita, Haya Shulman, and Michael Waidner. Behind the scenes of rpki. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 1413–1426, Los Angeles, California, USA, 2022. Association for Computing Machinery.

[121] Tomas Hlavacek, Philipp Jeitner, Donika Mirdita, Haya Shulman, and Michael Waidner. Stalloris: {RPKI} downgrade attack. In *31st USENIX Security Symposium (USENIX Security '22)*, pages 4455–4471, Boston, MA, USA, 2022. USENIX Association.

[122] Tomas Hlavacek, Philipp Jeitner, Donika Mirdita, Haya Shulman, and Michael Waidner. Beyond limits: How to disable validators in secure networks. In *Proceedings of the ACM SIGCOMM 2023 Conference*, pages 950–966, New York, NY, USA, 2023. Association for Computing Machinery (ACM).

[123] S. Hollenbeck and N. Kong. Security Services for the Registration Data Access Protocol (RDAP). RFC 7481, IETF, Mar 2015.

[124] R. Housley, J. Curran, G. Huston, and D. Conrad. The Internet Numbers Registry System. RFC 7020, IETF, Aug 2013.

[125] Amanda Hsu, Frank Li, and Paul Pearce. Fiat lux: Illuminating ipv6 apportionment with different datasets. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 7(1):1–24, 2023.

[126] C. Huitema. Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs). RFC 4380, IETF, Feb 2006.

[127] G. Huston, A. Lord, and P. Smith. IPv6 Address Prefix Reserved for Documentation. RFC 3849, IETF, Jul 2004.

[128] G. Huston and G. Michaelson. Textual Representation of Autonomous System (AS) Numbers. RFC 5396, IETF, Dec 2008.

[129] G. Huston and G. Michaelson. Validation of Route Origination Using the Resource Certificate Public Key Infrastructure (PKI) and Route Origin Authorizations (ROAs). RFC 6483, IETF, Feb 2012.

[130] Geoff Huston. BGP more specifics: routing vandalism or useful?, 2017.

[131] Geoff Huston. Bgp routing table analysis report. https://www.cidr-report.org, 2024. Accessed on April 24, 2025.

[132] Geoff Huston. The IPv6 transition. https://blog.apnic.net/2024/10/22/the-ipv6-transition/, 2024.

[133] Geoff Huston. BGP Updates in 2024, 2025. Accessed on May 05, 2025.

[134] IANA. IANA IPv4 Special-Purpose Address Registry. https://www.iana.org/assignments/iana-ipv4-special-registry/iana-ipv4-special-registry.xhtml, 2023.

[135] IANA. IANA IPv6 Special-Purpose Address Registry. https://www.iana.org/assignments/iana-ipv6-special-registry/iana-ipv6-special-registry.xhtml, 2023.

[136] IANA. root-servers.org. https://root-servers.org/, 2023.

[137] IANA. Internet assigned numbers authority. https://www.iana.org/numbers, 2025. Accessed on April 28, 2025.

[138] Hironori Imai, Masayuki Okada, and Akira Kanaoka. Poster: MOAI: Multiple Origin ASes Identification for IP Prefix Hijacking and Mis-Origination. In *NDSS*, 2020.

[139] Internet Assigned Numbers Authority (IANA). Iana — protocol registries and number resources. https://www.iana.org/numbers, 2024. Accessed on July 02, 2025.

[140] IPinfo. Introducing IPinfo's probe network. https://ipinfo.io/probe-network, 2024. Accessed on September 10, 2024.

[141] IPv4 and IPv6 reports. AS65000 and AS131072 BGP Routing Table Analysis Report. https://bgp.potaroo.net/as2.0/bgp-active.html. Accessed on March, 2023.

[142] irr.net. List of routing registries, 2021.

[143] IRR.net. Internet routing registry (irr), n.d. Accessed on July 02, 2025.

[144] Isolario Project. Available at https://isolario.it/, 2021.

[145] Paul Jaccard. Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bull Soc Vaudoise Sci Nat*, 37:547–579, 1901.

[146] Quentin Jacquemart, Guillaume Urvoy-Keller, and Ernst Biersack. A longitudinal study of BGP MOAS prefixes. In *International Workshop on Traffic Monitoring and Analysis*, pages 127–138. Springer, 2014.

[147] E. Jasinska, N. Hilliard, R. Raszuk, and N. Bakker. Internet Exchange BGP Route Server. RFC 7947, IETF, Sep 2016.

[148] Lin Jin, Shuai Hao, Haining Wang, and Chase Cotton. Your Remnant Tells Secret: Residual Resolution in DDoS Protection Services. In *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 362–373. IEEE, 2018.

[149] Joel Sommers. PyTricia. `https://github.com/jsommers/pytricia/tree/master`, 2012. Accessed on May 08, 2024.

[150] Juniper. Understanding bgp confederations, 2021.

[151] B. Kara and G. Simon. Power Efficient Multi-CDN Communication over Content Steering Server. In *ACM Multimedia Systems Conference (MMSys)*, 2024.

[152] Ken Keys, Young Hyun, Matthew Luckie, and Kim Claffy. Internet-scale IPv4 alias resolution with MIDAR. *IEEE/ACM Transactions on Networking*, 21(2):383–399, 2012.

[153] Vineet Khare, Miguel Juárez, and Oliver Hohlfeld. A look at the stability of internet routing information. In *Proceedings of the Internet Measurement Conference 2020*, pages 398–405, 2020.

[154] Lars Prehn Khwaja Zubair Sediqi and Oliver Gasser. Hyper-specific prefixes: Gotta enjoy the little things in interdomain routing. `https://www.ripe.net/media/documents/11-11-hsp_presentation_for_ripe_20230522.pdf`, 2023. Presentation at RIPE 86, 23 May 2023; Accessed on September 06, 2025.

[155] T. King, C. Dietzel, J. Snijders, G. Doering, and G. Hankins. BLACKHOLE Community. RFC 7999, IETF, Oct 2016.

[156] Thomas Krenc, Robert Beverly, and Georgios Smaragdakis. AS-level BGP community usage classification. In *Proceedings of the 21st ACM Internet Measurement Conference*, pages 577–592, 2021.

[157] Thomas Krenc and Anja Feldmann. Bgp prefix delegations: a deep dive. In *Proceedings of the 2016 Internet Measurement Conference*, pages 469–475, 2016.

[158] John Kristoff, Randy Bush, Chris Kanich, George Michaelson, Amreesh Phokeer, Thomas C Schmidt, and Matthias Wählisch. On measuring rpki relying parties. In *Proceedings of the ACM Internet Measurement Conference*, pages 484–491, Virtual Event, 2020. Association for Computing Machinery.

[159] James F. Kurose and Keith W. Ross. *Computer Networking: A Top-Down Approach*. Pearson, 8th edition, 2021.

[160] Ivana Kursan and Mirela Mihić. Business intelligence: The role of the internet in marketing research and business decision-making. *Management: journal of contemporary management issues*, 15(1):69–86, 2010.

[161] Craig Labovitz. The shift to cloud: How the pandemic changed internet traffic patterns. `https://nanog.org/news-stories/the-shift-to-cloud/`, 2021. Accessed on July 02, 2025.

[162] Craig Labovitz, Scott Iekel-Johnson, Danny McPherson, Jon Oberheide, and Farnam Jahanian. Internet inter-domain traffic. *ACM SIGCOMM Computer Communication Review*, 40(4):75–86, 2010.

[163] NLnet Labs. Routinator 3000. Available at `https://www.nlnetlabs.nl/projects/rpki/routinator/`, 2023.

[164] NLnet Labs. Routinator manual. `https://routinator.docs.nlnetlabs.nl/en/stable/manual-page.html`, 2024. Accessed on November 27, 2024.

[165] Zetong Lai, Zhiwei Yan, Guanggang Geng, and Hidenori Nakazato. Issuance policies of route origin authorization with a single prefix and multiple prefixes: A comparative analysis. *International Journal of Advanced Computer Science and Applications*, 15(3):1168–1176, 2024.

[166] Ravie Lakshmanan. Orange spain faces bgp traffic hijack after ripe account hacked by malware. *The Hacker News*, January 2024. Accessed on February 11, 2025.

[167] James Larisch, Timothy Alberdingk Thijm, Suleman Ahmad, Peter Wu, Tom Arnfeld, and Marwan Fayed. Topaz: Declarative and Verifiable Authoritative DNS at CDN-Scale. In *ACM SIGCOMM Conference*, 2024.

[168] M. Lepinski and S. Kent. An Infrastructure to Support Secure Internet Routing. RFC 6480, IETF, Feb 2012.

[169] Weitong Li, Zhexiao Lin, Md Ishtiaq Ashiq, Emile Aben, Romain Fontugne, Amreesh Phokeer, and Taejoong Chung. Rovista: Measuring and analyzing the route origin validation (rov) in rpki. In *Proceedings of the 2023 ACM on Internet Measurement Conference*, pages 73–88, Montreal, QC, Canada, 2023. Association for Computing Machinery.

[170] Yanbiao Li, Hui Zou, Yuxuan Chen, Yinbo Xu, Zhuoran Ma, Di Ma, Ying Hu, and Gaogang Xie. The hanging roa: A secure and scalable encoding scheme for route origin authorization. In *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*, pages 21–30, Virtual Event, 2022. IEEE.

[171] Yuefeng Liu, Chuanxiong Guo, Ying Cui, and Jie Lin. Cloud interconnection and traffic engineering for enterprise applications. In *IEEE INFOCOM 2015*, pages 1–9, 2015.

[172] Sharon Loane. The role of the internet in the internationalisation of small and medium sized companies. *Journal of International Entrepreneurship*, 3:263–277, 2005.

[173] Matthew Luckie, Robert Beverly, William Brinkmeyer, and KC Claffy. Speedtrap: internet-scale IPv6 alias resolution. In *Proceedings of the 2013 conference on Internet measurement conference*, pages 119–126, 2013.

[174] Matthew Luckie, Amogh Dhamdhere, Bradley Huffaker, and KC Claffy. As hegemony: A robust metric for as centrality. In *Proceedings of the ACM SIGCOMM 2019 Conference*, pages 64–78, 2019.

[175] Matthew Luckie, Bradley Huffaker, Amogh Dhamdhere, Vasileios Giotsas, and KC Claffy. As relationships, customer cones, and validation. In *Proceedings of the 2013 conference on Internet measurement conference*, pages 243–256, 2013.

[176] Andra Lutu, Marcelo Bagnulo, Cristel Pelsser, Kenjiro Cho, and Rade Stanojevic. An analysis of the economic impact of strategic deaggregation. *Computer Networks*, 81:147–163, 2015.

[177] Andra Lutu, Marcelo Bagnulo, and Rade Stanojevic. An economic side-effect for prefix deaggregation. In *2012 Proceedings IEEE INFOCOM Workshops*, pages 190–195. IEEE, 2012.

[178] Andra Lutu, Cristel Pelsser, Marcelo Bagnulo, and Kenjiro Cho. The aftermath of prefix deaggregation. In *Proceedings of the 2013 25th International Teletraffic Congress (ITC)*, pages 1–8. IEEE, 2013.

[179] Charles Lynn, Stephen Kent, and Karen Seo. X. 509 extensions for ip addresses and as identifiers, 2004.

[180] Doug Madory. What can be learned from bgp hijacks targeting cryptocurrency services, November 2022. Accessed on February 11, 2025.

[181] Jean-Michel Magiron. World - as number statistics by number. `https://www-public.telecom-sudparis.eu/~maigron/rir-stats/rir-delegations/world/world-asn-by-number.html`, 2024. Accessed on May 19, 2025.

[182] Marek Majkowski. Cloudflare servers don't own ips anymore – so how do they connect to the internet?, 2022. Accessed on November 10, 2024.

[183] MANRS. Prefix filter configuration tools, 2021.

[184] Alexander Marder, Matthew Luckie, Amogh Dhamdhere, Bradley Huffaker, KC Claffy, and Jonathan M Smith. Pushing the boundaries with bdrmapit: Mapping router ownership at internet scale. In *Proceedings of the ACM Internet Measurement Conference (IMC)*, pages 56–69, New York, NY, USA, 2018. Association for Computing Machinery.

[185] Sally J McMillan and Margaret Morrison. Coming of age with the internet: A qualitative exploration of how the internet has become an integral part of young people's lives. *New media & society*, 8(1):73–95, 2006.

[186] Xiaoqiao Meng, Zhiguo Xu, Beichuan Zhang, Geoff Huston, Songwu Lu, and Lixia Zhang. Ipv4 address allocation and the bgp routing table evolution. *ACM SIGCOMM Computer Communication Review*, 35(1):71–80, 2005.

[187] D. Meyer. BGP Communities for Data Collection. RFC 4384, IETF, Feb 2006.

[188] D. Meyer, J. Schmitz, C. Orange, M. Prior, and C. Alaettinoglu. Using RPSL in Practice. RFC 2650, IETF, Aug 1999.

[189] Donika Mirdita, Haya Schulmann, Niklas Vogel, and Michael Waidner. The cure to vulnerabilities in rpki validation, 2023.

[190] Donika Mirdita, Haya Schulmann, and Michael Waidner. Sok: An introspective analysis of rpki security, 2024. Available on arXiv.

[191] Misaka Network, Inc. Misaka network, 2025. Accessed on January 12, 2025.

[192] J. Mitchell. Autonomous System (AS) Reservation for Private Use. RFC 6996, IETF, Jul 2013.

[193] P. Mohapatra, J. Scudder, D. Ward, R. Bush, and R. Austein. BGP Prefix Origin Validation. RFC 6811, IETF, Jan 2013.

[194] Johannes Naab, Patrick Sattler, Jonas Jelten, Oliver Gasser, and Georg Carle. Prefix top lists: Gaining insights with prefixes from domain-based top lists on dns deployment. In *Proceedings of the Internet Measurement Conference*, pages 351—-357, 2019.

[195] National Institute of Standards and Technology (NIST). RPKI Monitor. `https://rpki-monitor.antd.nist.gov`. Accessed on January 16, 2025.

[196] National Institute of Standards and Technology (NIST). Rpki monitor. Accessed on January 16, 2024.

[197] National Institute of Standards and Technology (NIST). Secure inter-domain routing: Technical and policy considerations. Technical Report Special Publication 800-189 (Draft), NIST, 2019. Draft Version.

[198] Marcin Nawrocki, Jeremias Blendin, Christoph Dietzel, Thomas C Schmidt, and Matthias Wählisch. Down the black hole: dismantling operational practices of bgp blackholing at ixps. In *Proceedings of the Internet Measurement Conference*, pages 435–448, 2019.

[199] RIPE NCC. Ripe atlas measurement dumps - [2021/05/17-23], 2021.

[200] RIPE NCC. Ripe atlas probes, 2024. `https://atlas.ripe.net`.

[201] P. Nikander, J. Laganier, and F. Dupont. An IPv6 Prefix for Overlay Routable Cryptographic Hash Identifiers (ORCHID). RFC 4843, IETF, Apr 2007.

[202] Noction. Bgp prefix filtering, 2021.

[203] Leo Oliver, Gautam Akiwate, Matthew Luckie, Ben Du, and KC Claffy. Stop, DROP, and ROA: Effectiveness of Defenses through the lens of DROP. In *Proceedings of the 22nd ACM Internet Measurement Conference*, pages 730–737, 2022.

[204] OpenINTEL. OpenINTEL Data Access. `https://openintel.nl/data-access/`, 2015. Accessed on April 10, 2024.

[205] Eric Osterweil, Terry Manderson, Russ White, and Danny McPherson. Sizing estimates for a fully deployed rpki. Technical Report 1120005, Verisign Labs, 2012.

[206] Packet Clearing House. Internet exchange directory. `https://www.pch.net/ixp/dir`, 2025. Accessed on April 28, 2025.

[207] Ramakrishna Padmanabhan, Zhihao Li, Dave Levin, and Neil Spring. UAv6: alias resolution in IPv6 using unused addresses. In *Passive and Active Measurement: 16th International Conference, PAM 2015, New York, NY, USA, March 19-20, 2015, Proceedings 16*, pages 136–148. Springer, 2015.

[208] Ramakrishna Padmanabhan, John P Rula, Philipp Richter, Stephen D Strowes, and Alberto Dainotti. DynamIPs: Analyzing address assignment practices in IPv4 and IPv6. In *Proceedings of the 16th International Conference on emerging Networking EXperiments and Technologies*, pages 55–70, 2020.

[209] Craig Partridge and Mark Allman. Ethical Considerations in Network Measurement Papers. *Commun. ACM*, 59(10):58—-64, sep 2016.

[210] Larry L. Peterson and Bruce S. Davie. *Computer Networks: A Systems Approach*. Morgan Kaufmann, 6th edition, 2021.

[211] Potaroo. Bgp routing table analysis. `https://bgp.potaroo.net`. Accessed on January 15, 2025.

[212] Potaroo. AS6447 BGP Routing Table Analysis Report, May 2021.

[213] Potaroo. AS131072 IPv6 BGP Table Data. `https://bgp.potaroo.net/v6/as2.0/index.html`, 2024.

[214] Lars Prehn, Franziska Lichtblau, and Anja Feldmann. When wells run dry: the 2020 IPv4 address market. In *Proceedings of the 16th International Conference on emerging Networking EXperiments and Technologies*, pages 46–54, 2020.

[215] CAIDA Project. As rank: Visualizing the hierarchy of the internet. *Center for Applied Internet Data Analysis (CAIDA)*, 2023.

[216] OpenBSD Project. rpki-client manual. `https://man.openbsd.org/rpki-client`, 2024. Accessed on November 27, 2024.

[217] Lancheng Qin, Dan Li, Ruifeng Li, and Kang Wang. Themis: Accelerating the Detection of Route Origin Hijacking by Distinguishing Legitimate and Illegitimate MOAS. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 4509–4524, 2022.

[218] Junxue Qiu, Sandeep Singh, Ethan Katz-Bassett, Ítalo Cunha, et al. Measuring and characterizing cloud provider connectivity. In *Proceedings of the ACM Internet Measurement Conference (IMC)*, pages 100–115, New York, NY, USA, 2023. Association for Computing Machinery.

[219] Bruno Quoitin, Cristel Pelsser, Louis Swinnen, Olivier Bonaventure, and Steve Uhlig. Interdomain traffic engineering with bgp. *IEEE Communications magazine*, 41(5):122–128, 2003.

[220] Rapid7. Rapid7 Open Data, May 2021.

[221] Y. Rekhter, T. Li, and S. Hares. A Border Gateway Protocol 4 (BGP-4). RFC 4271, IETF, Jan 2006.

[222] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear. Address Allocation for Private Internets. RFC 1918, IETF, Feb 1996.

[223] Internet Health Report. ip2asn, November 2023. `https://github.com/InternetHealthReport/ip2asn`.

[224] A. Retana, R. White, V. Fuller, and D. McPherson. Using 31-Bit Prefixes on IPv4 Point-to-Point Links. RFC 3021, IETF, Dec 2000.

[225] Andreas Reuter, Randy Bush, Italo Cunha, Ethan Katz-Bassett, Thomas C Schmidt, and Matthias Wählisch. Towards a rigorous methodology for measuring adoption of rpki route validation and filtering. *ACM SIGCOMM Computer Communication Review*, 48(1):19–27, 2018.

[226] Andreas Reuter, Matthias Wählisch, and Thomas C Schmidt. Rpki miro: Monitoring and inspection of rpki objects. *ACM SIGCOMM Computer Communication Review*, 45(4):107–108, 2015.

[227] Philipp Richter, Mark Allman, Randy Bush, and Vern Paxson. A primer on IPv4 scarcity. *ACM SIGCOMM Computer Communication Review*, 45(2):21–31, 2015.

[228] RIPE NCC. YouTube Hijacking: A RIPE NCC RIS case study. https://www.ripe.net/publications/news/industry-developments/youtube-hijacking-a-ripe-ncc-ris-case-study, Mar 2008.

[229] RIPE NCC. Ripe database documentation: Getting started. Technical Report RIPE-679, RIPE Network Coordination Centre, 2017. Accessed on May 19, 2025.

[230] RIPE NCC. RIPE Atlas measurement platform, May 2021.

[231] RIPE NCC. RIPE RPKI Snapshots, May 2021. RPKI snapshots from 1st April 2015, until 1st April 2021.

[232] RIPE NCC. RIPE Routing Information Service. http://www.ripe.net/ris/, 2022. RIB snapshots from January 2017 to January 2023.

[233] RIPE NCC. What are internet exchange points? https://www.ripe.net/publications/docs/ripe-765, 2022. Accessed on July 02, 2025.

[234] RIPE NCC. RPKI snapshots [2020/09/01-2024/09/01]. https://ftp.ripe.net/rpki/, 2023. Accessed on February, 2023.

[235] RIPE NCC. Routing registry. https://www.ripe.net/manage-ips-and-asns/db/routing-registry, n.d. Accessed on July 02, 2025.

[236] RIPE Routing Information Service. http://www.ripe.net/ris/, 2021. RIB snapshots from January 2010 to April 2021.

[237] Nils Rodday, Ítalo Cunha, Randy Bush, Ethan Katz-Bassett, Gabi Dreo Rodosek, Thomas C Schmidt, and Matthias Wählisch. The resource public key infrastructure (rpki): A survey on measurements and future prospects. *IEEE Transactions on Network and Service Management*, 21(2):2353–2373, 2023.

[238] Ben Rosenblum and Nick Dunkin. The Multi-CDN Dilemma: Aggregated Edge Networks at Scale. In *SCTE TechExpo24*, 2024.

[239] Matthew Roughan, Walter Willinger, Olaf Maennel, Debbie Perouli, and Randy Bush. 10 lessons from 10 years of measuring and modeling the internet's autonomous systems. *IEEE Journal on Selected Areas in Communications*, 29(9):1810–1821, 2011.

[240] Route Views. RouteViews project. https://www.routeviews.org/routeviews/index.php/collectors/, 2024. Accessed on April 10, 2024.

[241] Route Views Project – University of Oregon. Available at http://www.routeviews.org/, 2021. RIB snapshots from January 2010 to April 2021.

[242] Erik Rye and Dave Levin. IPv6 Hitlists at Scale: Be Careful What You Wish For. In *Proceedings of the ACM SIGCOMM 2023 Conference*, pages 904–916, 2023.

[243] Ville Satopaa, Jeannie Albrecht, David Irwin, and Barath Raghavan. Finding a "Kneedle" in a Haystack: Detecting Knee Points in System Behavior. In *2011 31st International Conference on Distributed Computing Systems Workshops*, pages 166–171. IEEE, 2011.

[244] Patrick Sattler, Johannes Zirngibl, Mattijs Jonker, Oliver Gasser, Georg Carle, and Ralph Holz. Packed to the brim: Investigating the impact of highly responsive prefixes on internet-wide measurement campaigns. *Proc. ACM Netw.*, 1(CoNEXT3), 2023.

[245] Quirin Scheitle, Oliver Gasser, Minoo Rouhi, and Georg Carle. Large-scale classification of IPv6-IPv4 siblings with variable clock skew. In *2017 Network Traffic Measurement and Analysis Conference (TMA)*, pages 1–9. IEEE, 2017.

[246] D. Schinazi and T. Pauly. Happy Eyeballs Version 2: Better Connectivity Using Concurrency. RFC 8305, IETF, Dec 2017.

[247] Johann Schlamp, Ralph Holz, Oliver Gasser, Andreas Korsten, Quentin Jacquemart, Georg Carle, and Ernst W. Biersack. Investigating the Nature of Routing Anomalies: Closing in on Subprefix Hijacking Attacks. In *Traffic Monitoring and Analysis Workshop 2015*, Apr 2015.

[248] Brandon Schlinker, Todd Arnold, Italo Cunha, and Ethan Katz-Bassett. PEERING: Virtualizing BGP at the Edge for Research. In *Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies*, Orlando, FL, December 2019.

[249] Haya Schulmann, Niklas Vogel, and Michael Waidner. Rpki: Not perfect but good enough, 2024. Available on arXiv.

[250] Khwaja Zubair Sediqi. Measuring hyper-specific prefixes in the wild. `https://blog.apnic.net/2022/09/01/measuring-hyper-specific-prefixes-in-the-wild/`, 2022. Published: 1 September 2022; Accessed on September 06, 2025.

[251] Khwaja Zubair Sediqi. Syncing with rpki – exploring causes of delay in relying party synchronization (raci submission). Video (embedded on RIPE 88 website), `https://ripe88.ripe.net/archives/video/1384/`, 2024. Presented at RIPE 88, 23 May 2024; Accessed on September 06, 2025.

[252] Khwaja Zubair Sediqi, Anja Feldmann, and Oliver Gasser. Live long and prosper: Analyzing long-lived moas prefixes in bgp. In *2023 7th Network Traffic Measurement and Analysis Conference (TMA)*, pages 1–10, Napoli, Italy, 2023. IEEE.

[253] Khwaja Zubair Sediqi, Anja Feldmann, and Oliver Gasser. Live long and prosper: Analyzing long-lived moas prefixes in bgp. In *Proceedings of the 2023 7th Network Traffic Measurement and Analysis Conference (TMA)*. IEEE, 2024. Presentation at SEE12, 23 April 2024; Accessed on September 06, 2025.

[254] Khwaja Zubair Sediqi, Lars Prehn, and Oliver Gasser. Hyper-Specific Prefixes: Gotta Enjoy the Little Things in Interdomain Routing. *ACM SIGCOMM Computer Communication Review (CCR)*, 52(2), Jun 2022.

[255] Pavlos Sermpezis, Vasileios Kotronis, Konstantinos Arakadakis, and Athena Vakali. Estimating the impact of bgp prefix hijacking. In *2021 IFIP Networking Conference (IFIP Networking)*, pages 1–10. IEEE, 2021.

[256] Pavlos Sermpezis, Vasileios Kotronis, Alberto Dainotti, and Xenofontas Dimitropoulos. A survey among network operators on BGP prefix hijacking. *ACM SIGCOMM Computer Communication Review*, 48(1):64–69, 2018.

[257] Pavlos Sermpezis, Vasileios Kotronis, Petros Gigis, Xenofontas Dimitropoulos, Danilo Cicalese, Alistair King, and Alberto Dainotti. Artemis: Neutralizing bgp hijacking within a minute. *IEEE/ACM Transactions on Networking*, 26(6):2471–2486, 2018.

[258] Georgos Siganos and Michalis Faloutsos. Analyzing bgp policies: Methodology and tool. In *IEEE INFOCOM 2004*, volume 3, pages 1640–1651. IEEE, 2004.

[259] Felipe Victolla Silveira. Rpki repositories and the ripe database in the cloud, 2021. (Accessed on January 14, 2025).

[260] Site24x7. `https://www.site24x7.com`, 2006. Accessed on May 07, 2024.

[261] P. Smith and R. Evans. RIPE-532 - RIPE Routing Working Group Recommendations on IPv6 Route Aggregation, Nov 2011.

[262] P. Smith, R. Evans, and M. Hughes. RIPE-399 - RIPE Routing Working Group Recommendations on Route Aggregation, Dec 2006.

[263] J. Snijders and B. Maddison. Rpki validation re-reconsidered. `https://datatracker.ietf.org/doc/draft-spaghetti-sidrops-rpki-validation-update-03/`, 2023.

[264] Job Snijders. Internet routing registries: Misused and diminishing. `https://www.nanog.org/meetings/nanog71/presentations/Monday/Snijders.pdf`, 2017. Presentation at NANOG 71.

[265] Job Snijders and Geoff Huston. A profile for signed prefix lists for use in the resource public key infrastructure (rpki). `https://datatracker.ietf.org/doc/draft-ietf-sidrops-rpki-prefixlist/`, 2024.

[266] João Luís Sobrinho. An algebraic theory of BGP routing policies. In *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '03)*, pages 49–60, Karlsruhe, Germany, 2003. ACM.

[267] Raffaele Sommese, Leandro Bertholdo, Gautam Akiwate, Mattijs Jonker, Roland van Rijswijk-Deij, Alberto Dainotti, KC Claffy, and Anna Sperotto. Manycast^2: Using anycast to measure anycast. In *Proceedings of the ACM Internet Measurement Conference*, pages 456–463, 2020.

[268] Raffaele Sommese, Roland van Rijswijk-Deij, and Mattijs Jonker. This Is a Local Domain: On Amassing Country-Code Top-Level Domains from Public Data. *ACM SIGCOMM Computer Communication Review CCR*, 2024.

[269] Thorvald Sorensen. A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on danish commons. *Biologiske skrifter*, 5:1–34, 1948.

[270] K. Sriram, D. Montgomery, D. McPherson, E. Osterweil, and B. Dickson. Problem Definition and Classification of BGP Route Leaks. RFC 7908, IETF, Jun 2016.

[271] Florian Streibelt, Franziska Lichtblau, Robert Beverly, Anja Feldmann, Cristel Pelsser, Georgios Smaragdakis, and Randy Bush. BGP communities: Even more worms in the routing can. In *Proceedings of the ACM Internet Measurement Conference (IMC)*, pages 279–292, New York, NY, USA, 2018. Association for Computing Machinery.

[272] Florian Streibelt, Patrick Sattler, Franziska Lichtblau, Carlos H. Gañán, Anja Feldmann, Oliver Gasser, and Tobias Fiebig. How Ready is DNS for an IPv6-Only World? In *Passive and Active Measurement Conference (PAM)*, 2023.

[273] Stephen Strowes and Colin Petrie. BGP Even-More Specifics in 2017, 2017.

[274] Cisco Systems. Cisco annual internet report (2018–2023) white paper, March 2020. Accessed on February 11, 2025.

[275] Szymkiewicz–Simpson. On the numerical characterization of closed populations. *Annales de la Société Polonaise de Mathématique*, 11:66–77, 1934.

[276] Cecilia Testart, Philipp Richter, Alistair King, Alberto Dainotti, and David Clark. Profiling BGP serial hijackers: capturing persistent misbehavior in the global routing table. In *Proceedings of the Internet Measurement Conference*, pages 420–434, 2019.

[277] Cecilia Testart, Philipp Richter, Alistair King, Alberto Dainotti, and David Clark. To filter or not to filter: Measuring the benefits of registering in the rpki today. In *Passive and Active Measurement: 21st International Conference, PAM 2020, Eugene, Oregon, USA, March 30–31, 2020, Proceedings*, pages 71–87, Eugene, Oregon, USA, 2020. Springer.

[278] The Broadband Forum. TR-069: CPE WAN Management Protocol, Mar 2018.

[279] BGP Tools. Bgp tools, November 2023. `https://bgp.tools`.

[280] TUM. ZMapv6 on GitHub. `https://github.com/tumi8/zmap`, 2024.

[281] University of Oregon. Route Views Project. `http://www.routeviews.org/`, 2022. RIB snapshots from January 2017 to January 2023.

[282] Koen van Hove, Jeroen van der Ham-de Vos, and Roland van Rijswijk-Deij. rpkiller: Threat analysis of the bgp resource public key infrastructure. *Digital Threats: Research and Practice*, 4(4):1–24, 2023.

[283] Kevin Vermeulen, Burim Ljuma, Vamsi Addanki, Matthieu Gouel, Olivier Fourmaux, Timur Friedman, and Reza Rejaie. Alias resolution based on icmp rate limiting. In *Passive and Active Measurement: 21st International Conference, PAM 2020, Eugene, Oregon, USA, March 30–31, 2020, Proceedings 21*, pages 231–248. Springer, 2020.

[284] Pierre-Antoine Vervier, Olivier Thonnard, and Marc Dacier. Mind your blocks: On the stealthiness of malicious bgp hijacks. In *NDSS*, 2015.

[285] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczyński, and Wouter Joosen. Tranco List. `https://tranco-list.eu/`, 2019. Accessed on April 10, 2024.

[286] Q. Vohra and E. Chen. BGP Support for Four-Octet Autonomous System (AS) Number Space. RFC 6793, IETF, Dec 2012.

[287] Z. Wenzel, J. Klensin, R. Bush, and S. Huter. Guide to Administrative Procedures of the Internet Infrastructure. RFC 2901, IETF, Aug 2000.

[288] Wikipedia contributors. Tier 1 network — wikipedia, the free encyclopedia, 2024. Accessed on July 02, 2025.

[289] S. Williamson and L. Nobile. Transition of Nic Services. RFC 1261, IETF, Sep 1991.

[290] D. Wing and A. Yourtchenko. Happy Eyeballs: Success with Dual-Stack Hosts. RFC 6555, IETF, Apr 2012.

[291] Florian Wohlfart, Nikolaos Chatzis, Caglar Dabanoglu, Georg Carle, and Walter Willinger. Leveraging interconnections for performance: the serving infrastructure of a large cdn. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, pages 206–220, 2018.

[292] xTom. xtom facilities, 2025. Accessed on January 12, 2025.

[293] Z. Yan, R. Bush, G. Geng, T. de Kock, and J. Yao. Avoiding Route Origin Authorizations (ROAs) Containing Multiple IP Prefixes. RFC 9455, IETF, Aug 2023.

[294] Z. Chen, Z. Bischof, C. Testart, A. Dainotti. AS to Organization Mapping. `https://github.com/InetIntel/Dataset-AS-to-Organization-Mapping`, 2024. Covers data from September 2020 to September 2024.

[295] Man Zeng, Xiaohong Huang, Pei Zhang, and Dandan Li. Understanding the impact of outsourcing mitigation against bgp prefix hijacking. *Computer Networks*, 202:108650, 2022.

[296] Beichuan Zhang, Daniel Massey, and Lixia Zhang. Destination reachability and BGP convergence time [border gateway routing protocol]. In *IEEE Global Telecommunications Conference, 2004. GLOBECOM'04.*, volume 3, pages 1383–1389. IEEE, 2004.

[297] Xiaoliang Zhao, Dan Pei, Lan Wang, Dan Massey, Allison Mankin, S Felix Wu, and Lixia Zhang. An analysis of BGP multiple origin AS (MOAS) conflicts. In *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, pages 31–35, 2001.

[298] Maya Ziv, Liz Izhikevich, Kimberly Ruth, Katherine Izhikevich, and Zakir Durumeric. Asdb: a system for classifying owners of autonomous systems. In *Proceedings of the 21st ACM Internet Measurement Conference*, pages 703–719, 2021.