# Understanding IPv6 Aliases and Detection Methods

Mert Erdemir(✉), Frank Li, and Paul Pearce

Georgia Institute of Technology, Atlanta, GA, USA
{merterdemir,frankli,pearce}@gatech.edu

**Abstract.** Recent advancements in IPv6 address discovery methods provide new capabilities for Internet measurements. However, these measurement techniques are encountering a significant challenge unique to IPv6: large IPv6 prefixes that appear responsive on all addresses. The sheer sizes of these so-called IPv6 aliases preclude each responsive address as representing distinct devices; thus, these prefixes can confound measurements of IPv6 hosts. Although prior work proposed initial methods for identifying aliased regions, there has been limited characterization of IPv6 aliases and investigation into the resulting impact on the alias detection methods. In this work, we explore IPv6 aliasing in-depth, characterizing the properties of IPv6 aliases and exploring improvements to alias detection. We first analyze the state-of-the-art public IPv6 alias dataset, evaluating the accuracy and consistency of the alias resolutions. We uncover substantial misclassifications, motivating our development of a distinct high-confidence dataset of IPv6 aliases that enables us to correctly identify the distribution of aliased prefix sizes, detect real-world inconsistencies, and characterize the effects of different alias detection parameters. In addition, we show how small differences in the alias detection methods significantly impact address discovery (i.e., target generation algorithms). Our findings lay the foundation for how alias detection can be performed more effectively and accurately in the future.

## 1 Introduction

IPv6 adoption continues to grow, with more than 45% of Google's users connecting over IPv6 [33]. As the protocol landscape shifts, it is imperative that our measurement tools and methods adapt to the changing environment. Such adaption is challenging, though, as the nature of IPv6 combined with the vast address space size requires fundamentally different understanding and methods.

To perform Internet measurements, researchers must first identify active IPv6 hosts. Existing efforts have either harvested IPv6 addresses from Internet datasets (*e.g.* DNS), or developed generative approaches that predict which addresses are likely to be active. However, these approaches have encountered a problem unique to the vast size of the IPv6 address space: the existence of large prefixes that appear responsive on all addresses. These so-called *IPv6 aliases* (sometimes referred to as pseudo-dense regions) are too large to plausibly contain


C. Testart et al. (Eds.): PAM 2025, LNCS 15567, pp. 44–73, 2025.
https://doi.org/10.1007/978-3-031-85960-1_3

distinct hosts on each address. Although the aliasing is being utilized for various reasons, including, but not limited to, running multiple services on a single server, DDoS protection, load balancing, SYN proxies, or honeypots [23,29,48,62], it has confounded existing address discovery methods, driving them to produce numerous addresses within these regions that do not yield distinct hosts to fruitfully analyze. As we will show, **without accurately identifying and handling these aliased regions, Internet measurement results can become heavily skewed and paint an incorrect view of the IPv6 Internet.**

Despite the frequency with which prior IPv6 measurements have encountered aliased regions [7,17,36,37,44,46,51,54,64–66,76,77], there has been limited investigation into the characteristics of IPv6 aliases and the resulting impact on detection methods. Existing efforts have applied ad-hoc approaches for identifying and filtering aliased regions from their measurements, such as assuming the common prefix size of aliased subnets, probing randomly selected addresses within a prefix, and inferring aliasing if probed addresses are uniformly responsive [29,36,37,44,51,77,79]. These ad-hoc approaches have been used to produce the primary public dataset of detected aliased and non-aliased regions [29,30,79]. However, to date, we still lack a comprehensive evaluation of real-world alias properties, such as true alias sizes, and the influence of fundamental probing parameters within aliased subnets, such as probing frequency, the number of probes to use, or how many responsive probed addresses are needed (*i.e.* threshold).

In this work, we seek to understand the underlying properties of IPv6 aliases and how these characteristics inform alias detection methods. We begin by analyzing the accuracy of the state-of-the-art public dataset on aliased and non-aliased regions, exploring the extent to which the dataset correctly identifies aliases and their sizes. We identify substantial inaccuracies, that 85.3% of previously-identified aliased regions are mis-sized, reflecting both our current lack of deep understanding of IPv6 aliased regions and the ad-hoc detection methods applied. Motivated by our findings, we develop a high-confidence dataset of aliased Internet regions, which we then use to evaluate the true properties of IPv6 aliases and the impact of alias detection method parameters. Our analysis highlights and informs important design decisions with alias detection. In addition, we summarize the ethical considerations of our work in Appendix B.

Leveraging these insights, in a bulk dealiasing scenario, we find improved alias detection method parameters can reduce missed aliases (false negatives) up to 76.2%. When pairing this improvement with IPv6 Target Generation Algorithms (TGAs), we can reduce aliases in generated active targets by 80x, even when two dealiasing methodologies only differ slightly in detected aliases. We thus provide recommendations to enable more accurate and reliable alias detection methods in the future. Ultimately, our contributions include:

- Analyzing and characterizing the state-of-the-art public dataset of known aliased and dealiased IPv6 subnets, finding that 85.3% of aliases are mis-sized, and that more than 50% of aliased prefixes are less-specific than a /64 prefix granularity.

- Creating and characterizing a high-confidence dataset (HCD) of aliased prefixes for further research into IPv6 aliases.
- Showing that 98.7% of the aliased and 99.9% of the non-aliased subnets in the HCD remain the same for a long period (*i.e.* 3 months).
- Evaluating the selection of alias detection method parameters on our HCD, finding that improved parameters can reduce false negatives by 76.2%.
- Showing that requiring complete responsiveness to label a subnet as aliased leads to poor alias detection accuracy, which can be significantly improved by setting different threshold values.
- Showing that even a small number of aliases significantly impact TGAs, and deploying our recommended alias detection configuration can reduce aliases in generated active addresses by as much as 80x.
- Providing specific recommendations for future alias detection studies.

## 2 Background and Related Work

We begin with an overview of IPv6 and its aliasing phenomena, and prior work on detecting IPv6 aliases.

**Background on IPv6 and Aliases.** IPv6 addresses consist of 128 bits represented with 32 hexadecimal digits, each comprising 4 bits. These hexadecimal digits are referred to as *nybbles*. Due to the vast address space available in IPv6 (340 trillion trillion trillion addresses), address assignments to end sites are generous. RFCs on IPv6 allocation strategies suggest that allocation of the address spaces to networks should consider either /48 or /64 prefix sizes [13,56,59,60].

Given the abundance of addresses, a variety of novel address assignments have been utilized [73]. Perhaps the most prevalent and well-described addressing phenomena in IPv6 is *aliasing*. IPv6 aliasing occurs when large regions of contiguous IP address space are *fully responsive* to probes [29]. These regions, each of which may be larger than the entire IPv4 address space, are prohibitively large to reflect actual unique hosts. Instead, network devices or end hosts are configured to be responsive across entire ranges of addresses, presenting the appearance of full responsiveness.

This phenomenon can confound measurement as the basic act of counting hosts becomes challenging, leading to biased and incorrect results. In the context of generative IPv6 scanning [16,18,44,51,66,74,77], these regions present a fundamental challenge as, without intervention, algorithms will discover aliased regions as rich in active addresses. As a result, this reinforces further generation of addresses within the same regions, yielding millions of results which are, in actuality, non-distinct devices. Thus, developing methods to effectively identify aliases during IPv6 measurements is critical.

### 2.1 Related Work

We distinguish related work across three dimensions: IPv6 scanning, IPv6 address discovery, and IPv6 aliasing.

**IPv6 Scanning.** The introduction of fast Internet-wide scanning tools such as ZMap [22] enabled researchers to scan the entire IPv4 address space in a matter of hours. Unfortunately, given the exhaustive nature of these tools, they cannot be used to actively explore the IPv6 space. This gap gave rise to a series of methods aimed at producing generative IPv6 scanning tools. These Target Generation Algorithms (TGAs) input lists of known addresses and generate new addresses to explore [18,27,36,37,42,44,51,65,66,70,74,77]. A core challenge across all these works is aliasing, as it results in these tools over-generating addresses within aliased regions rather than in more meaningful networks to explore.

**IPv6 Address Discovery.** Besides TGAs, an alternative approach to IPv6 address discovery is collecting known IPv6 addresses from various sources. So-called *hitlists* [7,29,38,61,65,79] can be produced via both passive and active data sources [7,10,20,25,26,30,31,68,70]. Today, Gasser et al.'s *IPv6Hitlist* provides the largest public dataset of IPv6 addresses continuously sourced from various resources [2,3,9,21,45,47,52,57,67,71,78], and also resolves the aliased and non-aliased prefixes of those addresses [29,63,79]. The IPv6Hitlist is the primary dataset used for alias resolution by numerous studies [7,36,39,44,65,66,74,77,79]. In this work, we analyze this dataset in depth.

**Aliasing and Alias Detection in IPv6.** The problem of IPv6 aliasing is well-documented. Proposed alias detection methods have focused on fingerprinting the routers [6,46,50], probing techniques [49], unused addresses [54], protocol-specific features [6,72], delay sequences [69], analyzing application layer headers [1] and a combination of previously proposed methods [43]. However, recent developments in large-scale IPv6 scanning have created an increased demand for faster and less resource-intensive, probabilistic alias detection methodologies [17,37,38,44,51,65,66,74], which we focus on in this paper.

At the core of existing probabilistic methods is the assumption that within a large non-aliased prefix, the likelihood of an address being responsive, if selected uniformly at random, is exceedingly low. However, when exploiting this assumption, existing approaches make ad-hoc decisions on the prefix sizes to evaluate and the scan parameters. For example, Murdock et al. [51] only resolved aliases at the /96 and /112 prefix granularities, generating three random addresses for each prefix and sending 3 TCP SYN probes on port 80 to each address, classifying aliasing only when all addresses are responsive. Meanwhile, Gasser et al. [29,79] used a different alias detection method when resolving aliases for the IPv6Hitlist. For an evaluated prefix (*e.g.* dead:beef::/32), they generate 16 random addresses that cover all subprefixes in the next nybble (*e.g.* dead:beef:[0-f]000::/36) and send 3 probe packets on ICMPv6 and TCP/80 to each address. Prefixes where all addresses are responsive are considered aliased. We should note that we deploy this alias detection method on ICMPv6 for all the experiments in the paper since it ensures that the probes are evenly balanced across all next nybble sub-prefixes while still probing randomly selected targets. For prefixes more specific than a /64 (up to a /124), the IPv6Hitlist performs alias resolution only if 100 addresses are observed within that prefix. For /64 prefixes, a single observed address triggers alias detection. Prefixes less

specific than /64 are only considered if they are BGP announced. Due to these method decisions, aliases identified by these works are skewed towards specific prefix ranges, such as /64 s or /96 s.

Later, the IPv6Hitlist investigated aliased prefixes using detailed fingerprinting [29,79]. However, the analyses did not evaluate the effectiveness or the accuracy of the initial aliased detection method (and its parameters). Therefore, our work seeks to address this gap by incrementally characterizing IPv6 aliases in the wild, and identifying parameters for efficient, accurate alias identification.

## 3 Evaluating Public IPv6 Alias Data

Existing IPv6 alias datasets have largely originated as a by-product of active scanning [30,44,66] and attempts to generate hitlists [7,29,65,79]. As such, the methods used to identify aliases are ad-hoc and varied. For example, prior work focused on exploring aliases at specific granularities, such as /64 [79] and /96 [51] prefixes, rather than attempting to determine the actual aliased prefix size. Furthermore, the mechanism commonly used for alias detection—probing randomly selected addresses within a prefix and testing for complete responsiveness—has not been incrementally evaluated. Incorrectly classifying aliasing status and size can confound IPv6 measurement, leading to biased results.

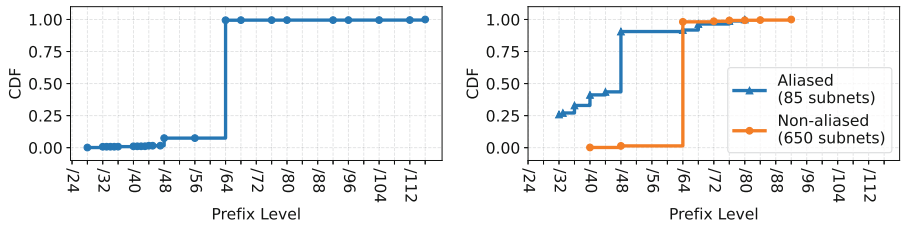
In this section, we seek to understand the correctness of the canonical existing IPv6 alias dataset, the IPv6Hitlist's Aliased Prefixes list [29,63]. We focus our efforts on understanding three aspects of the dataset: 1) prefix sizing (Sect. 3.1), 2) dataset inconsistencies (Sect. 3.2), and 3) the impact of scanning parameters (Sect. 3.3). We note that our exploration here is not a critique of the IPv6Hitlist, but rather an attempt to understand how commonly-used method parameters affect the correctness of alias detection.

We collected the Gasser et al. hitlist data on November 25, 2023, consisting of 61K non-overlapping aliased prefixes ranging in size from /28 s to /120 s. It also contained 90.7M non-aliased prefixes ranging from /16 s to /120 s.

### 3.1 Understanding and Evaluating Sizing

In this section, we will assess if the assumptions made by prior work reflect aliasing on the Internet, by both verifying the provided granularities and performing inconsistency checks on the datasets. We first examine how existing alias detection methods lead to incorrect sizing broadly; then, we build an understanding of what these results suggest in terms of the aliasing population.

**Scanning Setup.** We conducted all the experiments on a machine equipped with a 24-core AMD EPYC 7402P processor, Intel X550T 10GbE ethernet converged network adapter, and 256 GB RAM. We have a dedicated 1Gbit path to our local router (Juniper Networks MX304), with no stateful devices or filtering upstream, and the router's upstream is multiple 40Gbps links. We have conducted significant scanning activity on this network at significantly higher speeds than this study and observed no loss.




(a) The Aliased Subnets' Sizes (b) The Inconsistent Subnets' Sizes

**Fig. 1.** The prefix size distribution of (a) the aliased prefix sizes in the non-overlapping IPv6Hitlist's Aliased Prefixes dataset and (b) the inconsistent regions for both aliased and non-aliased regions in the IPv6Hitlist Datasets.

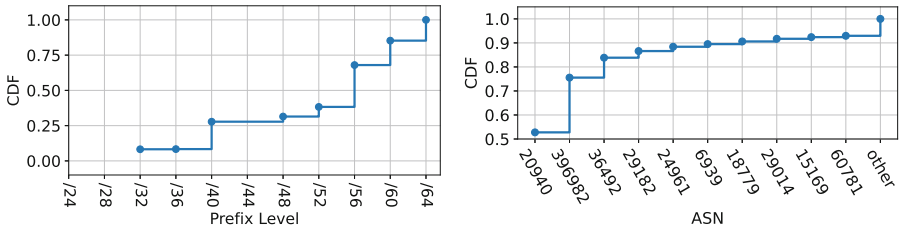
**IPv6Hitlist Prefix Size Distribution.** Figure,1a shows the distribution of prefix sizes in the IPv6Hitlist's Aliased Prefixes list. We should note that we do not consider more specific prefixes than /116 in our analysis since at /120 or more specific levels, the subnet space is small enough (256 addresses or less) where distinguishing aliased prefixes from dense but non-aliased subnets is probabilistically challenging unless scanning the entire region, which becomes unpractical in large scale settings. The dataset is dominated by /64 prefixes comprising 92% of all identified aliased prefixes. This distribution is unsurprising, as it stems from the methodology of Gasser et al. [29], whereby alias detection is triggered starting from /64 prefixes unless there are less specific BGP-announced prefixes. In other words, the IPv6Hitlist method does not proactively search for less specific aliased prefixes than /64. Even though these subnets are indeed fully responsive at a /64 granularity, it remains unclear if they are *actually* aliased at a /64 prefix; they could be aliased at less specific prefix levels.

**Inaccurate Alias Sizes.** To understand if the existing alias detection effort adequately identified alias sizes, we conduct an experiment where we identify the true alias prefix size of aliased /64 subnets in the IPv6Hitlist dataset.

Since /64 s dominate the dataset and are handled uniquely as the initial prefix size to trigger dealiasing, we begin by randomly sampling 1000 /64 aliased subnets. Although the 1000 sample size might sound small, it corresponds to almost 2% of the aliased /64 s and covers all the ASes in the dataset. Thus, we argue that it is sufficient to demonstrate the alias prefix size inaccuracies due to methodological choices. For each /64 subnet, we expand the prefix into the set of all possible less-specific prefixes at nybble intervals [29,79], starting from /24 s to /60 s. For /64 s that share common prefixes, we de-duplicate generated prefixes. We utilize the same alias detection methodology used by the IPv6Hitlist, as explained in Sect. 2.1.

We probe each of these addresses on ICMPv6 [66] using a purpose-built tool, shuffling the order of all addresses and prefixes probed. We send three back-to-back ICMPv6 Echo requests to each target address, and count a target responsive

if *any* probe packet results in a response, in an effort to account for packet loss. To address rate-limiting, we scan at an overall rate of 100 packets per second (pps), a rate three orders of magnitude lower than the default rate of ZMap [22]. To label a prefix as aliased, we require all 16 IPs to be responsive, aligning with prior methods [29,79].




(a) The True Alias Sizes (b) The 10 Most Common ASes

**Fig. 2.** (a) Distribution of the true alias sizes for the randomly sampled 1000 /64 s from the IPv6Hitlist. (b) The ten most frequent ASes of the mis-sized subnets in the sampled aliased /64 s. The y-axis is truncated at 0.5. Sampling is performed randomly, and the sample size is set to 2% of the aliased /64 s.

**Results.** Figure 2a shows the results of our experiment. In total, 849 of the tested /64 s were fully responsive to all tested IPs. We exclude four active but non-aliased, and 147 /64 s which were fully unresponsive at the time of scanning; we speculate such subnets are now offline due to IPv6 address churn. We find that broadly, most /64 s are misclassified; only 14.7% of purported /64 aliases are actually aliased at that granularity. Instead, the actual granularities of these aliased prefixes vary, with /56 s and /40 s being the most common, comprising 29.7% and 19.4% of our sampled aliased /64 s, respectively.

Table 1 shows the overall breakdown of the population of less-specific prefixes. We did not find any aliased /24 and /28 prefixes. Almost 82% and 72% of these subnets were fully unresponsive (*i.e.* inactive), respectively. Merging all true-sized aliased prefixes across all tested granularities resulted in 557 non-overlapping aliased subnets, in which /56 and /60 were the most common granularities, comprising 31.8% and 23.3% of the merged aliased prefixes.

**AS Distribution.** To further characterize the mis-sized aliased prefixes, we now examine their AS distribution. To map the prefixes to ASes, we use the December 31, 2023 snapshot of RIPE RIS collector rcc00 [53], which contains BGP routing data. Then, by using CAIDA's Inferred AS to Organization Mapping Dataset from December 31, 2023, we mapped ASes to organizations [11]. Figure 2b shows the ten most common ASes out of 45 distinct ASes. Nearly 85% of mis-sized aliased prefixes belong to three ASes, which comprise two logical ASes. Akamai (AS20940) accounts for 52.8% of incorrectly-sized subnets,

and 31.1% belong to Google (22.8% in AS396982 and 8.3% in AS36492). The remaining subnets correspond to 35 other ASes shown as *other* in the figure.

Overall, these results show that the vast majority of active /64 s believed to be aliased at a /64 granularity are actually aliased at less specific sizes, indicating a need for improved alias detection methods. Section 5 explores this question from the first principles across our high-confidence dataset.

### 3.2 Dataset Inconsistencies

The dynamic nature of alias detection can lead to different datasets disagreeing about whether a specific prefix is aliased or not. An inconsistency arises when a non-aliased region, such as a /96, appears to be present within a less-specific aliased region, such as a /32. This would be an unexpected behavior from an aliased region since such regions are anticipated to be fully responsive for *any* sub-prefix or address within them [29]. While this phenomenon has been observed [29] and discussed as SYN proxy behaviors [23], packet loss, or rate limiting, these inconsistencies have yet to be studied specifically.

Figure 1b explores inconsistencies in the IPv6Hitlist Aliased Prefixes dataset. We find that there are 85 aliased subnets, containing 650 non-aliased subnets at more specific granularities, indicating an inconsistency in alias labeling. We find most inconsistencies occur within /48 and /32 aliased subnets, and 96.8% of the inconsistent non-aliased subnets are /64 s. We note that the IPv6Hitlist's focus on /64 s means that it is unlikely to contain aliased /64 s with more-specific non-aliased subnets. These inconsistencies point directly to areas of exploration to understand methodology concerns. Of particular note is the notion that packet loss may generate these inconsistencies, which we explore further in Sect. 5.

### 3.3 Scanning Parameters

Since alias detection requires active scanning, understanding the effects of varying scanning parameters is critical to performing alias detection accurately. We examine three key parameters: 1) the number of probe packets to send per target, which can be thought of as retries for a given address, 2) how fast to scan overall, which influences the rate at which prefixes are probed, and 3) the order in which subnets are scanned (*e.g.* subnet-by-subnet vs generating all addresses to probe across all subnets, and then shuffling). We note that while some of these metrics may seem trivial, *e.g.* clearly, it is preferable to shuffle scanning across subnets, these restrictions become more challenging in the context of large-scale IPv6 Internet scanning. When conducting such scans, a natural formulation is to set an overall scanner rate-limit, and, when encountering an unknown prefix, to immediately evaluate that prefix directly [66,74]. Thus, understanding the precise effects of each of these scanning parameters influences the broad design of IPv6 scanning tools, in addition to alias detection methods.

**Experiments.** We begin to explore the effects of these parameters on the IPv6Hitlist Aliased Prefixes with coarse-grained experiments here, and we

**Table 1.** Breakdown of the true alias sizes for a random sample of 1000 /64 s (∼ 2%) from the IPv6Hitlist Aliased Prefixes dataset. We show the population of common, inactive and aliased subnets within [/24, /64] range, and the prefix size distribution of resulting aliased subnets.

| Prefix Size | /24 | /28 | /32 | /36 | /40 | /44 | /48 | /52 | /56 | /60 | /64 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Count | 87 | 91 | 126 | 231 | 306 | 377 | 475 | 518 | 786 | 935 | 1000 |
| Inactive | 81.6% | 71.4% | 69.1% | 72.3% | 73.9% | 69.8% | 57.3% | 29.0% | 17.7% | 14.2% | 14.7% |
| Aliased | 0 | 0 | 30.2% | 26.4% | 22.9% | 18.6% | 21.3% | 33.4% | 58.3% | 71.6% | 84.9% |
| % in Merged Aliases | 0% | 0 | 6.8% | 0.2% | 0.2% | 0 | 5.2% | 10.1% | 31.8% | 23.3% | 22.4% |

explore them in more depth based on our high-confidence data in Sect. 5. We experiment on the same set of randomly selected /64 s used earlier. We perform a set of experiments across three dimensions: 1) scan rate, testing 100pps and 1000pps, noting again that these scan rates are orders of magnitude below those of traditional scanning tools [22,28], 2) the number of probes sent per address, testing both a single probe versus three probes per address, and 3) packet order, trying both subnet-by-subnet or sequential, probing versus shuffling addresses to probe across the entire run. We argue that randomizing addresses across all runs is not feasible in large-scale IPv6 scanning. However, here, we focus on dealiasing as a separate process, rather than dealiasing during scanning.

In total, our evaluation comprises eight individual experiments. All addresses are scanned on ICMPv6, using the same alias detection method as the prior experiments. In order to classify a subnet as aliased, we require all 16 addresses to be responsive, based on the IPv6Hitlist method [29].

**Table 2.** Population of the misclassified subnets in the IPv6Hitlist scan parameter experiments. Misclassified subnets have at least one inactive IP address on ICMPv6.

| Experiments | Misclassified (100pps) | Misclassified (1000pps) |
|---|---|---|
| 1 probe, Randomized | 0.5% | 0.5% |
| 1 probe, Non-randomized | 1.6% | 1.1% |
| 3 probes, Randomized | 0.2% | 0.2% |
| 3 probes, Non-randomized | 1.1% | 1.0% |

**Results.** Figure 3 shows the percentage of subnets with active probed addresses as per the address scan order for all eight experiments. All experiments resulted in high response rates for active subnets, with none exhibiting lower than 98.5% responsiveness. Despite these high response rates, it should be noted that given that existing methods require *complete* responsiveness, all subnets that did not

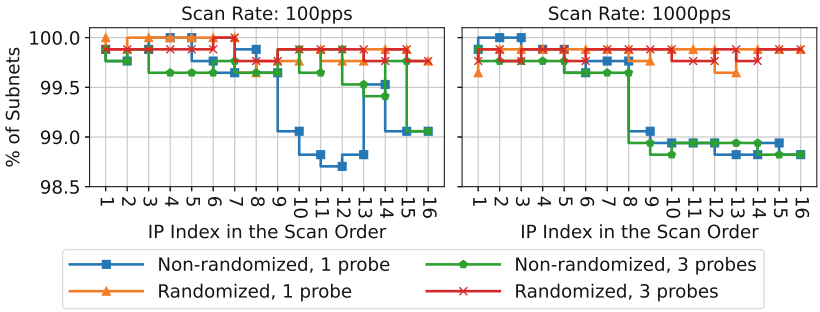



**Fig. 3.** Results of combining varying scanning parameters (*i.e.* the scan rate, the number of probes, the scan order) to measure their effects on the IP responsiveness in randomly sampled /64 aliased subnets over time. The y-axis is truncated to 98.5%.

yield 100% responsiveness (recall, these are aliased regions) result in incorrect classification as non-aliased. Table 2 shows how many subnets per experiment had at least one non-responsive address, which would result in a misclassification as non-aliased. We find the number of misclassified subnets would range from 0.1% to 1.6% of subnets across the eight experiments. As in Sect. 3.1, we excluded 151 inactive subnets from our analysis. We also note that although some of these misclassifications might stem from not performing TCP/80 probing, we aim to reveal the parameter effects in a simplified experimental setting.

**Effects of Shuffling.** Broadly, shuffling address probing across the entire experiment yields 54.5% and 68.7% fewer mislabeled subnets compared to the sequential probing, for 1000pps and 100pps, respectively.

**Effects of Number of Probe Packets.** Switching from 1 to 3 probe packets reduces the mislabeling by 60% when combined with shuffling, providing the best accuracy. Thus, we utilize a tweaked version of sending three probe packets during our high-confidence dataset (HCD) construction for the best accuracy.

**Effects of Scan Rate.** A slower scanning rate does not always yield better accuracy, especially when the addresses across all subnets are shuffled. During sequential probing experiments, we observe a 9.1% decrease in mislabeled subnets when we send three probe packets; one probe packet experiments result in a 31.2% decrease in mislabeled subnets, both when switching to 1000 pps rate from 100 pps rate. Manual investigation of the misclassified subnets in the slower scan rate experiments shows that the small differences mainly stem from the highly lossy networks, as also reported by Gasser et al. [29]. Thus, for larger experiment population sizes, utilizing faster rates with randomization can be more practical, as it maintains suitable accuracy. As a result, we use population-dependent scan rates in our HCD construction.

**Effects of Probe Order.** Address probe order matters, as the first address probed in a subnet is almost 85% less likely to be misidentified as unresponsive compared to the last address without randomization, especially at the faster

scan rates. These initial results show that combining different parameters (*i.e.* randomization and more probe packets) is crucial for better dealiasing accuracy, and point to the need for careful alias detection design and parameter selection.

## 4 Creating and Characterizing A High-Confidence Dataset

We now seek to understand the actual characteristics of IPv6 aliases in the wild, through creating a high-confidence dataset (HCD) of aliased subnets. This dataset affords the analysis of the *true* prevalence, size, composition, and distribution of aliases, which informs IPv6 measurements. Furthermore, we explore how different alias detection parameters influence alias classification (Sect. 5).

### 4.1 Data Collection

We begin by collecting domain names from the following sources utilized by prior work [7,29,79] (using snapshots between November and December 2023): all X.509 certificates found in Censys [21], the CAIDA DNS dataset [12], the Rapid7 Forward DNS dataset [58], and a collection of top lists [15,47,55,71,75]. For each of these data sources, we resolve all domains using ZDNS [41] against Google's public DNS resolver [32], querying for AAAA records. We then add to this dataset all IPv6 addresses from the IPv6Hitlist [29,63], CAIDA's Scamper [45], AddrMiner [65], and RIPE Atlas [14].

In total, we collect 284.5M unique IPv6 addresses. Given IPv6 churn [19,24], we then check if the collected addresses are still active via ICMPv6 scans [27] at the time of our dataset construction, which reduces our dataset to 98.5M responsive addresses. As building the HCD is a slow and resource-intensive task, and IPv6 aliases are dynamic over time [29], we construct the HCD off of a large random sample of all addresses (which is still representative of the full dataset).

We initially sampled 5M IPv6 addresses at random, but later identified that 78.4% of the sampled addresses belong to AS16509 (Amazon). We removed these addresses to avoid biasing our measurements towards the behavior of this one AS, as well as to avoid potentially heavily scanning a single AS. After filtering all Amazon IPs, no AS comprises the majority of our sampled dataset; thus, our final sample consists of 1.1M IPv6 addresses. The 3 most common ASes for the sampled addresses are AS47583 (Hostinger), AS12322 (Free SAS), and AS51468 (One.com), comprising 16.3%, 15.1%, and 7.9% of the sampled IPs, respectively.

Across these addresses, we calculate all distinct prefixes at the nybble granularity (every 4 bits), from /24 s to /116 s[1] (we do not explore beyond /116 s, as

[1] During construction, our HCD could encounter NAT64 /96 transition prefixes. We do not expect such prefixes to have a significant impact on our results, given: 1) Hsu et al. showed that there are very few publicly available NAT64 gateways [40], 2) such regions would either relay ICMP messages to the IPv4 Internet, thus having such prefixes take on sparsity [5] of the underlying IPv4 Internet, or would respond uniformly, thus being classified as either aliased or not aliased.

explained in Sect. 3.1). Our sampled addresses reside in a total of 11.9M unique subnets, of which prefix size breakdown is shown in Table 3.

**Table 3.** The number of unique subnets per prefix granularity in our HCD, calculated by computing common prefixes of the 1.1M input IPs, totaling up to 11,948,007 subnets.

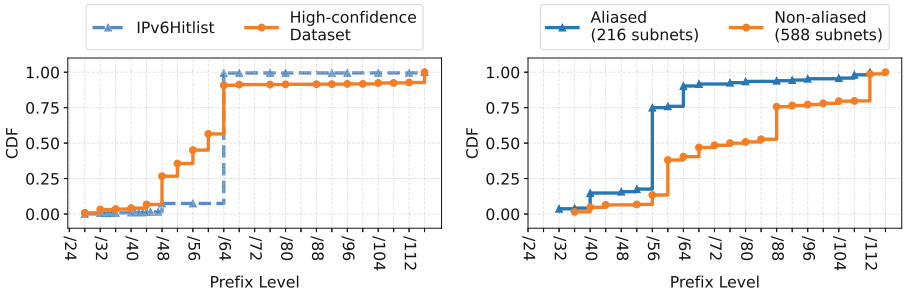
| Prefix | Count | Prefix | Count | Prefix | Count | Prefix | Count | Prefix | Count |
|---|---|---|---|---|---|---|---|---|---|
| /24 | 4,261 | /44 | 71,569 | /64 | 597,036 | /84 | 680,126 | /104 | 850,860 |
| /28 | 8,898 | /48 | 116,228 | /68 | 613,360 | /88 | 744,454 | /108 | 872,025 |
| /32 | 15,086 | /52 | 224,387 | /72 | 618,125 | /92 | 819,839 | /112 | 894,190 |
| /36 | 27,209 | /56 | 397,690 | /76 | 620,390 | /96 | 835,112 | /116 | 914,821 |
| /40 | 45,213 | /60 | 507,335 | /80 | 628,475 | /100 | 841,318 | | |

### 4.2 Method

To label all 11.9M subnets for the HCD, we run a three-round classification process. In the first two rounds, we probe and label subnets in ascending prefix size order, starting from /24 s up to /116 s (incrementing at the nybble granularity). The last round evaluates only non-aliased subnets that appear to cause aliasing inconsistencies (as defined in Sect. 3.2) based on the prior two rounds.

To mitigate the effects of rate-limiting, in all rounds, we follow a modified version of the dealiasing approach described in Sect. 3. Although rate-limiting is not an IPv6-specific problem, Vermeulen et al. showed that the rate-limiting in IPv6 is commonly triggered at scan rates even slower than 2Kpps, hinting at a high chance of subnet mislabeling when fast but practical scan rates are deployed [34,72]. Thus, specifically, for each prefix level, we pre-generate and randomly order all target addresses to probe, to avoid contiguous scanning of each subnet. We then round-robin probe all addresses across three iterations, thus avoiding back-to-back repeat probing of an address. Furthermore, we rate limit our probing of regions that we already identified as aliased (given that we evaluate from less specific to more specific prefixes). For example, if evaluating /48 s, we rate limit probing of any /48, that is within a /32 previously resolved as aliased, by the number of probes we send to this /32. We further explore the impact of these probing parameters on alias detection in Sects. 5.2 and 5.1.

We choose different scanning rates across prefix levels and rounds, depending on the number of addresses to probe. When there are fewer addresses, we scan slower to avoid rate limiting. Meanwhile, when there are more addresses, we can scan at a faster rate as the random shuffling of addresses more widely distributes the probing across subnets (and we still employ per-subnet rate limiting). Furthermore, by using multiple rounds, we can account for ambiguous classifications (where only a subset of the 16 probed addresses within a subnet are responsive) in one round, potentially due to some rate limiting or packet loss, by re-evaluating during a subsequent round.




(a) The HCD Aliased Subnets (b) The Inconsistent HCD subnets

**Fig. 4.** The prefix size distribution of (a) the non-overlapping HCD aliased subnets and the IPv6Hitlist, and (b) the non-overlapping inconsistent subnets in the HCD.

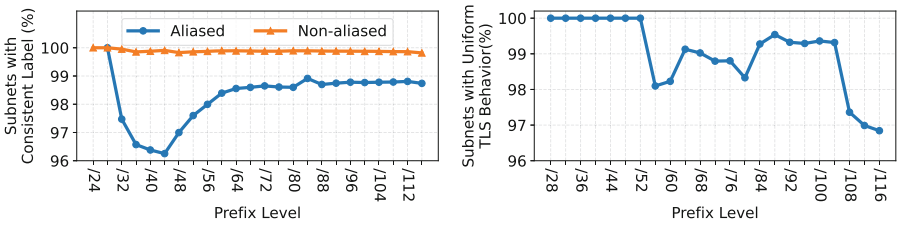
Concretely, in the first round, if there are $\leq 250K$ addresses for a prefix level, we scan at 100 pps with a 10 pps/subnet rate limit. If the address pool is in the range of $(250K, 1M]$ addresses, we scan at 1K pps with a 100 pps/subnet cap. For larger address pools, we scan at 10K pps (the default ZMap scan rate [22, 28]) while keeping a 100 pps/subnet cap. For the second round, we reduce the scan rates further, such that the subnet rate limit is always 10 pps/subnet, and when the address pool exceeds $250K$ addresses, we scan at 1K pps (instead of 10K pps). In the final round (which only evaluates non-aliased subnets exhibiting aliasing inconsistencies), we scan at only 10 pps, with a 1 pps/subnet cap.

For each round, we generate a new set of addresses per subnet (recall, we generate one address for all 16 sub-prefixes of the subnet). However, we classify based on merging across rounds, such that if each of the 16 sub-prefixes of the subnet is responsive in any round, we consider that sub-prefix as active for the sake of alias classification. We then classify a subnet as aliased if all 16 sub-prefixes are active (across any round). Finally, we merge aliased subnets within larger aliased subnets (e.g., an aliased /48 within an aliased /32) but preserve inconsistencies. Across the three rounds, we construct an HCD with 8,402 unique aliased subnets. In total, we classified 30% of all 11.9M subnets as aliased. We further perform a longitudinal analysis to understand aliasing behavior changes over time in Sect. 4.4, and utilize TLS fingerprinting to confirm identical behaviors of the aliased subnets' sub-prefixes in Sect. 4.5.

### 4.3 Prefix Level Distribution

Figure 4a shows the prefix size distribution of the 8,402 merged aliased subnets in our HCD. The figure also shows the prefix size distribution from the IPv6Hitlist Aliases dataset we presented in Fig. 1b as a blue dashed line.

The comparison paints a stark contrast. We find that 52.7% of the aliased subnets are less specific than /64, which accounts for 36.2% of the aliases. This contrasts with the prior work showing that aliases are significantly dominated by /64 s, comprising 92% of aliases from the IPv6Hitlist Aliased Prefixes.




(a) The Aliased and Non-aliased Subnets Maintaining their Label for 3 Months

(b) The Aliased Subnets Showing Uniform Behavior for TLS

**Fig. 5.** The population of (a) both the aliased and non-aliased subnets that have been classified the same over three months (see Sect. 4.4), (b) the aliased subnets showing a uniform behavior for ZGrab2 port 443 TLS scans for all of their 16 sub-prefixes (see Sect. 4.5). Note that the y-axis is truncated at 96%.

In addition, although /64 is the most common prefix size, our dataset shows that /48 s account for a significant 18.5% of the population followed by /60 s comprising 10.6% of the aliased prefixes. Moreover, as the datasets portray a different picture of IPv6 Aliasing, we further explore their similarities and differences in the labels they provide for the HCD input sample of 1.1M addresses in Appendix A. Even though the comparison shows that the labels across the two datasets are highly consistent (95.9%), the prefix sizes of the detected aliases for the identically labeled addresses are not. Table 5 shows that 4.4% of the aliased addresses are mapped to a more-specific granularity by the IPv6Hitlist, supporting the inaccurate alias size results in Sect. 3.1. We further investigate the impact of these small differences on the TGA generation in Sect. 5.5. These explorations further our motivation for improved alias detection methods beyond existing constructions, especially to account for dynamic prefix sizes.

### 4.4 Longitudinal Label Analysis

In order to understand whether the subnets remain aliased or not over time, we repeated HCD construction a total of 3 times, one month apart, totaling coverage of 3 months (*i.e.* 2024-04-25, 2024-05-30, and 2024-07-16).

Our analysis showed that 98.7% of the aliased subnets reported in Sect. 4.2 remained aliased in all construction instances. In contrast, only 0.1% of the non-aliased subnets changed their label to aliased in at least one of the later two HCD constructions. Moreover, we find that only 792 aliased (0.02%) and 606 non-aliased (0.01%) subnets exhibit a flip-flop behavior in their labels (*e.g.* having the opposite label only for the second iteration). The majority of this label-changing behavior happens in /116 granularity, accounting for 19.2% and 20.1% of the aliased and non-aliased flip-flop subnets, respectively.

Figure 5a shows the prefix-size breakdown of the subnets that stayed as aliased or non-aliased across three constructions. Although the non-aliased subnets remain the same across all prefix sizes compared to the aliased subnets,

we suspect this behavior originates from the majority of the non-aliased subnets being fully inactive. For the aliased subnets, the less-specific prefix sizes (except /28 s), especially in the [/32, /44] range, show slightly more variation across time compared to more-specific prefix sizes (although the vast majority of aliased subnets remain aliased over time). This variation in the [/32, /44] range mainly stems from significantly smaller subnet population sizes in the HCD, where a small number of non-uniform behavior instances appear as a larger portion among the tested subnets compared to other prefix sizes. Moreover, even though a tiny portion of the label changes happen in non-aliased to aliased direction (*i.e.* missing new aliases), suggesting that frequent HCD reconstruction might not be necessary, we further show how these small differences negatively impact the IPv6 address discovery in Sect. 5.5.

### 4.5 Label Validation with TLS Fingerprinting

Although the HCD construction utilizes a modified version of aliased detection deployed in prior work, we aim to gain more confidence in our HCD labels by investigating whether the aliases are indeed subnets behaving uniformly across IPs within it by fingerprinting application layer information. While doing so, we expect to see identical behaviors from the sub-prefixes of an aliased subnet, especially when probed for host-specific information, such as a TLS certificate.

We start our experiment by randomly sampling 1% of the aliased subnets that maintained the aliased label across three constructions, as reported in Sect. 4.4. Then, we pre-generate one address per 16 sub-prefixes for all sampled subnets. For each active address on port 443, we utilize ZGrab2 [21] to establish a TLS connection, if possible, and collect the TLS handshake logs. With this methodology, we aim to reveal the similarities in the sub-prefix behaviors for a subnet for a different protocol than ICMPv6. We categorize the behaviors as uniform and non-uniform. A subnet shows a uniform behavior if all the sub-prefixes: 1) share the same TLS certificate, 2) return an identical error (such as refusing the connection or canceling the request), or 3) are inactive on port 443. A non-uniform behavior arises when the responses of the sub-prefixes vary.

Figure 5b shows the prefix-size breakdown of what population of the aliased subnets showing uniform behavior. The aliased subnets commonly exhibit a uniform behavior across all prefix sizes, none comprising less than 96.8%. The results also indicate that groups of adjacent prefix sizes exhibit similar behaviors. For example, large regions such as prefixes from /28 s to /52 s unanimously show uniform behavior to TLS scans. Moreover, subnets within the prefix-size ranges of [/56, /80], [/84, /104], and [/108, /116] have similar non-uniform behavior populations. We argue that the high number of uniform TLS-scan behavior among all aliased subnets enhances confidence in the HCD.

**Limitations.** Since we consider some weak cases of uniform behavior (*e.g.* inactiveness), and only consider port 443, this validation alone does not robustly show that all these tested addresses are indeed single devices, or behave exactly the same; thus, it can possibly result in false positives. We suspect this approach

might be more successful in detecting these behaviors in CDNs than routers, resulting in an analysis of a limited population. Although confidence in our labels can be further improved by deploying other fingerprinting techniques [35, 72], we would like to emphasize that our main goal is to perform a rough demonstration of uniformity to give us more confidence in data quality.

### 4.6 Inconsistencies

Similarly to Sect. 3, we also explore inconsistencies in our HCD. After the second round of the HCD construction, we found 5,041 (588 non-overlapping) non-aliased subnets under 2,923 (216 non-overlapping) less-specific (*i.e.* larger) aliased subnets. Figure 4b shows the prefix distribution of the non-overlapping inconsistent subnets. Aliased subnets exhibiting inconsistencies are dominated by /56 s and /64 s, comprising 57.4% and 14.4% of the subnets, respectively. Moreover, 24.7% of the non-aliased subnets that cause inconsistencies appear at the /60 granularity, 23% are /80 s, and 19% are /112 s. We observe that 50.1% of the inconsistent non-aliased prefixes belong to AS47583, Hostinger. Similarly, 23.5% of the aliased subnets belong to the same AS, and 28.5% originate from 2 logical ASes (AS49392 and AS51659) of an organization, LLC Baxet.

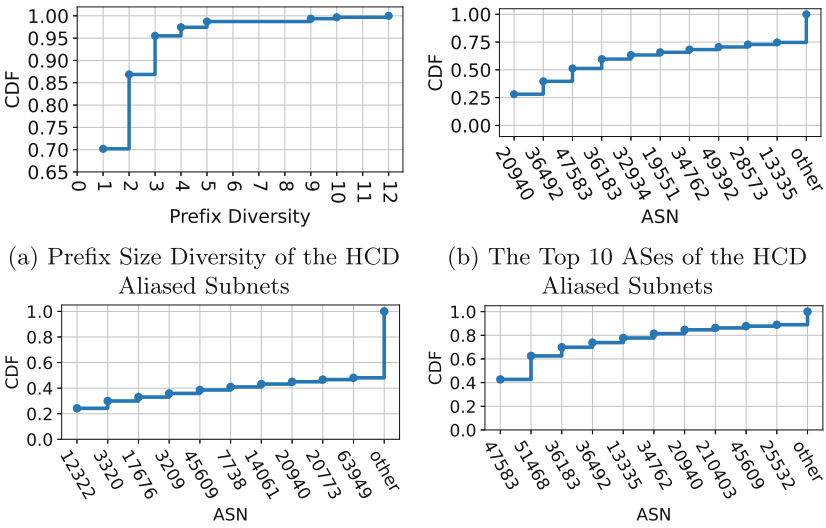
We found that 2.2% of these non-aliased subnets causing inconsistencies were indeed aliased, causing label changes in the HCD. In addition, 83.1% of these subnets were actually inactive, while their parent prefixes were active and aliased. Manual investigation indicates that the majority of these inconsistencies are legitimate inactive prefixes within larger aliased prefixes, but some exhibit stochastic inactive behavior explained in Sect. 5.1.

Although this experiment validates that aliasing inconsistencies exist in the wild and are not simply due to packet loss or network effects, further analysis of why this phenomenon happens in practice is left for future studies. We strongly encourage researchers to account for inconsistencies when performing measurements.

### 4.7 ASes with Aliases

Next, we characterize the aliased prefixes in our dataset by analyzing their ASes. We apply the same method as Sect. 3.1, mapping aliased subnets to ASNs using routing data, and then identifying each AS's organization by using the same datasets in Sect. 3.1. Figure 6b shows the distribution of the aliased subnets across ASes, depicting the 10 most common ASes. We see that 51.3% of the aliased subnets belong to only 3 ASes out of 312 total distinct ASes. Akamai (AS20940) accounts for 28.1% of these subnets; each Google (AS36492) and Hostinger (AS47583) account for 11.6%.

Figure 6a approaches this analysis from a different angle to understand how diverse the alias sizing is within these ASes. When we grouped the aliased prefixes by ASes, we observed that the ASes contain aliased subnets of varying (up to 12) prefix sizes. Most ASes (70.2%) had aliased subnets of a size, and 16.7% had aliased prefixes of 2 different sizes.




(a) Prefix Size Diversity of the HCD Aliased Subnets

(b) The Top 10 ASes of the HCD Aliased Subnets

(c) The Top 10 ASes for the Non-Aliased Input IPs

(d) The Top 10 ASes for the Aliased Input IPs

**Fig. 6.** Figure (a) shows the prefix size diversity within the ASes of the HCD Aliased Subnets, with the y-axis truncated at 0.65. The 10 most common ASes are shown for: (b) the HCD aliased subnets, (c) the non-aliased, and (d) the aliased input IPs.

We further classify the aliasing status of 1.1M input addresses, finding that 37.8% are in the aliased subnets. Figure 6c and Fig. 6d show the top 10 most common ASes for the non-aliased and aliased addresses, respectively. We see that 69.9% of the aliased addresses belong to 3 ASes: 42.7%, 19.8%, and 7.4% in Hostinger (AS47583), One.com (AS51486), and Akamai (AS36183), respectively. For non-aliased addresses, 24.2% were in Free SAS (AS12322), 5.7% in Deutsche Telekom (AS3320), and 3.1% in SoftBank Corp (AS17676).

## 5 Exploring Alias Detection Methods: HCD Experiments

In Sect. 4, we fully characterized our HCD and the aliased prefixes identified. We now investigate the effects of varying alias detection method parameters, leveraging our HCD to evaluate detection correctness. Since the alias detection methods entail active probing of addresses, the probing parameters can impact alias detection accuracy. Especially, because misinterpretation of prefixes negatively affects the aliased subnets more than non-aliased subnets, finding the right parameters and their values play a crucial role in IPv6 Aliasing studies. For example, if a non-aliased subnet gets labeled as non-aliased due to rate limiting or packet loss, the classification would still hold; however, for the aliased subnets, this would result in a misclassification, negatively impacting measurements.

Therefore, this section aims to understand what parameter values result in the most accurate and efficient large-scale bulk alias detection (*i.e.* performing alias detection on a given set of subnets at once). By building upon the lessons learned at each step, we produce guidance on how best to perform alias detection.

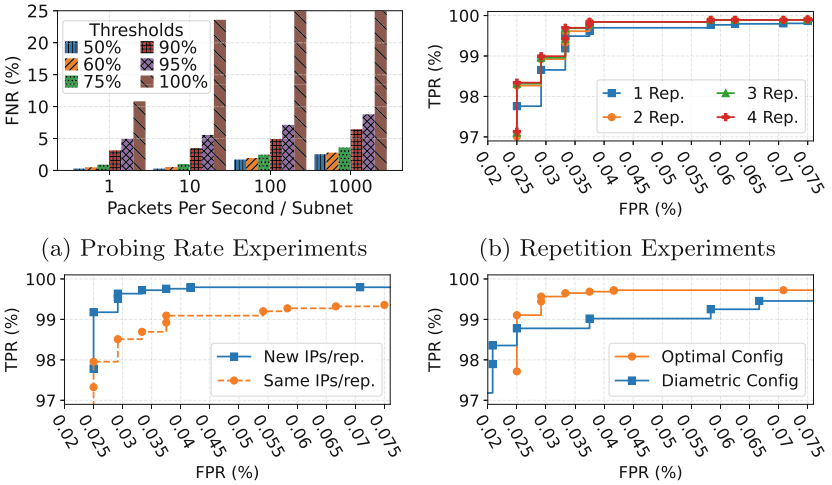
For all the experiments, we utilize the method of sending 16 probes proposed in the IPv6Hitlist [29] since it ensures the randomly generated targets are evenly distributed across all next nybble subprefix values. We think that this method takes the fragmented prefixes into account better compared to the method proposed by Murdock et al. [51]. For example, in a scenario where a non-aliased subnet neighbors only one aliased subnet in its next bit value, the random address generation without considering sub-prefixes could just sample from this fully active neighboring subnet, resulting in a misclassification despite the subnet actually being non-aliased.

We start by evaluating the rate at which we should probe the subnets, (Sect. 5.1). Then, we assess whether performing repeated probing improves the detection of aliased regions (Sect. 5.2). Next, Sect. 5.3 investigates whether probing new targets versus the same targets when performing repetitions improves classification accuracy. We then put these parameters together and compare alias detection results when using our recommended parameters against a measurement conducted using a diametric configuration in Sect. 5.4. Finally, in Sect. 5.5, we show the practical impact of improved alias resolution by evaluating how small differences in the alias detection methodologies can propagate to larger problems in IPv6 scanning outcomes.

**Subnets with Stochastic Network Behavior.** During our initial explorations, we observed high loss in some inconsistent aliased regions. A longitudinal activeness analysis at varying prefix granularities showed a stochastic behavior as these subnets change active status without a pattern. Also, collecting traceroutes during this dynamic behavior led to the same route. We think that these networks (*e.g.* AS49392) might be behind a DDoS protection or a proxy service since most packets are being dropped at the same last hop without a specific pattern. Although why we observe such behavior is unclear, it shows that alias detection is a highly time-sensitive task [29]. Thus, we filter these subnets from our aliased subnets list for the rest of the paper, resulting in 8,360 aliased prefixes. In addition, we exclude the aliased subnets that are no longer active during any of the experiments due to the churn in aliased networks.

### 5.1 Subnet Probing Rates

One of the most challenging problems in IPv6 alias detection is accounting for rate limiting [72]. Probing a network too fast could trigger rate limiting or cause packet loss (*e.g.* overprobing a middlebox on the path more than it can process), resulting in false negatives, which can significantly impact the IPv6 scanning measurements. Thus, in this section, we aim to evaluate the impact of probe rate on aliased subnets to find out what rates one can avoid rate limiting.




(a) Probing Rate Experiments

(b) Repetition Experiments

(c) Address Selection Experiments

(d) Configuration Experiments

**Fig. 7.** ROC Curves of False and True Positive rates, for 8,360 aliased and 24K non-aliased subnets (1K per prefix size, selected at random), (a) aliased subnet probing rate (with the y-axis truncated at 25%), (b) the repetition, (c) the address selection, and (d) the HCD optimal configuration vs a diagonal configuration experiments, with the y-axis (x-axis) truncated at 97%([0.02, 0.075]%).

We perform four experiments on the HCD aliased prefixes. For the experiments, we utilize the rate-limited subnet probing approach from Sect. 4.2 with 1 pps/subnet, 10 pps/subnet, 100 pps/subnet, and 1000 pps/subnet. We probe all the aliased subnets for a minute at each given rate, and continuously analyze their network behaviors. This methodology also enables us to determine when the rate limit is triggered in a wide range of 1–60K probes per minute.

Figure 7a shows the false negative rates at different threshold values for all four experiments. The threshold value determines the least number of responsive addresses required to label a subnet as aliased throughout an entire one-minute run (*e.g.* the threshold value of 100% represents full responsiveness). By analyzing different threshold values, we aim to understand if requiring complete responsiveness is accurate enough for aliased subnet classification.

**Complete Responsiveness.** Albeit commonly deployed in prior work [29,51,66], requiring complete responsiveness to label a subnet as aliased compared to when the threshold is set to 95% results in 2.2x higher FNR for 1 pps/subnet, and roughly 4.3x higher for all other scan rates. These initial results indicate that even only considering the scan-rate parameter, finding an optimal threshold value is as crucial as finding parameter values for higher accuracy.

**Scan Rates.** Scan rates faster than 10 pps/subnet lead to significantly higher FNR for all the threshold values. For example, although 1 pps/subnet and 10 pps/subnet have a 0.29% FNR at a 50% threshold setting, both 100 pps/subnet

and 1000 pps/subnet result in 5.9x and 8.75x higher FNR, respectively. Even though 1 pps/subnet outperforms all other experiments, especially for threshold configurations set ≤75%, utilizing 10 pps/subnet is more practical due to not only providing a similar FNR but also running 10x faster.

**Conclusion.** We argue that running at 10 pps/subnet rate provides the most optimal settings, especially when a stochastic network behavior is present. The results hint that choosing a threshold value around 75% rather than 100% reduces the FNs by more than 95%.

### 5.2 Repetition

In this experiment, we aim to explore whether repeated experiments improve aliased subnet labeling accuracy due to sending multiple probes accounting for the packet loss. We perform our evaluation over 8,360 HCD aliased subnets, and 1,000 non-aliased subnets selected at random per prefix level size between /24 and /116 at 4-bit intervals (*i.e.* 24,000 non-aliased prefixes in total). Testing non-aliased prefixes is crucial since small but densely active non-aliased regions might introduce false positives in repeated scans. Our non-aliased samples follow a similar AS distribution of the non-aliased input IPs (see Fig. 6c).

Next, we utilize the same rate-limited subnet probing strategy from Sect. 4.2 at 10 pps/subnet rate. However, this time, we repeat the same experiment 10 times, shuffling the scan order of the addresses each time. We label an address as active at repetition $R_x$ if it responds to at least one of the probes within $R_{\leq x}$ repetitions. Also, the threshold values represent the number of required responsive addresses per subnet across repetitions.

Figure 7b is a ROC curve showing how the true positive rates (TPR) and false positive rates (FPR) for varying repetition values (plotted as curves) at different alias labeling threshold values (plotted as data points on the curves). For clarity, we truncate the y-axis to show TPR above 97%, and the x-axis to show FPR above 0.025% and below 0.07%. We only show the first four repetitions since further repetitions do not result in increased TPR, but FPR.

**Different Threshold Values.** When no repetition is performed, both the threshold values of 9 and 10 result in the same 0.375% FPR, with the TPR of 99.69% and 99.62%, respectively. Although the TPR can be improved by setting lower threshold values, it also results in significantly higher FPR (*e.g.* 1.6x higher at threshold 8). When the threshold value is set to 9, performing 2–5 repetitions does not increase either TPR or FPR, converging them at 99.88% and 0.0375, respectively. The TPR improvements over more repetitions beyond 5 are minimal, rendering these parameter values more costly than more accurate.

However, after nine repetitions, we observe a 12% increase in FPR, remaining the same thereafter. Although there are only a few FPs present in the results, we manually investigate each prefix. 85.7% of the prefixes map to two ASes, AS209737 and AS61317. We observe the same stochastic behavior mentioned earlier on 57.2% of the prefixes. The rest shows fully responsive behavior two

weeks after the HCD construction, which we attribute to the dynamic nature of the aliases and the IPv6 churn.

**Conclusion.** We find that when repeating experiments, setting a lower threshold value results in significant improvements compared to complete responsiveness. Thus, performing two repetitions, while setting the alias labeling threshold to 9 gives the optimal results, while considering the probing cost. This is because performing more repetitions with smaller threshold values is more likely to increase the FPR, and provide smaller TPR gains, which do not compensate for the number of probes needed to be sent, and the longer experiment times.

### 5.3 Address Selection During Repetition

Here, we explore one final alias detection method parameter related to address selection during scan repetition. Previously, we re-scanned the same set of addresses across each scan repetition. However, upon each repetition, one could potentially probe a new set of randomly selected addresses per subnet. This approach largely precludes re-scanning the same addresses to confirm responsiveness (especially when prior probes are incorrectly inferred as inactive), but allows subsequent repetitions to evaluate the responsiveness of different addresses within subnets. Nevertheless, this might result in more FPs, especially at more specific granularities such as /116, due to increasing the likelihood of finding more active addresses in really dense regions.

To evaluate the impact of this parameter, we conduct the same experiment as in Sect. 5.2, and test with the same addresses versus new addresses for all subnets per scan repetition. During labeling, we check the activeness of the sub-prefixes, rather than individual addresses. We consider a sub-prefix active if at least one of the addresses under it is responsive across all repetitions. Figure 7c is a ROC curve showing the TPR and FPR under both address selection methods when the experiments were repeated two times. We do not present the results for one repetition as it is the first step, and only show the cumulative result at the end of the second repetition. We should also note that the same addresses curve represents the same 2 repetitions curve in Fig. 7b. Due to running experiments at different times, and filtering out separate sets of inactive subnets for both parameter experiments, the orange line is shifted to the bottom by less than 0.5% in Fig. 7c. The shift might sound significant at our TPR scales, but we find it expected since the new addresses approach is more likely to find active regions compared to the same addresses approach as explained above.

**Comparison.** We observe that the new addresses curve always stays above the same addresses curve, improving the TPR by 76.9% compared to using the same addresses, at the previously recommended threshold value of 9, when experiments are repeated two times. However, the new approach results in a higher FPR, 0.042%, whereas the same addresses approach has a 0.0375% FPR. Performing more repetitions with the new approach introduces two new FPs at the /116 granularity. This shows that for small but dense, non-aliased regions, generating new addresses per repetition is more likely to label them as aliased.

Interestingly, the same FPR can be achieved with the new addresses approach by setting the threshold value to 10, increasing the TPR with a 79.6% overall improvement.

**Conclusion.** We argue that generating new addresses per repetition significantly increases the TPR as its curve always stays above the same addresses approach. Although the previously suggested threshold value, 9, results in a 99.79% TPR, setting the threshold to 10 not only decreases FPR to 0.038% but also provides nearly the same TPR (*i.e.* 99.78%). Therefore, we recommend using the new addresses approach with the new adjusted threshold value of 10.

### 5.4 Optimal Alias Detection Parameters

Based on the presented experiments, we conclude that the optimal alias detection parameters are:

- Rate limiting individual subnets at 10 pps/subnet
- Performing two scan repetitions
- Setting alias labeling threshold to 10
- Generating new random addresses per subnet for each scan repetition

To demonstrate that our recommended dealiasing parameters result in significant improvements to the dealiasing accuracy, we experiment with our recommended configurations versus a diametric configuration: dealiasing without scan repetition in shuffled address scan order, sending 1 probe packet per address at 10 pps (chosen to minimize the potential impact of packet loss and rate limit).

Figure 7d is a ROC curve showing the TPR and FPR under both configurations. We observe that with two repetitions, our optimized configuration curve stays above the diametric configuration curve for all the threshold values. However, for thresholds 11–16, the FPR of the diametric configuration stays behind the optimal values, due to performing two repetitions and subnet-based rate-limiting. Noting the significant impact of TPR in IPv6 studies, we argue that this difference is affordable, considering that the optimal configuration reduces the misclassified aliases by 73.3% at the complete responsiveness threshold. For the recommended threshold value of 10 active subprefixes, the HCD configurations improve the accuracy by 76.2%. In addition, at this threshold, the diametric and the HCD configurations show different FPRs of 0.025% and 0.042%, respectively.

Therefore, we find that performing dealiasing based on the traditional experiment configurations at slow scan rates does not always yield better accuracy, while carefully selecting dealiasing parameters and considering different labeling threshold values result in a substantial increase in the dealiasing performance.

### 5.5 Impact of Aliasing Accuracy on TGAs

Modern IPv6 host discovery uses TGA models to generate *potentially* active targets. How a TGA generates these targets highly depends on what seed input

data is provided to the model. To prevent generation within aliased regions, seed addresses are typically dealiased before being inputted to TGAs, which creates a direct dependency between dealiasing method parameters and generated targets. Therefore, in this section, we aim to show how small amounts of misclassified addresses (*i.e.* 3.8% as shown in Appendix A) can heavily impact the TGAs, causing them to find a significant number (in some cases the majority) of the active addresses from the aliased regions.

In other words, we evaluate the effects of different IPv6 dealiasing method parameters on IPv6 host discovery by experimenting with state-of-the-art TGAs that are actively being used by the IPv6Hitlist to generate the Hitlist targets, and to trigger alias detection on the prefixes that are shared across multiple active targets [79]. Applying different seed dealiasing methods, we generate addresses with these models, and explore how many are aliased (labeled by the HCD). The models we deploy are: 1) IPv6Hitlist's version of 6Tree [44], 2) 6Graph [77], and 3) 6VecLM [18].

We start with filtering 0.3% of the 1.1M addresses, which we used in constructing the HCD, that cannot be labeled by the IPv6Hitlist dataset. Then, we use the remaining addresses to produce two dealiased datasets: 1) Non-aliased addresses as labeled by the IPv6Hitlist, 2) Non-aliased addresses as labeled by the HCD. Both datasets share at least 95.7% of their addresses. Finally, we ran each TGA model two times, each time inputting only one of the dealiased datasets. We follow the prior work's address generation approach, keeping all the parameters at their default values [79], and set the generation budget to 100M for both 6Graph and 6Tree.

Table 4 shows the number of generated and aliased addresses, the number of identical addresses generated for both the HCD and Hitlist dealiased inputs, and the HCD coverage for each model. We note that the number of generated addresses and the analyses exclude the seed input addresses of each dataset to quantify a model's actual generative performance. We observe that 9.8M, 21.6M, and 3.1K addresses that 6Tree, 6Graph, and 6VecLM generated are shared among the two dealiased datasets, respectively.

For all generated addresses, we perform an offline alias classification by using the HCD. Since the HCD coverage is limited with its input set, the majority of candidates cannot be mapped to a region in the HCD. However, we argue that the unlabeled addresses have a negligible impact on our overall analysis. First, considering that the HCD is constructed from randomly chosen addresses, we expect it to generalize to other addresses. Second, for all model-dataset pairs except the 6Graph-Hitlist pair, the number of HCD-covered addresses is larger than the experiment hit rates, indicating that the HCD coverage would be sufficient to characterize most active addresses. Thus, we limit our analysis to the candidate addresses that can be HCD-labeled.

Although the two dealiased datasets are slightly different, the models generated significantly more aliased targets when using the IPv6Hitlist, resulting in 83x and 19x more targets in responsive aliased regions for 6Graph and 6VecLM, respectively. Also, 6Tree using the HCD only produced 18 active aliased targets

**Table 4.** The number of the generated addresses for each model-dataset pair in the TGA experiments, the HCD coverage, the generated aliased address population, and the aliased active targets population. §: The number of generated addresses excludes the seed addresses. †: 45 targets are produced. ‡: 18 responsive targets are present.

| Model | # of Genenerated Targets§ | | | Targets Present in the HCD | | % of Aliased in Generated | | % of Aliased Hits in All Hits | |
|---|---|---|---|---|---|---|---|---|---|
| | HCD | Hitlist | Common | HCD | Hitlist | HCD | Hitlist | HCD | Hitlist |
| 6Graph [77] | 99.6M | 99.6M | 21.6M | 11.2% | 16.9% | 0.1% | 6.2% | **0.87%** | 72.61% |
| 6Tree [44] | 819.9M | 285.4M | 9.8M | 1.0% | 3.0% | 0%† | 0.2% | **0.0%**‡ | 32.57% |
| 6VecLM [18] | 53.5K | 57.3K | 3.1K | 31.5% | 40.2% | 0.4% | 6.8% | **2.13%** | 40.89% |

under a single subnet, whereas using the IPv6Hitlist generated 32.6% of active targets within aliases. Although the proportion of aliased targets to all generated addresses may seem small, these aliased addresses actually *do* have a drastic impact on IPv6 host discovery. As shown in Table 4, **between 32.6–72.6% of the active addresses found by the models are in aliased regions, when using the IPv6Hitlist for dealiasing seeds. In comparison, when dealiasing with the HCD, only 0.9–2.13% of active addresses are in aliases.** Thus, even a few inaccuracies in input dealiasing heavily impact the IPv6 host discovery process, motivating the need for accurate dealiasing methods.

## 6 Conclusion

IPv6 aliasing is a common yet challenging problem that confounds IPv6 Internet scanning and measurements. In this work, we found that existing methods detected aliases at largely incorrect granularities, and these inaccuracies are distributed across ASes. We uncovered that almost 53% of aliases exist at less specific levels than /64, unlike previously observed /64 dominance. Further, we confirmed that alias inconsistencies exist in practice, where aliased prefixes contain non-aliased subprefixes. These findings demonstrate the care with which IPv6 aliases must be handled.

To aid future measurements, we identified parameter recommendations for the alias method used in prior work (random probing of the 16 sub-prefixes of a subnet): probing subnets at 10 pps/subnet, repeat probing addresses twice while generating new addresses for each repetition, and using a 62.5% threshold for labeling a subnet as aliased. Compared to a diametric configuration, we showed that this configuration reduced missed aliases by 76.2%.

We also evaluated the impact of alias classification accuracy on TGAs, observing that even a few misclassified aliases in TGA input seeds can cause some TGAs to generate a majority (over 70%) of active addresses within aliases, compared to less than 1% of generated active addresses within aliases when the input aliases are correctly labeled. This stark result illustrates the importance of accurate alias detection.

Future work can expand upon our initial exploration of dealiasing parameters for different use cases, such as performing real-time alias detection. As online TGA models, like 6Sense, have been shown to be more effective, the need for an efficient and accurate real-time alias detection is growing [74]. Real-time alias detection requires novel aliasing strategies and parameters to be explored, like the optimal number of addresses to probe per subnet, and whether different configurations should be applied per prefix size and AS. These new methods can also be used to explore the services distinctively run by the aliased subnets. Ultimately, this study serves to lay the foundation for further improvements in IPv6 alias detection, in support of broader IPv6 measurements.

## A Overlap with the IPv6Hitlist

In order to understand the similarities and differences between the labels provided by the HCD and the IPv6Hitlist, we compare our HCD labeling of the input sample of 1.1M addresses to that from using the IPv6Hitlist Aliased Prefixes dataset. Even though 0.3% of the addresses were not in the IPv6Hitlist dataset, precluding comparison, roughly 95.9% of the addresses are labeled identically in both datasets, and 3.8% had conflicting labels in the IPv6Hitlist dataset. These contradictory labels consist of 1% of the non-aliased IPs, and 2.8% of the aliased IPs in the HCD. We further explore how this minor misidentification results in TGAs generating significantly more aliased addresses in Sect. 5.5.

**Table 5.** Comparison of aliased or non-aliased prefix sizes for IPs with identical aliasing labels in both the HCD and the IPv6Hitlist. †: The IPv6Hitlist dataset mapped two non-aliased IPs to a more specific subnet.

| | HCD Match Less Spec. | HCD Match Same | HCD Match More Spec. |
|---|---|---|---|
| Hit. & HCD Aliased | 4.4% | 95.5% | 0.1% |
| Hit. & HCD Non-Aliased | 0.0%† | 17.9% | 82.1% |

Although the alias labeling across the two datasets is highly consistent, the size of detected aliases is not, as shown in Table 5. We see that both datasets have the same prefix size for 79.4% of aliased addresses and 16.7% of non-aliased addresses. Interestingly, 4.4% of the aliased addresses are mapped to a less-specific prefix granularity in our HCD, indicating that the aliased subnet is a larger region than inferred by the IPv6Hitlist dataset; also, accounting for 98% of the size mismatches for the aliased addresses. Moreover, 83.2% of the non-aliased addresses are mapped to a more specific prefix size in our HCD (*i.e.* /116), hinting that future online alias detection methodologies might benefit

from starting to explore from more specific prefix sizes to label the addresses in non-aliased regions, resulting in more efficient measurements. However, we should note that one exception for such methodology is the inconsistent regions, which we explored as a phenomenon in Sect. 4.6.

## B Ethical Considerations

Due to the nature of our measurement work, we strongly recognize the importance of ethics in scanning studies. We uphold the ethical standards previously established in Belmont [8] and Menlo [4] reports. Probing of aliased subnets was done following the best practices of our community [22]. Namely, our scanning was conducted via a university network that had PTR records indicating the research nature of the machines. Each machine hosted an opt-out webpage. However, we have not received any requests to opt out during our study. Scans were randomized where appropriate, and all except full-Internet scans were rate-limited at no faster than 1K pps/subnet with a 10K pps cap. In addition, all data used in this study is from publicly available sources. No human subjects were involved in this research.

## References

1. Albakour, T., Gasser, O., Smaragdakis, G.: Pushing alias resolution to the limit. In: ACM Internet Measurement Conference (IMC) (2023)
2. Alexa: Alexa top 1 million (2021). http://s3.amazonaws.com/alexa-static/top-1m.csv.zip. Accessed 2021
3. Ark: Ark IPv6 Topology Dataset (2024). https://catalog.caida.org/dataset/ipv6_allpref_topology
4. Bailey, M., Dittrich, D., Kenneally, E., Maughan, D.: The Menlo report. IEEE Secur. & Privacy (2012)
5. Bano, S., et al.: Scanning the Internet for liveness. In: ACM SIGCOMM (2018)
6. Beverly, R., Brinkmeyer, W., Luckie, M., Rohrer, J.P.: IPv6 alias resolution via induced fragmentation. In: Passive and Active Measurement (PAM). Springer, Cham (2013)
7. Beverly, R., Durairajan, R., Plonka, D., Rohrer, J.P.: In the IP of the beholder: strategies for active IPv6 topology discovery. In: ACM Internet Measurement Conference (IMC) (2018)
8. The National Commission for the Protection of Human Subjects of Biomedical and Behavioral Research: The Belmont report - ethical principles and guidelines for the protection of human subjects of research (1979). https://www.hhs.gov/ohrp/regulations-and-policy/belmont-report/read-the-belmont-report/index.html
9. Bloomquist, Z.: TLDR 2 - a continuously updated historical TLD records archive (2024). https://github.com/flotwig/TLDR-2

10. Borgolte, K., Hao, S., Fiebig, T., Vigna, G.: Enumerating active IPv6 hosts for large-scale security scans via DNSSEC-signed reverse zones. In: IEEE Symposium on Security and Privacy (S&P). IEEE (2018)
11. CAIDA: Inferred as to organization mapping dataset (2024). https://www.caida.org/catalog/datasets/as-organizations/
12. CAIDA: The IPv6 DNS names dataset (2024). https://www.caida.org/catalog/datasets/ipv6_dnsnames_dataset/
13. Centre, R.N.C.: IPv6 address allocation and assignment policy (2020). https://www.ripe.net/publications/docs/ripe-738
14. Centre, R.N.C.: Ripe atlas (2024). https://www.ripe.net/analyse/raw-data-sets
15. Cloudflare: Cloudflare radar (2024). https://radar.cloudflare.com/domains
16. Cui, T., Gou, G., Xiong, G.: 6GCVAE: gated convolutional variational autoencoder for IPv6 target generation. In: Lauw, H.W., Wong, R.C.-W., Ntoulas, A., Lim, E.-P., Ng, S.-K., Pan, S.J. (eds.) PAKDD 2020. LNCS (LNAI), vol. 12084, pp. 609–622. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-47426-3_47
17. Cui, T., Gou, G., Xiong, G., Liu, C., Fu, P., Li, Z.: 6GAN: IPv6 multi-pattern target generation via generative adversarial nets with reinforcement learning. In: IEEE Conference on Computer Communications (INFOCOM). IEEE (2021)
18. Cui, T., Xiong, G., Gou, G., Shi, J., Xia, W.: 6vecLM: language modeling in vector space for IPv6 target generation. In: ECML PKDD 2020. Springer, Cham (2021)
19. Dhamdhere, A., Luckie, M., Huffaker, B., Claffy, K., Elmokashfi, A., Aben, E.: Measuring the deployment of IPv6: topology, routing and performance. In: ACM Internet Measurement Conference (IMC) (2012)
20. van Dijk, P.: Finding v6 hosts by efficiently mapping IP6.arpa (2012). https://7bits.nl/blog/posts/finding-v6-hosts-by-efficiently-mapping-ip6-arpa
21. Durumeric, Z., Adrian, D., Mirian, A., Bailey, M., Halderman, J.A.: A search engine backed by Internet-wide scanning. In: ACM SIGSAC Conference on Computer and Communications Security (CCS) (2015)
22. Durumeric, Z., Wustrow, E., Halderman, J.A.: ZMap: fast Internet-wide scanning and its security applications. In: USENIX Security Symposium (2013)
23. Eddy, W.: TCP SYN Flooding Attacks and Common Mitigations. RFC 4987 (2007). https://www.rfc-editor.org/info/rfc4987
24. Elmokashfi, A., Dhamdhere, A.: Revisiting BGP churn growth. ACM SIGCOMM (2014)
25. Fiebig, T., Borgolte, K., Hao, S., Kruegel, C., Vigna, G.: Something from nothing (there): collecting global ipv6 datasets from DNS. In: Passive and Active Measurement (PAM). Springer, Cham (2017)
26. Fiebig, T., Borgolte, K., Hao, S., Kruegel, C., Vigna, G., Feldmann, A.: In rDNS we trust: revisiting a common data-source's reliability. In: Passive and Active Measurement (PAM). Springer, Cham (2018)
27. Foremski, P., Plonka, D., Berger, A.: Entropy/IP: uncovering structure in IPv6 addresses. In: ACM Internet Measurement Conference (IMC) (2016)
28. Gasser, O.: ZMapv6: Internet scanner with ipv6 capabilities (2024). https://github.com/tumi8/zmap
29. Gasser, O., et al.: Clusters in the expanse: understanding and unbiasing ipv6 hitlists. In: ACM Internet Measurement Conference (IMC) (2018)
30. Gasser, O., Scheitle, Q., Gebhard, S., Carle, G.: Scanning the IPv6 Internet: towards a comprehensive hitlist. In: International Workshop on Traffic Monitoring and Analysis (TMA) (2016)
31. Gont, F., Chown, T.: Network Reconnaissance in IPv6 Networks. RFC 7707 (2016). https://www.rfc-editor.org/info/rfc7707

32. Google: Google public DNS (2024). https://dns.google/
33. Google: IPv6 statistics (2024). https://www.google.com/intl/en/ipv6/statistics.html
34. Guo, H., Heidemann, J.: Detecting ICMP rate limiting in the Internet. In: Passive and Active Measurement (PAM). Springer, Cham (2018)
35. Holzbauer, F., Maier, M., Ullrich, J.: Destination reachable: what ICMPv6 error messages reveal about their sources. In: ACM Internet Measurement Conference (IMC) (2024)
36. Hou, B., Cai, Z., Wu, K., Su, J., Xiong, Y.: 6Hit: a reinforcement learning-based approach to target generation for Internet-wide IPv6 scanning. In: IEEE Conference on Computer Communications (INFOCOM). IEEE (2021)
37. Hou, B., Cai, Z., Wu, K., Yang, T., Zhou, T.: 6scan: a high-efficiency dynamic Internet-wide IPv6 scanner with regional encoding. IEEE/ACM Trans. Netw. (2023)
38. Hou, B., Cai, Z., Wu, K., Yang, T., Zhou, T.: Search in the expanse: towards active and global IPv6 hitlists. In: IEEE Conference on Computer Communications (INFOCOM) (2023)
39. Hsu, A., Li, F., Pearce, P.: Fiat lux: illuminating IPv6 apportionment with different datasets. In: ACM on Measurement and Analysis of Computing Systems (POMACS) (2023)
40. Hsu, A., Li, F., Pearce, P., Gasser, O.: A first look at nat64 deployment in-the-wild. In: Passive and Active Network Measurement (PAM) (2024)
41. Izhikevich, L., et al.: ZDNS: a fast DNS toolkit for Internet measurement. In: ACM Internet Measurement Conference (IMC) (2022)
42. Li, X., Liu, B., Zheng, X., Duan, H., Li, Q., Huang, Y.: Fast IPv6 network periphery discovery and security implications. In: IEEE/IFIP Dependable Systems and Networks (DSN) (2021)
43. Liu, M., et al.: FBAR: an effective method for resolving large-scale IPv6 aliases. Int. J. Commun. Syst. **36**(18) (2023)
44. Liu, Z., Xiong, Y., Liu, X., Xie, W., Zhu, P.: 6Tree: efficient dynamic discovery of active addresses in the IPv6 address space. Comput. Netw. (2019)
45. Luckie, M.: Scamper: a scalable and extensible packet prober for active measurement of the Internet. In: ACM SIGCOMM Internet Measurement Conference (IMC) (2010)
46. Luckie, M., Beverly, R., Brinkmeyer, W., Claffy, K.: Speedtrap: Internet-scale IPv6 alias resolution. In: ACM Internet Measurement Conference (IMC) (2013)
47. Majestic: Majestic million (2024). https://majestic.com/reports/majestic-million
48. Majkowski, M.: Abusing Linux's firewall: the hack that allowed us to build Spectrum (2018). https://blog.cloudflare.com/how-we-built-spectrum/
49. Marchetta, P., Persico, V., Pescapé, A.: Pythia: yet another active probing technique for alias resolution. In: ACM Conference on Emerging Networking Experiments and Technologies (CoNEXT) (2013)
50. Marder, A.: Apple: alias pruning by path length estimation. In: Passive and Active Measurement (PAM). Springer, Cham (2020)
51. Murdock, A., Li, F., Bramsen, P., Durumeric, Z., Paxson, V.: 6Gen - target generation for Internet-wide IPv6 scanning. In: ACM Internet Measurement Conference (IMC) (2017)
52. NCC, R.: IPmap (2024). https://ipmap.ripe.net/
53. NCC, R.: Routing information service (RIS) (2024). https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris/

54. Padmanabhan, R., Li, Z., Levin, D., Spring, N.: UAV6: alias resolution in IPv6 using unused addresses. In: Passive and Active Measurement (PAM). Springer, Cham (2015)
55. Pochat, V.L., Van Goethem, T., Tajalizadehkhoob, S., Korczyński, M., Joosen, W.: Tranco: a research-oriented top sites ranking hardened against manipulation. In: Network and Distributed System Security Symposium (NDSS) (2018)
56. Popoviciu, C., Hahn, C., Bonness, O., de Velde, G.V., Chown, T.: IPv6 Unicast Address Assignment Considerations. RFC 5375 (2008). https://www.rfc-editor.org/info/rfc5375
57. Rapid7: Project sonar (2013). https://www.rapid7.com/research/project-sonar/
58. Rapid7: Rapid7 forward DNS (2023). https://opendata.rapid7.com/sonar.fdns_v2/
59. RIPE: Best current operational practice for operators: Ipv6 prefix assignment for end-users - persistent vs non-persistent, and what size to choose (2023). https://www.ripe.net/publications/docs/ripe-690
60. Roberts, R., Huston, G., Narten, D.T.: IPv6 Address Assignment to End Sites. RFC 6177 (2011). https://www.rfc-editor.org/info/rfc6177
61. Rye, E., Levin, D.: IPv6 hitlists at scale: be careful what you wish for. In: ACM SIGCOMM (2023)
62. Schindler, S., Schnor, B., Kiertscher, S., Scheffler, T., Zack, E.: Honeydv6: a low-interaction IPv6 honeypot. In: International Conference on Security and Cryptography (SECRYPT) (2013)
63. Service, I.H.: Understanding and unbiasing IPv6 hitlists (2024). https://ipv6hitlist.github.io/
64. Song, G., et al.: Towards the construction of global IPv6 hitlist and efficient probing of IPv6 address space. In: IEEE/ACM International Symposium on Quality of Service (IWQoS). IEEE (2020)
65. Song, G., et al.: AddrMiner: a comprehensive global active IPv6 address discovery system. In: USENIX Annual Technical Conference (USENIX ATC) (2022)
66. Song, G., et al.: DET: enabling efficient probing of IPv6 active addresses. IEEE/ACM Trans. Netw. (2022)
67. Spamhaus: The spamhaus project (2024). https://www.spamhaus.org
68. Strowes, S.D.: Bootstrapping active IPv6 measurement with IPv4 and public DNS. arXiv preprint arXiv:1710.08536 (2017)
69. Tao, Y., Hu, G., Hou, B., Cai, Z., Xia, J., Fong, C.C.: An alias resolution method based on delay sequence analysis. Comput. Mater. Continua **63**(3) (2020)
70. Ullrich, J., Kieseberg, P., Krombholz, K., Weippl, E.: On Reconnaissance with IPv6: a pattern-based scanning approach. In: International Conference on Availability, Reliability and Security. IEEE (2015)
71. Cisco umbrella popularity list (2024). http://s3-us-west-1.amazonaws.com/umbrella-static/index.html
72. Vermeulen, K., et al.: Alias resolution based on ICMP rate limiting. In: Passive and Active Measurement (PAM). Springer, Cham (2020)
73. Vyncke, E.: IPv6 over Social Networks. RFC 5514 (2009). https://www.rfc-editor.org/info/rfc5514
74. Williams, G., et al.: 6sense: Internet-wide ipv6 scanning and its security applications. In: USENIX Security Symposium (USENIX Security) (2024)
75. Xie, Q., et al.: Building an open, robust, and stable Voting-Based domain top list. In: USENIX Security Symposium (USENIX Security). USENIX Association (2022)

76. Yang, T., Cai, Z., Hou, B., Zhou, T.: 6forest: an ensemble learning-based approach to target generation for internet-wide ipv6 scanning. In: IEEE Conference on Computer Communications (INFOCOMM) (2022)
77. Yang, T., Hou, B., Cai, Z., Wu, K., Zhou, T., Wang, C.: 6graph: a graph-theoretic approach to address pattern mining for internet-wide ipv6 scanning. Comput. Netw. (2022)
78. Yeow, A.: Bitnodes API (2024). https://bitnodes.io/api/
79. Zirngibl, J., Steger, L., Sattler, P., Gasser, O., Carle, G.: Rusty clusters? Dusting an IPv6 research foundation. In: ACM Internet Measurement Conference (IMC) (2022)