

Sibling Prefixes: Identifying Similarities in IPv4 and IPv6 Prefixes

Fariba Osali
Max Planck Institute for Informatics
Saarbrücken, Germany
fosali@mpi-inf.mpg.de

Khawaja Zubair Sediqi
Max Planck Institute for Informatics,
Saarland University
Saarbrücken, Germany
zsediqi@mpi-inf.mpg.de

Oliver Gasser
IPinfo
Seattle, USA
oliver@ipinfo.io

Abstract

Since the standardization of IPv6 in 1998, both versions of the Internet Protocol have coexisted in the Internet. Clients usually run algorithms such as Happy Eyeballs, to decide whether to connect to an IPv4 or IPv6 endpoint for dual-stack domains. To identify whether two addresses belong to the same device or service, researchers have proposed different forms of alias resolution techniques. Similarly, one can also form siblings of IPv4 and IPv6 addresses belonging to the same device. Traditionally, all of these approaches have focused on individual IP addresses.

In this work, we propose the concept of “sibling prefixes”, where we extend the definition of an IPv4-IPv6 sibling to two IP prefixes—one IPv4 prefix and its sibling IPv6 prefix. We present a technique based on large-scale DNS resolution data to identify 76k IPv4-IPv6 sibling prefixes. We find sibling prefixes to be relatively stable over time. We present SP-Tuner algorithm to tune the CIDR size of sibling prefixes and improve the perfect match siblings from 52% to 82%. For more than half of sibling prefixes, the organization names for their IPv4 and IPv6 origin ASes are identical, and 60% of all sibling prefixes have at least one of the prefixes with a valid ROV status in RPKI. Furthermore, we identify sibling prefixes in 24 hypergiant and CDN networks. Finally, we plan to regularly publish a list of sibling prefixes to be used by network operators and fellow researchers in dual-stack studies.

CCS Concepts

• Networks → Network measurement.

Keywords

Sibling Prefix; IPv4; IPv6; Dual-stack Domains; DNS

ACM Reference Format:

Fariba Osali, Khawaja Zubair Sediqi, and Oliver Gasser. 2025. 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. <https://doi.org/10.1145/3730567.3732917>

1 Introduction

Since the beginnings of the standardization of IPv6 in 1998 [20], 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 [18]. With the exhaustion of the IPv4 address space [47, 49] and the preference for IPv6 over IPv4 in connection establishments [56, 66], IPv6 sees continuous growth, with more than 219,000 IPv6 prefixes in the global routing table in November 2024 [46], more than 75% of top 1k popular domains being accessible over IPv6 in August 2022 [61], and more than 45% of clients accessing Google via IPv6 in November 2024 [28]. 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 [1, 2, 5, 6, 55]. To date, however, siblings have only been analyzed on an IP address level.

In this paper, 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 paper are as follows:



This work is licensed under a Creative Commons Attribution 4.0 International License. *IMC '25, Madison, WI, USA*

© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1860-1/2025/10
<https://doi.org/10.1145/3730567.3732917>

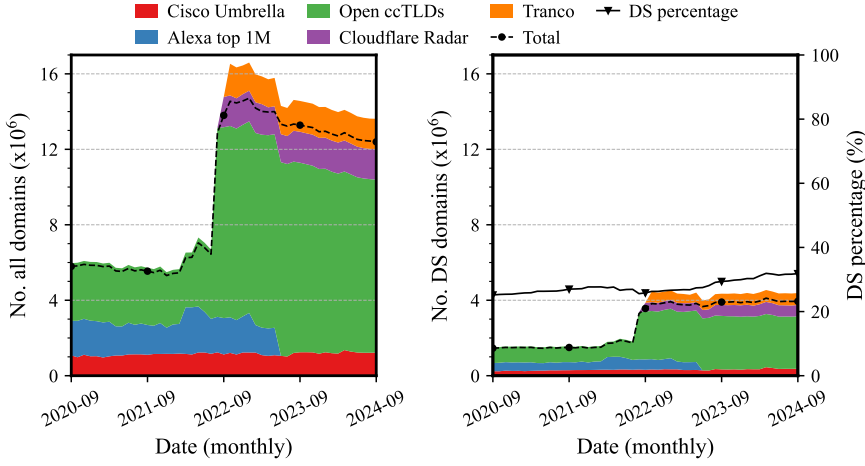


Figure 1: Number of all domains (left) and dual-stack (DS) domains (right) over time in the OpenINTEL dataset.

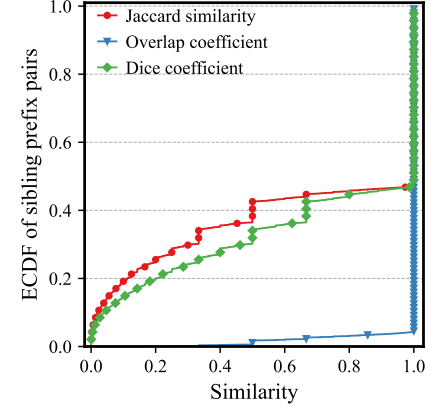


Figure 2: Comparison of Jaccard, Dice, and overlap coefficient similarity metrics.

- **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 Section 3.
- **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 Sections 3.3 and 3.4. 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 (ROV) status, finding over 60% of sibling prefix pairs having at least one valid origin AS, see Section 4.
- **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 Section 6.

2 Datasets

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

2.1 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 the OpenINTEL DNS dataset [42] to obtain DNS resolution results from large-scale measurements. The OpenINTEL dataset consists of DNS resolution results of the Alexa top 1M [3], Cloudflare Radar [17], Tranco [65], Cisco Umbrella [16], 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 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 dual-stack (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” TLD domains 6.35M when it was added to the set of open ccTLDs [59]. 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 the paper, we leverage 3.95 million unique DS domains to identify sibling prefixes.

2.2 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 [52] 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 paper.

2.3 AS to Organization Mapping Datasets

For all ASes of sibling prefixes, we identify their organization using the AS organization dataset from Chen et al. [15]. 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 [10] for the analysis prior to October 2022, and the Chen et al. [15] 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 Section 4.5.

2.4 Hypergiants and CDN Datasets

We use the list of hypergiant (HG) networks provided by Böttger et al. [9] and Gigis et al. [27], and the list of content distribution networks (CDNs) [13] 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 Section 4.7.

2.5 ASdb Datasets

To infer the business type of the origin ASes for sibling prefixes, we use the ASdb dataset [67]. 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 Section 4.6).

2.6 RPKI Dataset

We examine the validity of each pair of sibling prefixes in the Resource Public Key Infrastructure (RPKI). For this purpose, we download the RPKI data of all five Regional Internet Registries (RIRs) [50] from September 2020 to September 2024 for every month. With the RPKI data, we identify the route origin validation (ROV) state for sibling prefix pairs. We use the RPKI dataset in Section 4.8.

2.7 Port Scan Dataset

We scan a set of 14 well-known ports [4, 22, 54] 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 Section 3.6.

3 Methodology

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 3 provides an overview of our methodology to identify sibling prefixes.

3.1 Identifying Sibling Prefixes

As shown in Figure 3, identifying sibling prefixes consists of four steps: (1) Identifying dual-stack (DS) 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 (DS) 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 3. 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 Section 2) 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 [32] 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):

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

Note that we also investigate alternative similarity metrics, see Section 3.2.

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

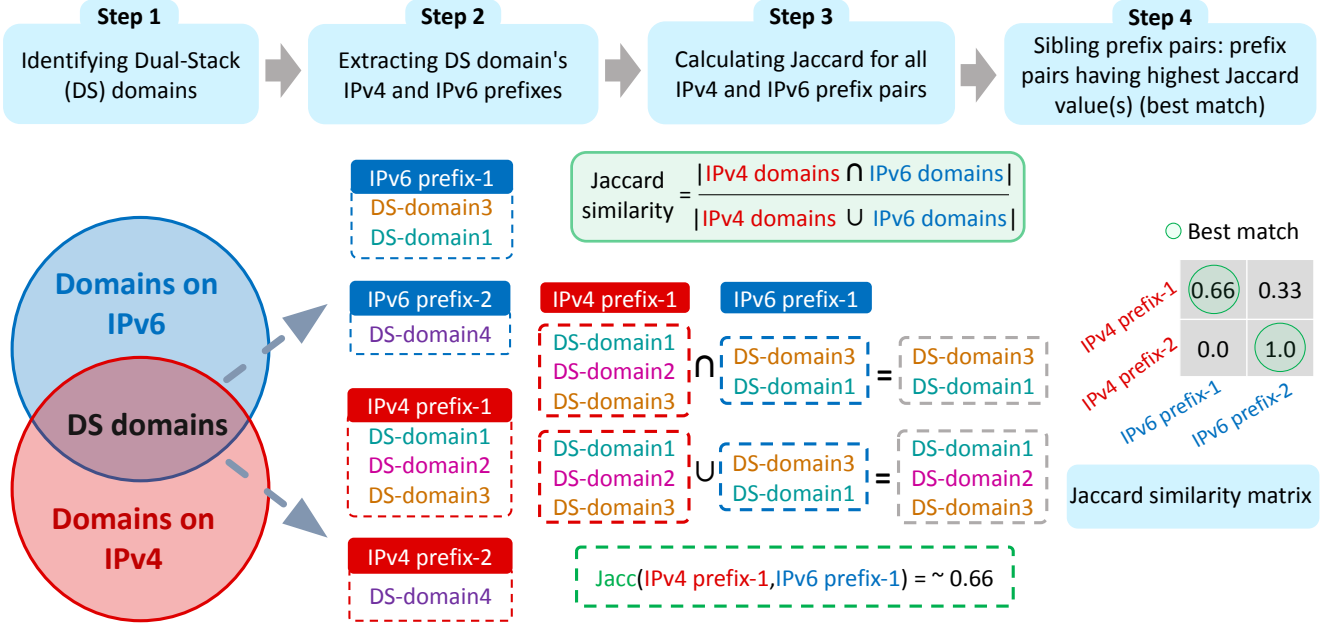


Figure 3: Methodology to identify sibling prefixes using Jaccard similarity index.

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 the paper.

3.2 Examining Possible Similarity Metrics

Selecting a suitable similarity metric is crucial for identifying sibling prefixes. We compare the overlap coefficient [62] and Dice coefficient [60] with the Jaccard similarity index [32].

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):

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

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):

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

Next, we identify sibling prefixes by applying the Jaccard similarity index, overlap coefficient, and Dice coefficient on DS domains and compare the results. Figure 2 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 2 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 to slight overlaps in sets [60]. On the other hand, the Jaccard index provides less biased and more balanced similarity results for variable set sizes [25]. In our study, as IPv4 and IPv6 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.

3.3 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

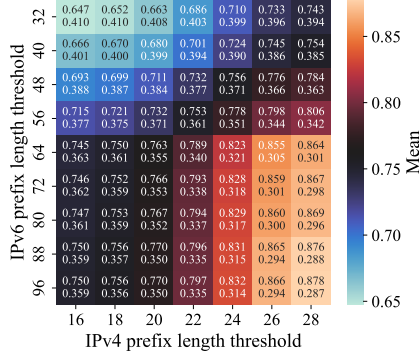


Figure 4: Heatmap of SP-Tuner Algorithm Performance: Mean Jaccard Index (top) and Standard Deviation (bottom) across IPv4 (x-axis) and IPv6 (y-axis) CIDR size thresholds.

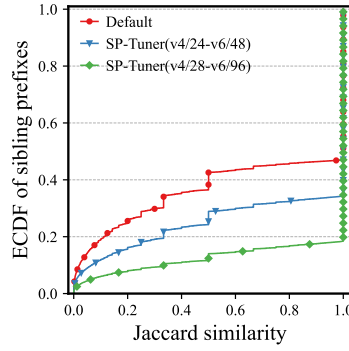


Figure 5: CDF of Jaccard similarity for sibling prefixes comparing default values with SP-Tuner algorithm at routable and optimal IPv4/IPv6 thresholds.

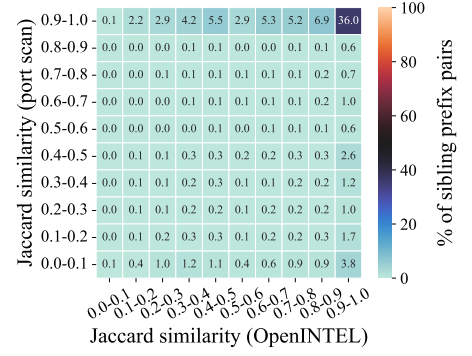


Figure 6: Heatmap of open port scanning: Each cell shows the percentage of sibling prefixes, with the x-axes and y-axes being the Jaccard values for DNS (OpenINTEL) and port scan data, respectively.

thoroughly analyze both the less specific (see Appendix A.1 for more details) and more specific variants of SP-Tuner algorithm. In the remainder of the paper we leverage SP-Tuner using more specific CIDR sizes as it produces better Jaccard similarity 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 [33] 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.

3.4 Effectiveness of SP-Tuner

To evaluate the effectiveness of the SP-Tuner algorithm, we apply the algorithm on the most recent domain snapshot from September 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 algorithm calculates the Jaccard value of sibling prefixes by

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
 1: **for** (sibling_prefix_v4, sibling_prefix_v6) ∈ sibling_prefix_pairs **do**
 2: tree_v4 ← sibling_prefix_v4
 3: tree_v6 ← sibling_prefix_v6
 4: new_jacc ← 0
 5: **while** new_jacc ≤ curr_jacc **do**
 6: prefix_v4_subprefixes ← getNextSubprefixes(sibling_prefix_v4)
 7: prefix_v6_subprefixes ← getNextSubprefixes(sibling_prefix_v6)
 8: **for** prefix_v4 ∈ prefix_v4_subprefixes **do**
 9: **for** prefix_v6 ∈ prefix_v6_subprefixes **do**
 10: new_jacc ← max(Jaccard(prefix_v4, prefix_v6))
 11: **if** HasBranch(prefix_v4, prefix_v6) **then**
 12: sibling_prefix_pairs ← UpdateBranches()
 13: **if** (prefix_v4_len ≥ prefix_v4_len_thresh)
 14: **and** (prefix_v6_len ≥ prefix_v6_len_thresh) **then**
 15: **return** (prefix_v4, prefix_v6)

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 4 shows our results for SP-Tuner thresholds up to /28 IPv4 and /96 IPv6 prefix sizes. We refer the interested reader to Figure 19 in Appendix A.2 for the complete heatmap. The color in Figure 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 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 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 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 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.

3.5 Ground Truth Evaluation

To compare our identified sibling prefixes with a real-world deployment, we use RIPE Atlas probes [40] 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 3). 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 (VPSSes) from IPinfo’s probe network [31] hosted by different VPS providers (e.g., Google, Azure, Vultr, AWS). For the VPSSes where we get both an IPv4 and IPv6 address match in our siblings, 53 are within our best-match 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.

3.6 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 [23] and ZMapv6 [63] on 14 common open ports [4, 22, 54], 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 6 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 scan results for sibling prefixes. Moreover, the values in the topmost row of Figure 6 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.

3.7 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 Section 2.1). 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

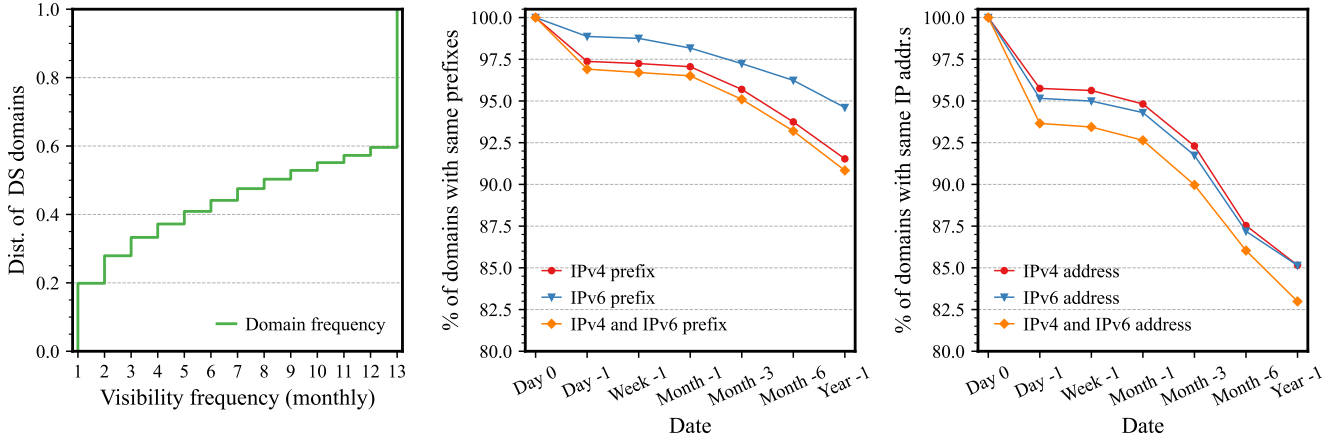


Figure 7: Consistently visible DS domains (left), prefix changes (center), and address changes (right).

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 Section 3.6). 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.

3.8 Ethical Considerations

In this work, we follow ethical Internet measurement guidelines [21, 23, 45]. First, we use publicly available datasets (see Section 2). Second, for ground truth evaluation and comparison of DNS data with open ports (see Sections 3.5 and 3.6), 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.

4 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 (DS) 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.

4.1 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 7 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 7 shows the percentage of DS domains having 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

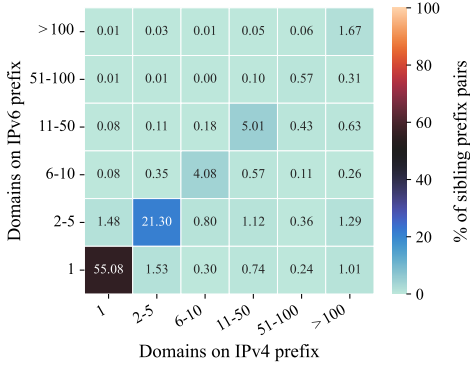


Figure 8: Sibling prefix pairs count and classification based on number of dual-stack domains in each IPv4 and IPv6 prefix of sibling prefixes

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 7. 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.

To summarize: 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.

4.2 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 8, 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.

To summarize: The majority of sibling prefixes contain single domains. Moreover, we observe a relatively high number of IPv4

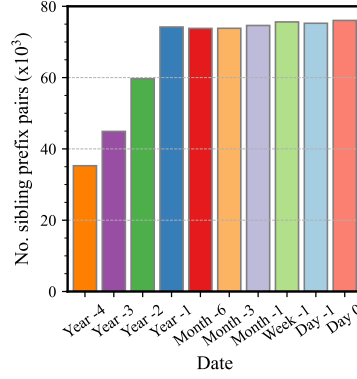


Figure 9: Number of sibling prefixes at different points in time. (Reference date: September 11, 2024)

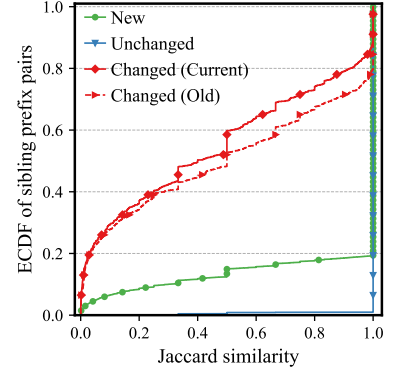


Figure 10: Improvement of Jaccard similarity of sibling prefixes over time.

sibling prefixes with more than 100 dual-stack domains. Finally, sibling prefixes tend to have a similar number of domains for IPv4 and IPv6.

4.3 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 9. 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 Section 2.1) 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 the OpenINTEL data in May 2023 (see Figure 1 right). When looking at the number of sibling prefixes in Figure 14 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 analysis still provide valuable insights into sibling prefix changes over time.

Figure 10 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

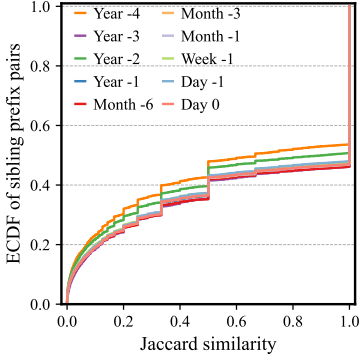


Figure 11: Jaccard similarity of sibling prefixes for various data points in time. (Reference date: September 11, 2024)

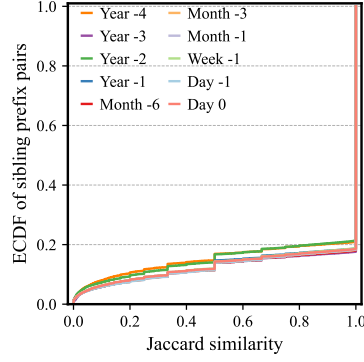


Figure 12: CDF of Jaccard similarity values for sibling prefixes on various data snapshots in time.

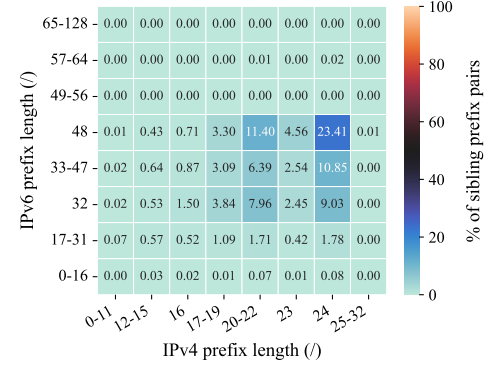


Figure 13: Distribution of CIDR sizes in sibling prefixes.

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 11 with sibling prefixes after applying SP-Tuner in Figure 12. 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 12, 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.

To summarize: 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 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.

4.4 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 Appendix A.2, Figure 36). Therefore, we focus on the default case as it shows a more nuanced distribution as can be seen in Figure 13. 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 [57] more specific than /24 in IPv4 and /48 in IPv6 are very rare among sibling prefixes.

To summarize: 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.

4.5 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 [14]. 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 14 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

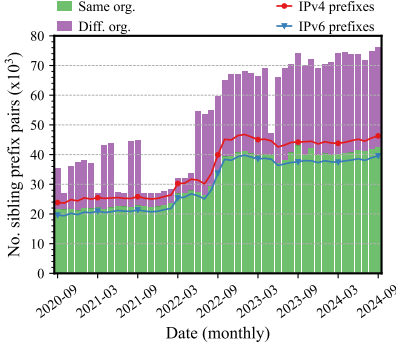


Figure 14: Sibling prefix pairs count for the same and different organizations over time. Lines represent the number of unique IPv4 and IPv6 prefixes in sibling prefixes.

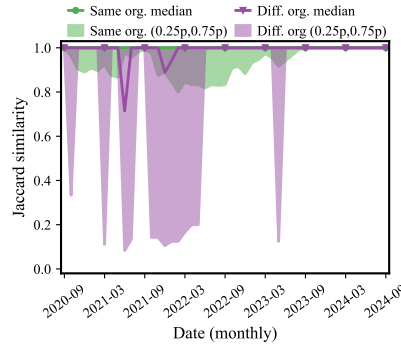


Figure 15: Median Jaccard similarity values of sibling prefix pairs for same and different organizations over time.

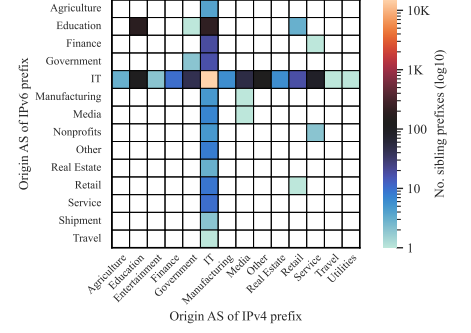


Figure 16: Business types heatmap for origin ASes of sibling prefixes, with IPv4 on x-axis and IPv6 on y-axis, indicating the count of sibling pairs in each cell.

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 [58], 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), 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 15 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. Moreover, other sites like Catchpoint [12], services that use multi-CDN [34, 51], alongside, some authoritative DNS that return more than one address to the queries might be potential reasons for different organizations category.

To summarize: 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.

4.6 Business Types

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 [67] 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). We also present a more inclusive analysis of the business relationship for sibling prefixes, including those with the same origin AS number in Appendix A.4.

In Figure 16 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

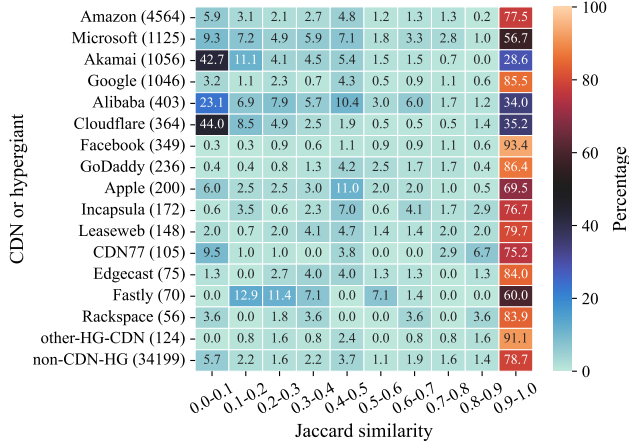


Figure 17: Heatmap of Jaccard similarity values (%) for sibling prefixes of Hypergiant and CDN networks, and non-CDN-Hypergiant ASes.

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 summarize: 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.

4.7 Hypergiants and CDNs

Next, we analyze to what extent sibling prefixes are used by hypergiants (HGs) and content distribution networks (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. [15] and use the list of publicly known HGs [9] and CDNs [13]. Note, we only consider SP-Tuner output sibling prefixes where IPv4 and IPv6 prefixes belong to the same organization of HG or CDN.

Figure 17 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.

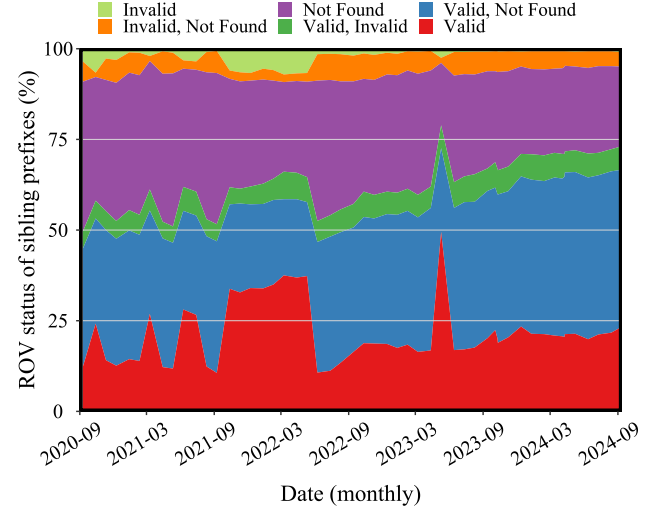


Figure 18: Route Origin Validation (ROV) status of sibling prefix pairs in the RPKI data over time.

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, 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 [24, 36, 38]. 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.

To summarize: 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.

4.8 RPKI Validity

The resource public key infrastructure (RPKI) can be used to check the legitimacy of a prefix announcement in BGP. Thus, we use RPKI data [50] 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 18 illustrates the route origin validation (ROV) 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 [41] and can help protect sibling prefixes from routing incidents such as BGP hijacks [8, 11, 48]. 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 (ROA) 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.

To summarize: 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 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. [5] 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 [6] 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. [19] 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. [55] 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. [1, 2] explored SNMPv3, 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. [35] introduced MIDAR, a tool leveraging the IP ID for alias resolution. In 2013, Luckie et al. [37] 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. [43] 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 [7], 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. [64] presented a tool called “Limited Ltd.” which exploits ICMP rate limiting for alias resolution.

Address assignment: In 2020, Padnamabhan et al. [44] 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 [53] 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. [29] 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. [39], 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.

6 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.

Choosing the right prefix size: 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 [47, 48] 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 [26]. 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.

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 [30, 38]. 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.

7 Conclusion

In this paper, 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.

Acknowledgements

We thank the anonymous reviewers for their valuable feedback, as well as our shepherd, Todd Arnold. We also thank the OpenIN-TEL team for providing the DNS data and the RIPE Atlas team for providing the probe data. Part of this work was carried out during Fariba’s internship at the National Institute of Informatics, Japan (NII).

References

- [1] Taha Albakour, Oliver Gasser, Robert Beverly, and Georgios Smaragdakis. 2021. Third time’s not a charm: Exploiting SNMPv3 for router fingerprinting. In *ACM Internet Measurement Conference (IMC)*.
- [2] Taha Albakour, Oliver Gasser, and Georgios Smaragdakis. 2023. Pushing Alias Resolution to the Limit. In *ACM Internet Measurement Conference (IMC)*.
- [3] Amazon.com. 2023. Amazon Alexa. <https://www.alexa.com/topsites>.
- [4] Shehar Bano, Philipp Richter, Mobin Javed, Srikanth Sundaresan, Zakir Durumeric, Steven J. Murdoch, Richard Mortier, and Vern Paxson. 2018. Scanning the Internet for Liveness. *ACM SIGCOMM Computer Communications Review (CCR)* (2018).
- [5] Arthur Berger, Nicholas Weaver, Robert Beverly, and Larry Campbell. 2013. Internet nameserver IPv4 and IPv6 address relationships. In *ACM Internet Measurement Conference (IMC)*.
- [6] Robert Beverly and Arthur Berger. 2015. Server siblings: Identifying shared IPv4/IPv6 infrastructure via active fingerprinting. In *Passive and Active Measurement (PAM)*.
- [7] Robert Beverly, William Brinkmeyer, Matthew Luckie, and Justin P Rohrer. 2013. IPv6 alias resolution via induced fragmentation. In *Passive and Active Measurement (PAM)*.
- [8] BGPmon. 2019. BGPmon Hijacking events. <https://www.bgpmon.net/category/hijack/>.
- [9] Timm Böttger, Felix Cuadrado, and Steve Uhlig. 2018. Looking for hypergiants in peeringDB. *ACM SIGCOMM Computer Communications Review (CCR)* (2018).
- [10] CAIDA. 2012. The CAIDA UCSD AS to Organization Mapping Dataset. https://catalog.caida.org/dataset/as_organizations/.
- [11] CAIDA. 2024. BGP Hijacks Observatory. <https://dev.hicube.caida.org/feeds/hijacks/events>.
- [12] Catchpoint Systems, Inc. 2025. Catchpoint - Internet Performance Monitoring. <https://www.catchpoint.com>
- [13] CDN Planet. 2024. Content delivery network. <https://www.cdnplanet.com/cdns/>.
- [14] Z. Chen, Z. Bischof, C. Testart, and A. Dainotti. 2024. AS to Organization Mapping. <https://github.com/InetIntel/Dataset-AS-to-Organization-Mapping>.
- [15] Zhiyi Chen, Zachary S Bischof, Cecilia Testart, and Alberto Dainotti. 2023. Improving the Inference of Sibling Autonomous Systems. In *Passive and Active Measurement (PAM)*.
- [16] Cisco. 2024. Umbrella Top 1M Lists. <https://umbrella.cisco.com/>.
- [17] Cloudflare. 2024. Cloudflare Radar Top List. <https://radar.cloudflare.com/>.
- [18] Jakub Czyz, Mark Allman, Jing Zhang, Scott Iekel-Johnson, Eric Osterweil, and Michael Bailey. 2014. Measuring IPv6 Adoption. *ACM SIGCOMM Computer Communications Review (CCR)* (2014).
- [19] Jakub Czyz, Matthew J. Luckie, Mark Allman, and Michael Bailey. 2016. Don’t Forget to Lock the Back Door! A Characterization of IPv6 Network Security Policy. In *Network and Distributed Systems Security (NDSS)*.
- [20] S. Deering and R. Hinden. 1995. Internet Protocol, Version 6 (IPv6) Specification. RFC 1883 (Proposed Standard).
- [21] David Dittich and Erin Kenneally. 2012. The Menlo Report: Ethical Principles Guiding Information and Communication Technology Research. *SSRN Electronic Journal* (2012).
- [22] Zakir Durumeric, David Adrian, Phillip Stephens, Eric Wustrow, and J. Alex Halderman. 2024. Ten Years of ZMap. In *ACM Internet Measurement Conference (IMC)*.
- [23] Zakir Durumeric, Eric Wustrow, and J. Alex Halderman. 2013. ZMap: Fast Internet-wide scanning and its security applications. In *USENIX Security Symposium*.
- [24] 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. 2021. The ties that un-bind: decoupling IP from web services and sockets for robust addressing agility at CDN-scale. In *Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*.
- [25] Sam Fletcher, Md Zahidul Islam, et al. 2018. Comparing sets of patterns with the Jaccard index. *Australasian Journal of Information Systems* (2018).
- [26] Oliver Gasser, Randy Bush, Massimo Candela, and Russ Housley. 2024. Publishing End-Site Prefix Lengths. <https://www.ietf.org/archive/id/draft-gasser-opsawg-prefix-lengths-02.html>.
- [27] Petros Gigis, Matt Calder, Lefteris Manassakis, George Nomikos, Vasileios Kotronis, Xenofontas Dimitropoulos, Ethan Katz-Bassett, and Georgios Smaragdakis. 2021. Seven years in the life of Hypergiants’ off-nets. In *ACM Internet Measurement Conference (IMC)*.
- [28] Google. 2024. Google IPv6 Statistics. <https://www.google.com/intl/en/ipv6/statistics.html>.
- [29] Amanda Hsu, Frank Li, and Paul Pearce. 2023. Fiat lux: Illuminating IPv6 apportionment with different datasets. *ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Science* (2023).

- [30] Geoff Huston. 2024. The IPv6 transition. <https://blog.apnic.net/2024/10/22/the-ipv6-transition/>.
- [31] IPinfo. 2024. Introducing IPinfo's probe network. <https://ipinfo.io/probe-network>.
- [32] Paul Jaccard. 1901. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bull Soc Vaudoise Sci Nat* (1901).
- [33] Joel Sommers. 2012. PyTricia. <https://github.com/jsommers/pytricia/tree/master>.
- [34] B. Kara and G. Simon. 2024. Power Efficient Multi-CDN Communication over Content Steering Server. In *Proceedings of the ACM Multimedia Systems Conference (MMSys)*.
- [35] Ken Keys, Young Hyun, Matthew Luckie, and Kim Claffy. 2012. Internet-scale IPv4 alias resolution with MIDAR. *IEEE/ACM Transactions on Networking (ToN)* (2012).
- [36] James Larisch, Timothy Alberdingk Thijm, Suleman Ahmad, Peter Wu, Tom Arnfeld, and Marwan Fayed. 2024. Topaz: Declarative and Verifiable Authoritative DNS at CDN-Scale. In *Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*.
- [37] Matthew Luckie, Robert Beverly, William Brinkmeyer, and kc claffy. 2013. Speedtrap: internet-scale IPv6 alias resolution. In *ACM Internet Measurement Conference (IMC)*.
- [38] Marek Majkowski. 2022. Cloudflare servers don't own IPs anymore – so how do they connect to the Internet? <https://blog.cloudflare.com/cloudflare-servers-dont-own-ips-anymore/>
- [39] Johannes Naab, Patrick Sattler, Jonas Jelten, Oliver Gasser, and Georg Carle. 2019. Prefix Top Lists: Gaining Insights with Prefixes from Domain-based Top Lists on DNS Deployment. In *ACM Internet Measurement Conference (IMC)*.
- [40] RIPE NCC. 2024. RIPE Atlas Probes. <https://atlas.ripe.net>.
- [41] NIST. 2024. NIST RPKI Monitor. <https://rpki-monitor.antd.nist.gov/>.
- [42] OpenINTEL. 2015. OpenINTEL Data Access. <https://openintel.nl/data-access/>.
- [43] Ramakrishna Padmanabhan, Zhihao Li, Dave Levin, and Neil Spring. 2015. UAV6: alias resolution in IPv6 using unused addresses. In *Passive and Active Measurement (PAM)*.
- [44] Ramakrishna Padmanabhan, John P Rula, Philipp Richter, Stephen D Strowes, and Alberto Dainotti. 2020. DynamIPs: Analyzing address assignment practices in IPv4 and IPv6. In *ACM CoNEXT*.
- [45] Craig Partridge and Mark Allman. 2016. Ethical Considerations in Network Measurement Papers. *Commun. ACM* (2016).
- [46] Potaroo. 2024. AS131072 IPv6 BGP Table Data. <https://bgp.potaroo.net/v6/as2.0/index.html>.
- [47] Lars Prehn, Franziska Lichtblau, and Anja Feldmann. 2020. When wells run dry: the 2020 IPv4 address market. In *ACM CoNEXT*.
- [48] Lancheng Qin, Dan Li, Ruifeng Li, and Kang Wang. 2022. Themis: Accelerating the Detection of Route Origin Hijacking by Distinguishing Legitimate and Illegitimate MOAS. In *USENIX Security Symposium*.
- [49] Philipp Richter, Mark Allman, Randy Bush, and Vern Paxson. 2015. A primer on IPv4 scarcity. *ACM SIGCOMM Computer Communications Review (CCR)* (2015).
- [50] RIPE NCC. 2024. RPKI snapshots. <https://ftp.ripe.net/rpki/>.
- [51] Ben Rosenblum and Nick Dunkin. 2024. The Multi-CDN Dilemma: Aggregated Edge Networks at Scale. In *SCTE TechExpo24*.
- [52] Route Views. 2024. RouteViews project. <https://www.routeviews.org/routeviews/index.php/collectors/>.
- [53] Erik Rye and Dave Levin. 2023. IPv6 Hitlists at Scale: Be Careful What You Wish For. In *Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*.
- [54] Patrick Sattler, Johannes Zirngibl, Mattijs Jonker, Oliver Gasser, Georg Carle, and Ralph Holz. 2023. Packed to the Brim: Investigating the Impact of Highly Responsive Prefixes on Internet-wide Measurement Campaigns. (2023).
- [55] Quirin Scheitle, Oliver Gasser, Minoo Rouhi, and Georg Carle. 2017. Large-scale classification of IPv6-IPv4 siblings with variable clock skew. In *Network Traffic Measurement and Analysis Conference (TMA)*.
- [56] D. Schinazi and T. Pauly. 2017. Happy Eyeballs Version 2: Better Connectivity Using Concurrency. RFC 8305 (Proposed Standard).
- [57] Khwaja Zubair Sediqi, Lars Prehn, and Oliver Gasser. 2022. Hyper-specific prefixes: gotta enjoy the little things in interdomain routing. *ACM SIGCOMM Computer Communications Review (CCR)* (2022).
- [58] Site24x7. 2006. <https://www.site24x7.com>.
- [59] Raffaele Sommese, Roland van Rijswijk-Deij, and Mattijs Jonker. 2024. This Is a Local Domain: On Amassing Country-Code Top-Level Domains from Public Data. *ACM SIGCOMM Computer Communications Review (CCR)* (2024).
- [60] Thorvald Sorensen. 1948. 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* (1948).
- [61] Florian Streibelt, Patrick Sattler, Franziska Lichtblau, Carlos H. Gañán, Anja Feldmann, Oliver Gasser, and Tobias Fiebig. 2023. How Ready is DNS for an IPv6-Only World?. In *Passive and Active Measurement (PAM)*.
- [62] Simpson Szymkiewicz. 1934. On the numerical characterization of closed populations. *Annales de la Société Polonaise de Mathématique* (1934).
- [63] TUM. 2024. ZMapv6 on GitHub. <https://github.com/tumi8/zmap>.
- [64] Kevin Vermeulen, Burim Ljuma, Vamsi Addanki, Matthieu Gouel, Olivier Fourmaux, Timur Friedman, and Reza Rejaie. 2020. Alias resolution based on icmp rate limiting. In *Passive and Active Measurement (PAM)*.
- [65] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczyński, and Wouter Joosen. 2019. Tranco List. <https://tranco-list.eu/>.
- [66] D. Wing and A. Yourtchenko. 2012. Happy Eyeballs: Success with Dual-Stack Hosts. RFC 6555 (Proposed Standard).
- [67] Maya Ziv, Liz Izhikevich, Kimberly Ruth, Katherine Izhikevich, and Zakir Dumeric. 2021. ASdb: a system for classifying owners of autonomous systems. In *ACM Internet Measurement Conference (IMC)*.

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 ← {DS_domains, IPv4_addresses}
tree_v6 ← {DS_domains, IPv6_addresses}
curr_jacc ← 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) ∈ sibling_prefix_pairs **do**

```

2:   tree_v4 ← sibling_prefix_v4
   tree_v6 ← sibling_prefix_v6
4:   new_jacc ← 0
   while new_jacc ≤ curr_jacc do
6:     prefix_v4_superprefixes ← GetNextSuperprefixes(sibling_prefix_v4)
     prefix_v6_superprefixes ← GetNextSuperprefixes(sibling_prefix_v6)
8:     for (prefix_v4, prefix_v6) ∈ {prefix_v4_superprefixes, prefix_v6_superprefixes} do
         new_jacc ← max(Jaccard(prefix_v4, prefix_v6))
10:    if (IsASNumChange(prefix_v4)) || (IsASNumChange(prefix_v6)) then
         return (prefix_v4, prefix_v6)
12:    else if (prefix_v4_len ≤ v4_len_thresh)
         || (prefix_v6_len ≤ v6_len_thresh) then
14:      return (prefix_v4, prefix_v6)

```

A Appendix

A.1 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.

The Figure 22 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 22. 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.

A.2 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 19. 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.

A.3 Detail Analysis of Hypergiants and CDNs of Sibling Prefixes

Figures 23 and 24 and Figure 25 shows a detailed information of distribution of Jaccard similarity values for sibling prefixes in HGs, CDNs, for the default case as observed in BGP, using /24-/48 thresholds, and the /28-/96 thresholds of SP-Tuner algorithm.

A.4 Detail Analysis of Business Type of Sibling Prefixes

As a detailed information on the business types of the origin ASes of sibling prefixes explained in Section 4.6, we consider the following two cases:

A.4.1 Origin AS of Sibling Prefixes for Business Type Analysis. 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 20 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 16 in Section 4.6 explained for the number of sibling prefixes. As it can be seen in Figure 20, 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 Section 4.6 still holds.

A.4.2 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 ASes and how much this affects the analysis we explained Section 4.6.

Figure 21 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 Section 4.5. Another significant difference in Figure 21 comparing to Figure 20 and Figure 16 in Section 4.6 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 21. 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

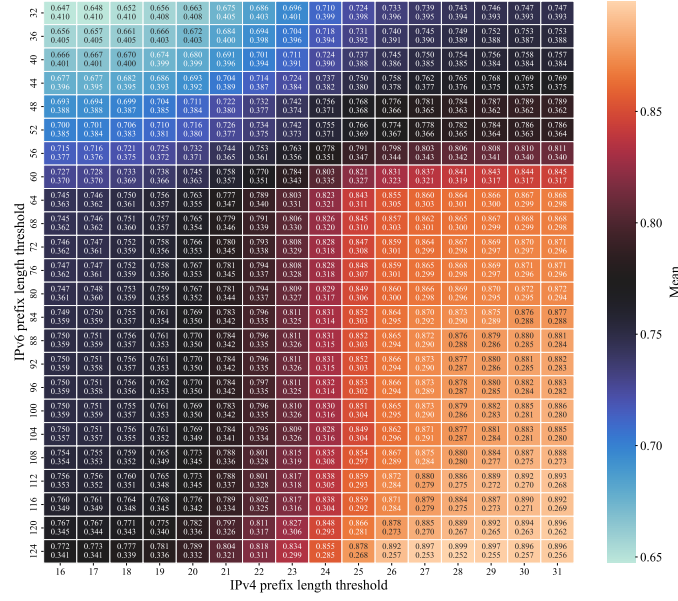


Figure 19: Heatmap of SP-Tuner Algorithm Performance: Mean Jaccard Index (top) and Standard Deviation (bottom) across IPv4 (x-axis) and IPv6 (y-axis) CIDR size thresholds.

previous analyses concerning the business type of the origin ASes for sibling prefixes.

A.5 Over Time Analysis of Sibling Prefixes

Figure 26 illustrates over time Jaccard similarity changes for sibling prefixes for the default case, Figure 27 for the SP-Tuner with a /24-/48 threshold case. On the other hand, Figure 28 illustrates the impact of the SP-Tuner algorithm with a /24-/48 threshold on Jaccard similarity of sibling prefixes for various data snapshots over time.

A.6 Origin ASes of Sibling Prefixes

Figures 29 and 30 depict the number of sibling prefixes over time for the same and different organizations in bars, the lines present

the number of unique IPv4 and IPv6 prefixes for the default case and the SP-Tuner case with /24-/48 thresholds. On a similar order of sibling prefixes for the default and SP-Tuner case, Figures 31 and 32 show the median Jaccard value changes over time for both cases.

A.7 Domains and CIDR sizes of Sibling Prefixes

In Figure 33 and Figure 34, we illustrate the percentage of sibling prefixes based on the number of domains for the default and the /24-/48 thresholds of SP-Tuner algorithm. Finally, we end the appendix of the paper by showing the subnet/CIDR sizes of sibling prefixes for the SP-Tuner algorithm with /24-/48 thresholds in Figure 35 and for /28-/96 thresholds in Figure 36.

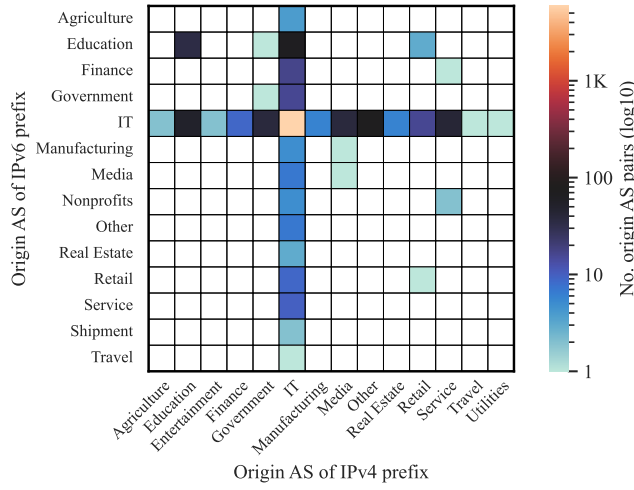


Figure 20: Business types heatmap for origin ASes of sibling prefixes, with IPv4 on x-axis and IPv6 on y-axis, indicating the count of origin AS pairs in each cell.

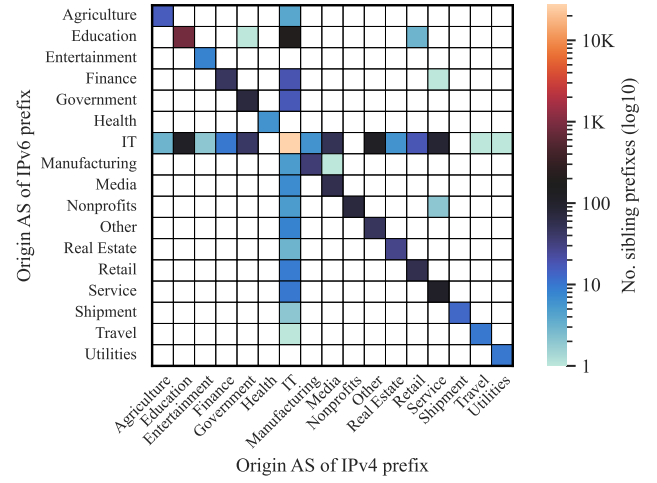


Figure 21: Business types heatmap for origin ASes of sibling prefixes, with IPv4 on x-axis and IPv6 on y-axis, indicating the count of sibling prefix pairs in each cell.

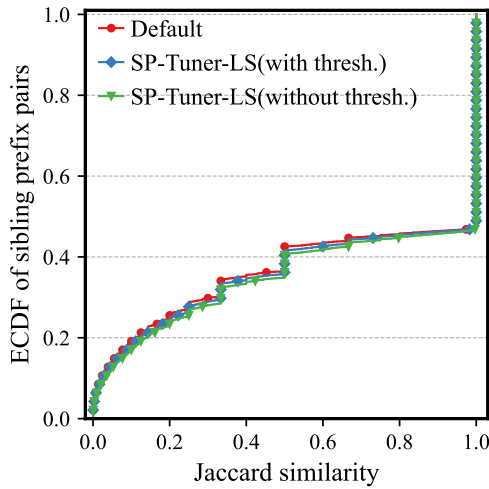


Figure 22: Jaccard similarity after applying SP-Tuner-LS on sibling prefix pairs data.

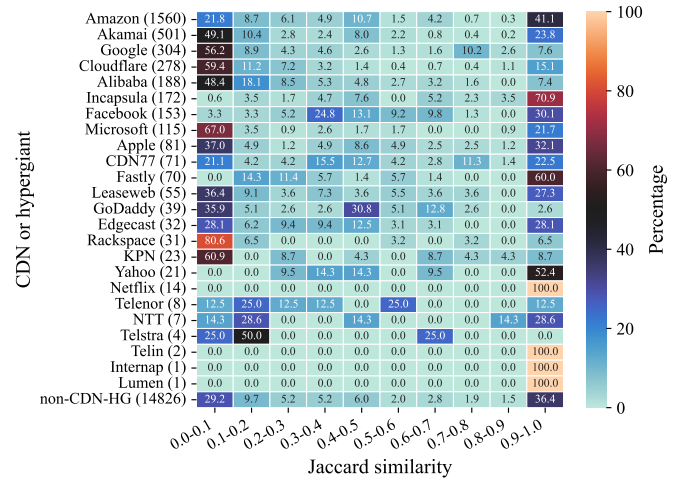


Figure 23: Heatmap of Jaccard similarity for the CDN, HGs and non-CDN-HG for the default case.

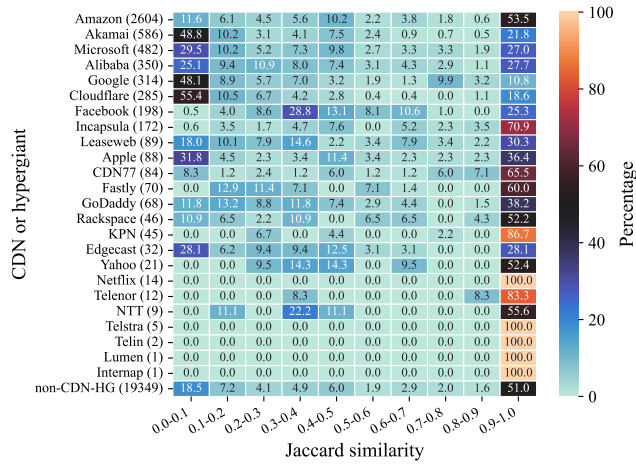


Figure 24: Heatmap of Jaccard similarity for the CDN, HGs and non-CDN-HG for the SP-Tuner thresholds of /24-/48.

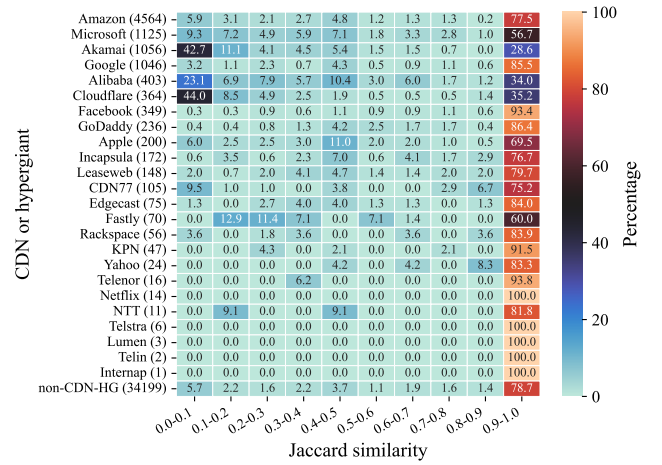


Figure 25: Heatmap of Jaccard similarity for the CDN, HGs and non-CDN-HG for the SP-Tuner thresholds of /28-/96.

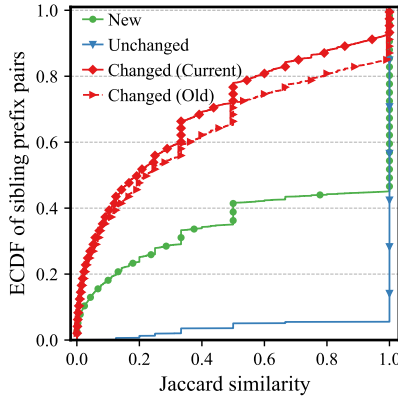


Figure 26: Changes of Jaccard similarity index value over time for the default case.

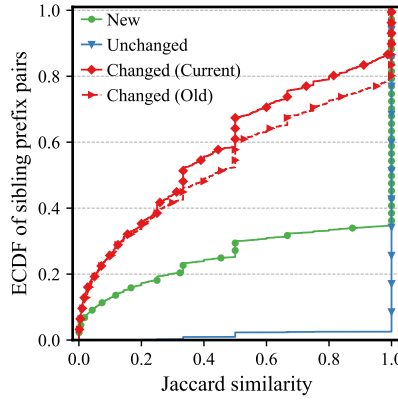


Figure 27: Changes of Jaccard similarity index value over time for SP-Tuner with /24-/48 thresholds.

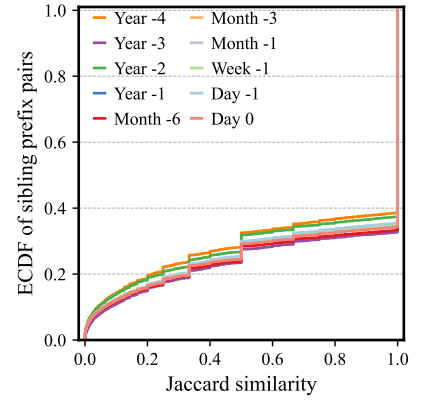


Figure 28: Impact of SP-Tuner with 24-48 thresholds on Jaccard similarity values on various snapshots over time.

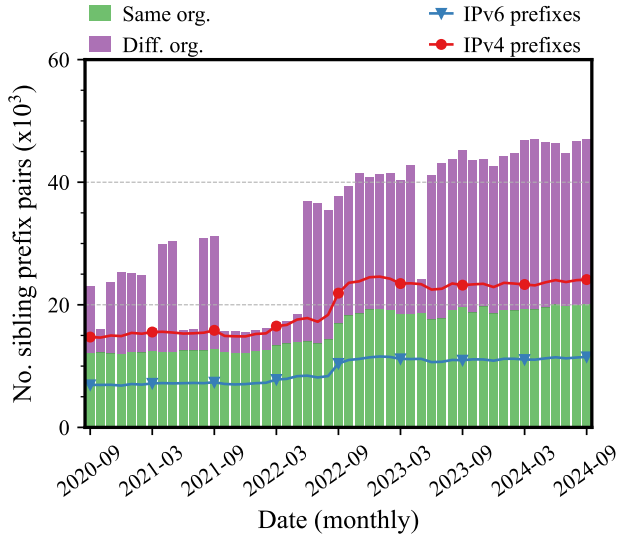


Figure 29: Sibling prefixes (bars) over time in same and different organizations for default case with lines showing the number of unique IPv4 and IPv6 prefixes over time.

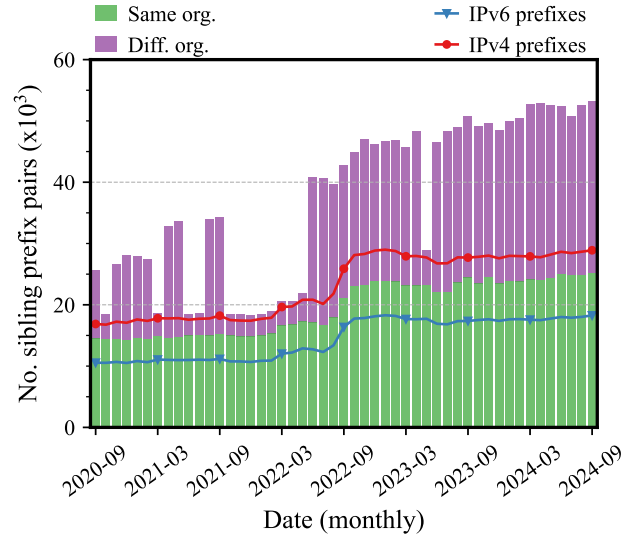


Figure 30: Sibling prefixes (bars) over time in same and different organizations for SP-Tuner with /24-/48 thresholds, lines showing the number of unique IPv4 and IPv6 prefixes over time.

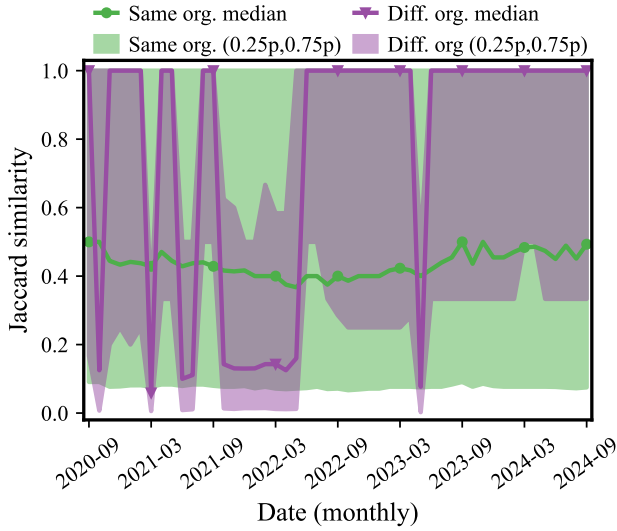


Figure 31: Median Jaccard for same and different organization over time for default case.

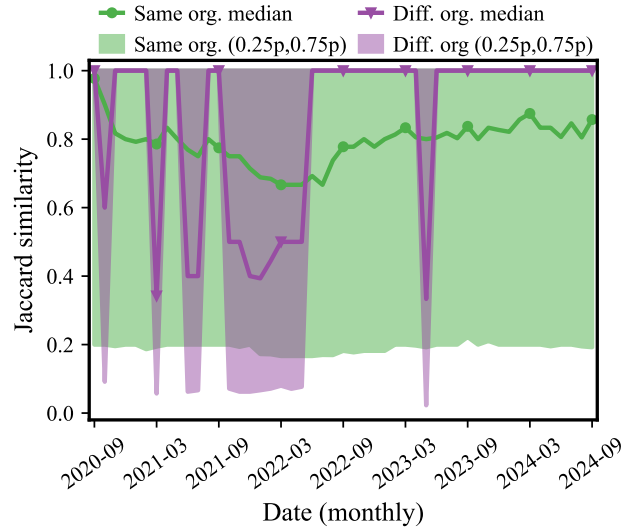


Figure 32: Median Jaccard for same and different organization over time for SP-Tuner with /24-/48 thresholds.

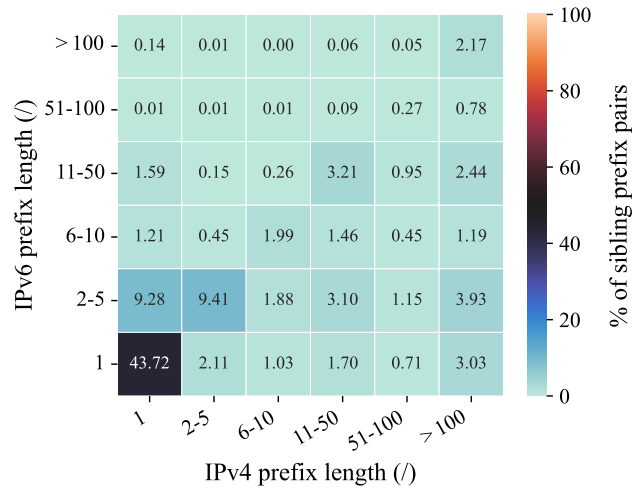


Figure 33: Heatmap of sibling prefixes % based on number of domains for default case.

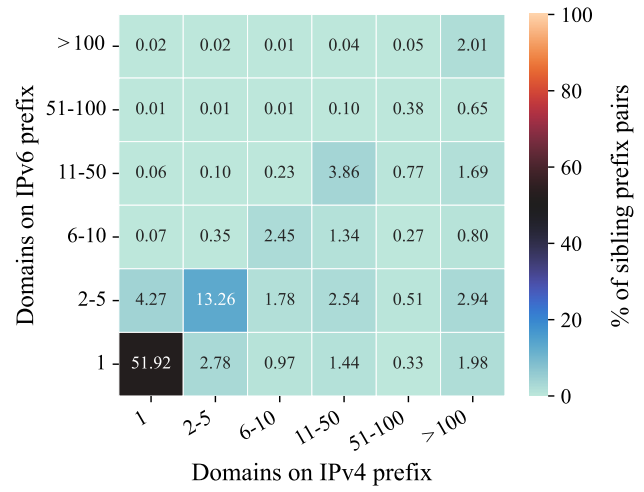


Figure 34: Heatmap of sibling prefixes % based on number of domains for SP-Tuner with 24-48 thresholds.

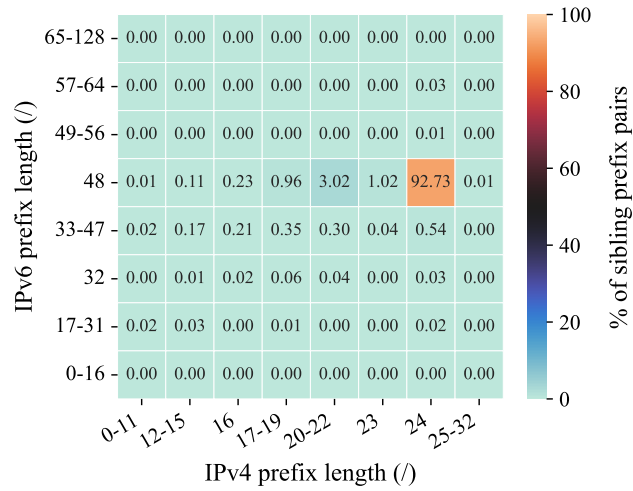


Figure 35: Heatmap of sibling prefixes CIDR size % for SP-Tuner with /24-/48 thresholds.

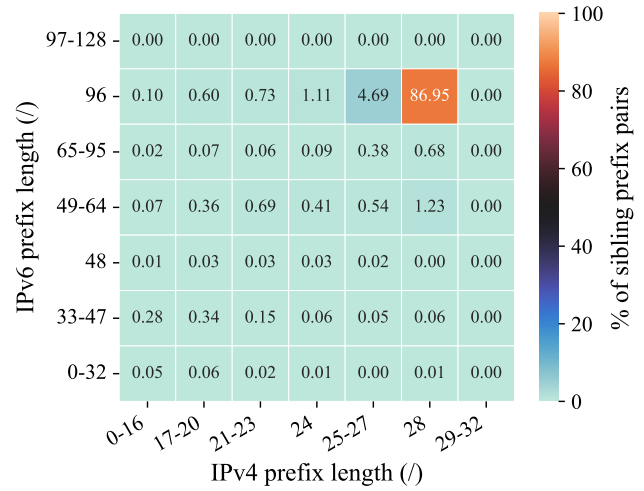


Figure 36: Heatmap of sibling prefixes CIDR size % for SP-Tuner with /28-/96 thresholds.