Target Acquired? Evaluating Target Generation Algorithms for IPv6

Lion Steger*, Liming Kuang*, Johannes Zirngibl*, Georg Carle*, Oliver Gasser†
* Technical University of Munich, Germany
{steger, kuangl, zirngibl, carle}@net.in.tum.de
† Max Planck Institute for Informatics, Germany
oliver.gasser@mpi-inf.mpg.de

***Abstract*—Internet measurements are a crucial foundation of IPv6-related research. Due to the infeasibility of full address space scans for IPv6 however, those measurements rely on collections of reliably responsive, unbiased addresses, as provided *e.g.*, by the *IPv6 Hitlist* service. Although used for various use cases, the hitlist provides an unfiltered list of responsive addresses, the hosts behind which can come from a range of different networks and devices, such as web servers, customer-premises equipment (CPE) devices, and Internet infrastructure.**

In this paper, we demonstrate the importance of tailoring hitlists in accordance with the research goal in question. By using PeeringDB we classify hitlist addresses into six different network categories, uncovering that 42% of hitlist addresses are in ISP networks. Moreover, we show the different behavior of those addresses depending on their respective category, *e.g.*, ISP addresses exhibiting a relatively low lifetime. Furthermore, we analyze different Target Generation Algorithms (TGAs), which are used to increase the coverage of IPv6 measurements by generating new responsive targets for scans. We evaluate their performance under various conditions and find generated addresses to show vastly differing responsiveness levels for different TGAs.

I. Introduction

The adoption of IPv6 is continuously increasing, with on average 40% of all Google users connecting via IPv6 in March 2023 [1]. Due to the sheer size and sparse population of the IPv6 address space, exhaustive scans such as in IPv4 [2], [3] are infeasible in the IPv6 Internet. Therefore, Internet measurements targeting IPv6 hosts rely on up-to-date collections of responsive addresses, often known as *Hitlists*. Moreover, the success of these measurements heavily depends on the quality of their input, reliable targets, and high coverage of the active IPv6 Internet. While use cases for such hitlists can vary greatly, hitlists are usually a collection of addresses belonging to different types of devices, such as routers, web servers, or customer-premises equipment (CPE) devices, treated as a homogeneous set. This is very inefficient for many measurement studies, as these targets can be expected to be found in completely different network types. For example, a study on self-hosted video platforms would mainly target educational and company networks, while a study on web content will target vastly different networks, such as Content Delivery Networks (CDNs) and hosting providers. These studies could profit from a categorization of hitlist addresses, as this could allow more focused scans resulting in a reduced scanning overhead and lower load on the network.

978-3-903176-58-4 ©2023 IFIP

The most popular and commonly used *IPv6 Hitlist* by Gasser *et al.* [4], [5] combines IPv6 addresses from different sources and performs regular scans to ensure reliable responsiveness. However, little is known about the current and historic composition of the *IPv6 Hitlist*, namely which categories of addresses it contains and whether there is a bias towards CPE devices, routers, or servers. This makes the use of the hitlist unnecessarily difficult and inefficient for many measurement studies. We address this problem by analyzing the different network categories represented in the data provided by the hitlist service and showing how the categorization of the contained addresses improves the hitlists' usability.

In addition to hitlists, different approaches exist to increase IPv6 address coverage, *e.g.*, by generating new targets. This is often achieved through so-called Target Generation Algorithms (TGAs), which employ different methods such as machine learning [6], [7] and other pattern recognition techniques [8], [9]. Similar to hitlists, little is known about characteristics of TGAs with respect to input from different categories, whether they exbhibit biases towards specific address categories, or whether their results can be improved given more specific input. Therefore, existing TGAs could benefit from categorizing their input, enabling them to spend their algorithmic and scanning budget on application-tailored target generation.

In this paper, we perform an in-depth analysis of the *IPv6 Hitlist* as well as TGAs by categorizing IPv6 addresses. This research enables fellow researchers to make better use of the *IPv6 Hitlist* and TGAs. Our contributions in this work are:

1. ***IPv6 Hitlist* address categorization:** We analyze the *IPv6 Hitlist* by Gasser *et al.* with respect to IP address categories. We show that it includes addresses from a variety of categories, *e.g.*, Internet Service Provider (ISP) and Network Service Provider (NSP) in the input but also the set of responsive addresses, finding a general bias towards ISP networks with $42\,\%$ of responsive addresses.
2. **Characterization of address categories:** We evaluate whether addresses from differing categories exhibit different behavior over time. We show that addresses from educational and content serving networks are more stable with a median of over 200 days uptime, while ISP addresses are often only responsive during a single scan. ISP

Table I: List of target generation algorithms with publicly available code used in this work.

Year	Authors	Name	Scanning	Ref
2016	Foremski et al.	Entropy/IP	Static	[9]
2019	Liu et al.	6Tree	Dynamic	[11]
2020	Song et al.	DET	Dynamic	[12]
2020	Cui et al.	6GCVAE	Static	[6]
2021	Cui et al.	6VecLM	Static	[13]
2021	Cui et al.	6GAN	Static	[7]
2021	Hou et al.	6Hit	Dynamic	[14]
2022	Yang et al	6Graph	Static	[8]
2022	Yang et al	6Forest	Static	[15]
2023	Hou et al.	6Scan	Dynamic	[16]

and NSP addresses almost exclusively respond to ICMP, with less than $10\,\%$ response rate to any other protocol.

3. **Effectiveness analysis of TGAs:** We evaluate the effectiveness of different TGAs to identify previously unknown addresses. Furthermore, we analyze whether categorized input leads to a change in behavior for TGAs, finding stark contrasts in metrics such as number of generated and responsive addresses and responses to different protocols. For example, output generated from the ISP category has up to $50\,\%$ responsiveness, however almost exclusively to ICMP with below $10\,\%$ for any other protocol, whereas CDN addresses can generate $65\,\%$ responsiveness to HTTP.
4. **Data and Code:** We publish our adaptations to the used TGAs, generated and responsive addresses, analysis scripts and tools used throughout this work, as well as an ongoing categorization of the *IPv6 Hitlist* addresses [10]. In order for users of the *IPv6 Hitlist* to benefit from our findings, we update the service, include the newly discovered addresses and provide categorized statistics and data to the established service.

II. Background

We introduce TGAs, the *IPv6 Hitlist* service [4] and used data to categorize IP addresses.

A. Target Generation Algorithms

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

B. IPv6 Hitlist Service

The *IPv6 Hitlist* service from Gasser *et al.* [4] was started in 2018 and is maintained since. It collects IPv6 addresses from different sources and conducts scans for ICMP, TCP/80 (HTTP) and TCP/443 (HTTPS), UDP/53 (DNS) and UDP/443 (QUIC) on a regular basis. It was updated in 2022 by Zirngibl *et al.* [5] to improve the quality of the service. Their hitlist holds over 1.09 billion unique IPv6 addresses. Before scanning, they apply different filters, including a blocklist used to ensure opt-out possibilities for networks and ethical scanning, followed by a filter removing addresses which are known to receive bogus DNS injections falsely interpreted as responses. The injections are linked to the Great Firewall of China (GFW), which injects DNS messages regardless if the target host is responsive or not, introducing a strong bias towards DNS responses. Addresses which do not respond to any other protocol than DNS are therefore filtered. After this, another $360\,\mathrm{M}$ addresses are being marked as *aliased* and removed.

Gasser *et al.* [4] described aliased prefixes as subnets for which every contained address is mapped to and responded to by one single host, *e.g.*, through the `IP_FREEBIND` feature of Linux. Zirngibl *et al.* [5] showed that some of these prefixes are only fully responsive and used by multiple hosts. However, each of these prefixes (mostly /64) is infeasible to scan by itself and introduces massive biases of the hitlist. Therefore, the *IPv6 Hitlist* service runs a detection and filters the prefixes. The list of addresses after all filters is then scanned with probes for different protocols, of which $6.8\,\mathrm{M}$ were responsive to at least one protocol at March 4, 2023.

C. Categorization

PeeringDB, run by a community of network-operators, collects information about peerings and interconnections of networks around the world. Alongside this information, network operators can assign a category to their Autonomous System (AS) from twelve categories, including *Content* (Content Delivery Network, short CDN), *Cable/DLS/ISP* (Internet Service Provider, short ISP), *Educational/Research* (Universities, Research Institutes, short EDU), *NSP* (Network Service Provider, Transit networks) and *Non-Profit* (Non-Profit Organizations, short ORG), which are the five categories relevant in this work. Alongside PeeringDB, there was an AS classification by CAIDA, but it was discontinued in 2021 [17]. Furthermore, Ziv *et al.* [18] proposed ASdb in 2021, a system utilizing machine learning approaches to categorize networks at AS level with high accuracy. We did however not consider it in our project since their data only goes back to 2021 while the historic data from Gasser *et al.* [4] starts at 2018.

III. Related Work

The unpredictability of active addresses in the vast IPv6 address space leaves a lot of room for innovative discovery approaches, making the field of TGAs very interesting within

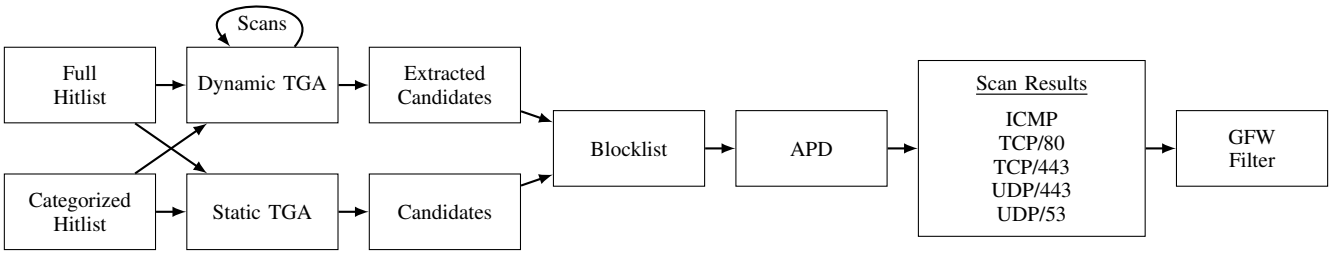

Figure 1: Pipeline to analyze TGAs (see Table I) and their performance within different IP address categories.

IPv6 research. Discovery strategies were already described in RFC7707 [19] based on drafts dating back to 2012. In 2015, Ullrich *et al.* [20] were among the first to publish on this topic. They propose an algorithm which iterates through different patterns of a training set, selecting sub-patterns with the highest amount of matching addresses. New addresses are generated from combining the undetermined bits of the patterns, outperforming the strategies laid out in RFC7707. In 2016, Foremski *et al.* [9] presented Entropy/IP, while Murdock *et al.* [21] presented 6Gen in 2017. The latter identifies dense regions in the input seeds and grows each input address into an independent cluster based on Hamming distance. All addresses which are inside the clusters and do not belong to the input seeds are regarded as candidate addresses. The authors claim that 6Gen outperforms Entropy/IP by a factor of 1-8 for identical input data sets. These early results already show large differences between TGAs and a detailed comparison including different input sets is required.

The remaining TGAs collected for this work (see Table I) follow similar approaches. They extract structural information from IPv6 seed sets and apply different methodologies to improve the quality of generated addresses. They report different response rates which are hardly comparable. In 2022, Zirngibl *et al.* [5] applied four TGAs during their improvement of the *IPv6 Hitlist*. They find that 6Graph and 6Tree generate the highest number of responsive addresses but do not evaluate algorithms in more detail and different input scenarios.

Rye *et al.* [22] took a slightly different approach when introducing *edgy* in 2020, focusing on the efficient discovery of the IPv6 periphery, *i.e.*, not servers or clients, but last hop routers. With *edgy* they were able to discover more than $64\,\mathrm{M}$ active last hop router addresses. One year later, Li *et al.* [23] describe a similar approach, discovering more than $50\,\mathrm{M}$ last hop router addresses through tracerouting non-existent IPv6 addresses in known or suspected customer subnets of ISPs. Lastly, Beverly *et al.* [24] presented their work focussing on IPv6 topology discovery. They develop and analyze strategies to collect new interfaces by efficient TTL-limited probing of target address sets, finding $1.3\,\mathrm{M}$ new router interfaces from their single vantage point.

IV. Data Sources and Target Generation

In the following, we describe our data sources and the approach to evaluate the collected TGAs introduced in Section II.

A. Data Sources

We use the complete historic data from the *IPv6 Hitlist* service [4] from July 1, 2018 until March 3, 2023. We analyze the historic data to gain insights into its categorical composition over time and the responsiveness and stability within each category. To map addresses to the AS announcing the respective prefix, we use historic Border Gateway Protocol (BGP) Route Views data [25] for one route collector from each scan date. These mappings to ASes are further used to identify the respective category based on historic PeeringDB data [26].

We use the full list of responsive addresses from the *IPv6 Hitlist* service from March 3, 2023 as input for TGAs in the following. Furthermore, we divide the input into filtered versions, *i.e.*, the addresses inside the active hitlist filtered based on the PeeringDB categories *Content*, *ISP*, *NSP*, *Non-Profit*, and *Educational*. While PeeringDB has more network categories, we excluded all categories with less than $5\,\%$ representation in the hitlist, additionally including the two categories *Educational* and *Non-Profit* to test the algorithms on smaller seed sets.

B. Target Generation Methodology

Figure 1 shows our pipeline to test TGAs on the different input files. In our study we run and evaluate the ten TGAs listed in Table I. The algorithms run on a machine with an NVIDIA GeForce RTX 2080 GPU, a 24-core Intel Xeon Silver 4214 CPU and $256\,\mathrm{GB}$ of RAM.

We run the static TGAs without any modifications apart from input files or output hyper-parameters. We modify the hyper-parameters number of epochs for 6GAN and the generation budget for 6GCVAE. To run algorithms in a feasible timeframe, we set the total number of epochs of 6GAN to ten and run 6VecLM with only the first of the predefined temperature hyper-parameters. 6GAN offers multiple modes for seed classification, of which we choose the *Entropy Clustering* method since the authors report the highest number of generated addresses for this method, which is the metric we optimize for. Since 6GAN and 6VecLM define the amount of input that is processed via hard-coded values and Entropy/IP is not intended for input greater than $100\,\mathrm{k}$ addresses, we randomize the input data set with a static, reproducible seed. Whenever possible we set the output budget to $10\,\mathrm{M}$, since we wanted to keep the size of the candidate sets at a similar scale to the input, *i.e.*, the hitlist. We implement the approach described by the authors of 6Graph to generate candidates from the dense regions identified by

their algorithm. For 6Forest, this process is not fully described or implemented. Therefore, we follow the same procedure as for 6Graph, generating distance-based targets and additionally generating full combinations if the number of wildcards, *i.e.*, free dimensions, is smaller than four.

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

After the successful generation of all candidate sets, we combine them into one target file. As a first step, the target file is stripped of duplicates and filtered by applying a blocklist, which we actively maintain in order to adhere to ethical scanning guidelines (see Section IV-C). Next, we conduct aliased prefix detection, as described by Gasser *et al.* [4] and additionally use the known aliased prefixes from the *IPv6 Hitlist* service as a filter. The remaining, non-aliased addresses are then used as input to the ZMapv6 port scanning tool[1]. We scan from a single vantage point in Munich, Germany, connected to the German National Research and Education Network. As a last step, the responses to the UDP/53 (DNS) probes are passed through a GFW filter, which we describe in Section V-C.

During the analysis stage, we take the following steps: First, all duplicates and overlaps with the respective seed set are removed for all candidate sets. Then the filtered candidate sets and filtered scan file are matched to identify the responsive portions of the candidate sets. This provides the final result for each TGA.

C. Ethical Considerations

During this work, we strictly follow ethical considerations for scanning as described in [27], [28]. We limit the rate of all scans, apply a blocklist and filter aliased prefixes based on our own detection but also the list of published aliased prefixes by Gasser *et al.* [4]. We evaluated dynamic TGAs whether they adhere to our scan limits and executed them in an environment where we can monitor their behavior and apply our own blocklist. We inform about our scans based on reverse DNS, a website hosted on the scanning machine and in WHOIS. We respond to all opt-out requests and add address ranges to our internal blocklist.

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

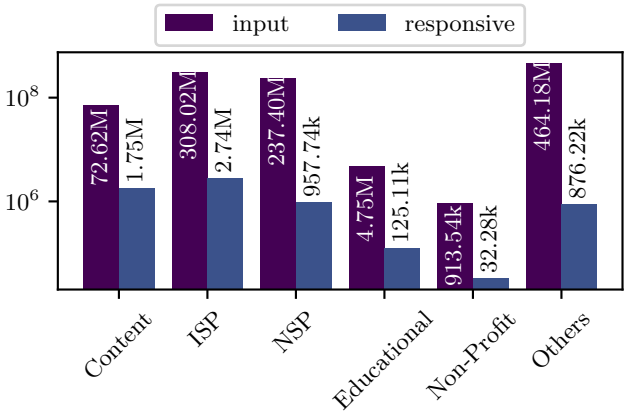

Figure 2: Prevalence of different categories in the *IPv6 Hitlist* on March 3, 2023. Note the logarithmic y-axis.

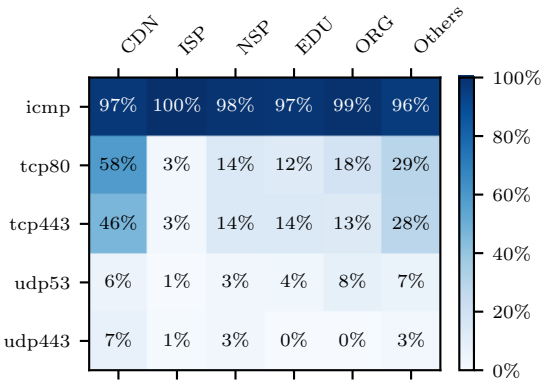

Figure 3: Responses to the different protocols per category within the *IPv6 Hitlist* on March 3, 2023.

V. Results

We analyze the *IPv6 Hitlist* composition with regard to different network categories, compare different TGAs, and investigate the influence of the GFW on our measurements.

A. Hitlist Categorization

First, we analyze the different network categories of the *IPv6 Hitlist*'s input and responsive addresses. The network categories represented in the *IPv6 Hitlist* show different prevalence and behavior. Figure 2 shows the distribution of addresses across categories in the full hitlist input as well as its responsive part. The responsive addresses as well as the full hitlist are dominated by ISP and CDN addresses, with almost 50 % combined.

Next, we analyze the responsiveness in more detail, by looking at different probe protocols and network categories. Figure 3 shows how many protocol-specific responses the latest scan receives per category, relative to the total number of IP addresses per category which responded to at least one protocol probe. Addresses belonging to CDNs have the highest relative number of responses to HTTP and HTTPS probes, with a low, but still comparatively large number of QUIC responses.

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

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

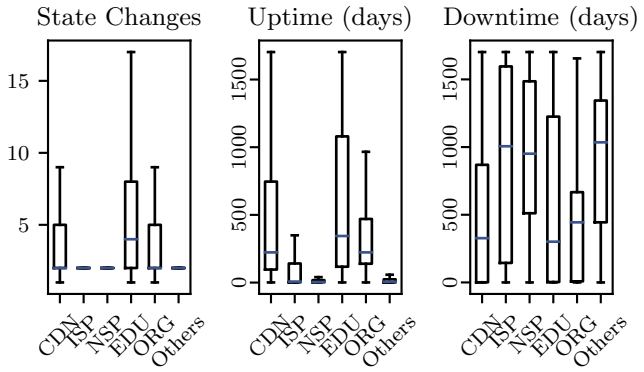

Figure 4: Stability of responsive IPv6 addresses of the *IPv6 Hitlist* per category. Data from scans is used from July 2018 to March 2023, while addresses newly discovered from November 2022 onwards are excluded to reduce the impact of newer scans.

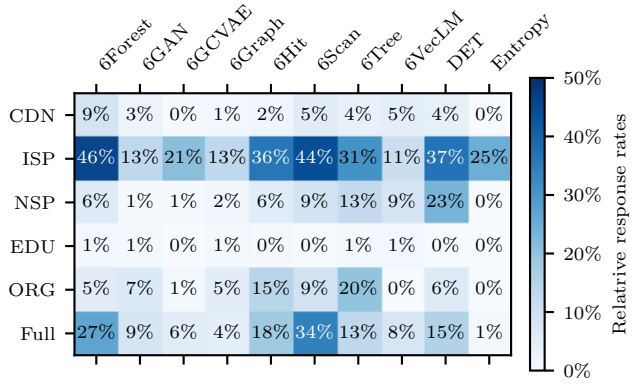

Figure 5: Relative response rate for every candidate set generated by the TGAs with different categorized input sets as well as the full hitlist as input.

This is expected, since web hosting via HTTP/S is one of the primary functions of CDNs, which are also among the first to deploy QUIC at scale [29]. ISP addresses, on the other hand, show almost no response to any protocol other than ICMP.

The high total share of ISP addresses in the hitlist, together with the low response rate to any protocol other than ICMP shows the importance of categorizing hitlists before using them as input for application specific scans, as a large part of scanning traffic can be avoided by carefully selecting target addresses from specific categories.

To better understand the stability of addresses within the *IPv6 Hitlist*, we analyze the categories represented in the hitlist over time using three *IP stability* metrics. First, the number of *state changes*, *i.e.*, the number of times an IP was added to or removed from the responsive part of the hitlist. This can be seen as a lower-bound for the times an IP address changes between online and offline. Second and third, we look at the summed up number of uptimes and downtimes of each address, starting when an IP address is first added to the hitlist, and ending at the time of analysis. These three metrics combined make up the *IP stability* of an address over time. A stable IP address has a small number of state changes, high uptime, and low downtime, the opposite being true for an unstable IP address. For this analysis, in order to avoid analyzing IP addresses with not enough historic data, we exclude all addresses added to the hitlist within the last 100 days.

Figure 4 shows that the different network categories exhibit very distinctive behavior in IP stability. Most addresses from the categories ISP, NSP, and Others have exactly one state change, but median uptimes of less than a week, meaning that they are included once in the responsive hitlist for seven days and never again afterwards. ISP networks often use prefix rotation to avoid tracking of their clients and enhance their privacy [30], [31], which means that devices like home routers often change IP address. Including them in hitlists leads to an increase in unstable targets, which is underlined in the results of this analysis. This also applies to NSP networks, which offer similar services and partly to IP addresses in the Others category. In contrast to this, addresses in CDN, Educational, and Non-Profit networks have much higher uptimes, as addresses hosting content have to be available reliably. The higher number of state changes in these networks can be due to maintenance periods or changes in ownership of the respective servers.

This means that for longitudinal measurement studies which focus on protocols other than ICMP, addresses from categories such as NSP and ISP should be used with care, as they have only very limited periods of responsiveness and generally respond less to protocols other than ICMP, compared to addresses from Content Delivery, Educational or Non-Profit Organization networks.

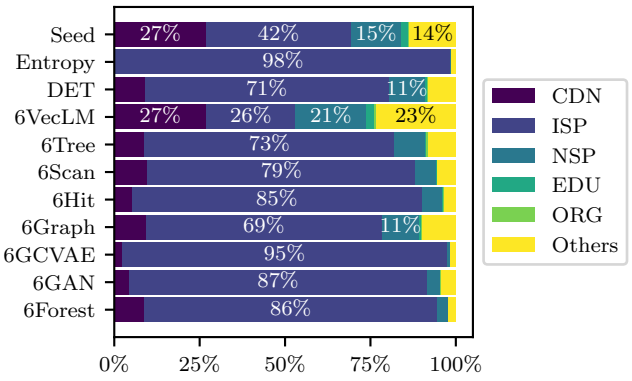

Figure 6: Category distribution of responsive addresses from each Target Generation Algorithm applied on the full set of responsive addresses from the *IPv6 Hitlist*.

B. Target Generation Algorithms

Given the diverse composition of the *IPv6 Hitlist* with respect to covered network categories and the differences in responsiveness and IP stability for these categories, we set out to analyze the properties of TGAs. We evaluate the generated candidate sets of all TGAs in two ZMap scans on March 17 and March 23, 2023 as described in Section IV-B. The scans combined the generation based on the full set of responsive addresses as well as the categorized input sets. The second scan was conducted due to an error in the first one, through which the candidate sets of the dynamic algorithms were not included. For this second scan, including only the candidate sets from the dynamic altorithms, Aliased Prefix Detection (APD) was replaced with a filter for known aliased prefixes from the *IPv6 Hitlist* service. We merge the results from both scans and present the results in the following, together with the different metrics used for the evaluation.

Generation rate and candidate set size. The various algorithms generate vastly different numbers of candidate addresses. Moreover, the generated addresses are also highly dependent on the used seed set. As already elaborated, the prevalence of categories in the hitlist and therefore the size of the categorized seed sets vary (see Figure 2). Therefore, we compare the *generation rate* of the algorithms, *i.e.*, the size of the candidate set relative to the seed set size. Algorithms like 6VecLM and 6GAN have relatively low generation rates, which is due to the fact that these algorithms limit the amount of processed input to a hard coded value (see Section IV). Algorithms such as 6Graph can generate more than $100\,\mathrm{M}$ candidate addresses, which is $1638\,\%$ of the size of the seed sets. With the Non-Profit seed set, 6VecLM is unable to generate a candidate set, as the seed set is smaller than the predefined input size, which we could not successfully modify. For the exact sizes and generation rates, see Appendix Table III.

If the scanning budget of a measurement is critical, large candidate sets should either be sampled or another algorithm should be chosen. If large candidate sizes are desired, large inputs or algorithms with high generation rates are best suited.

Response rates. Internet measurement studies are not only dependent on a scanning budget, but also strive to avoid unnecessary probes which are unlikely to trigger responses. Therefore, it is important to analyze the response rate, *i.e.*, the portion of addresses which responds to at least one protocol, for the different candidate sets. As can be seen in Figure 5, a larger candidate set does not lead to a higher response rate. Instead response rates are more strongly linked with the input set category as well as the difference between dynamic and static algorithms. Dynamic algorithms, due to their ability to adapt their generation strategy based on the results of their scans, have among the top response rates for all categories, up to $45\,\%$ for some. On the other hand, static algorithms rarely show response rates over $15\,\%$, with 6Forest being one of the few exceptions. Using ISP addresses as input shows the best response rates for almost all algorithms, even better than with uncategorized input. Candidate addresses generated from educational networks, on the contrary, have the lowest response rate at hardly over $1\,\%$ for any algorithm.

Again, for measurements with limited scanning budget, choice of input is shown to be critical for the efficiency of the TGAs and subsequently the scans.

Category distribution in responsive addresses. While all TGAs receive the same input, not only do their candidate sets vary greatly in size, but also in the distribution of represented network categories. Figure 6 shows the category distributions in the candidate sets generated by the algorithms and the seed set when using the full hitlist as input. Most algorithms show a strong bias towards ISP addresses, which are also present in the seed data set, although at a much lower percentage. Especially the relatively small percentage of generated CDN addresses is in stark contrast to the ratios of the seed set. When using categorized input, all but two algorithms generate 95–$100\,\%$ of their addresses in the same category as the input. The only exceptions are 6GCVAE and Entropy/IP, which generate up to $62\,\%$ and $13\,\%$ from other categories for some inputs, respectively. These findings show the need to filter the input to TGAs depending on the desired use case, as the algorithms exhibit a strong bias towards some categories.

AS origin distributions. While categorization on an AS level via PeeringDB already gives us some information of the origin of the contained addresses, the exact AS distributions still hold some more information. The cumulative AS distribution of the candidate sets generated from the full hitlist are shown in Figure 7. Most candidate sets generated by the TGAs are more biased towards single ASes: The majority of TGAs contain 50–$95\,\%$ addresses from a single AS, whereas the top ten ASes of the seed set cover only around $50\,\%$ of their addresses. The most popular AS for all but one candidate sets is AS12322 (FREE SAS, an ISP network from France). The only exception to this bias is the candidate set of 6VecLM, which contains only five addresses from AS12322 and is even more evenly distributed among ASes than the seed set dataset. While AS12322 is also the AS with the highest share in the seed data set, it covers only

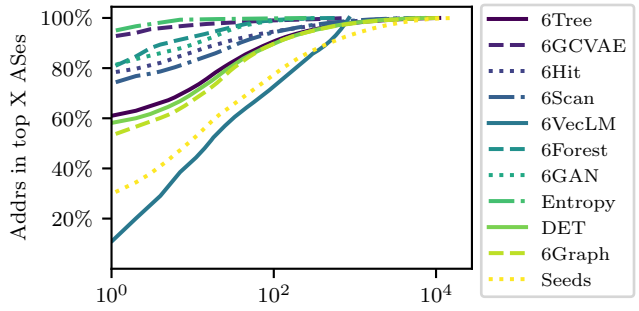

Figure 7: Cumulative AS distribution of the responsive candidate sets generated by the algorithms using the full hitlist as input. Note the logarithmic x-axis.

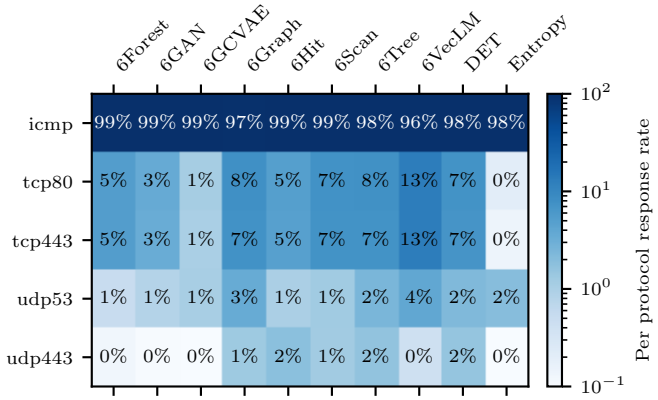

Figure 8: Response rates to the different protocols per algorithm generated on full hitlist input. Note the color map log scale.

30 % of it. Looking at the structure of the addresses from this AS responding to ICMP, it is visible that over 99 % of them have the host part set to `::1`. They are all within the same /39 prefix and only differ in the 10 to 15 nibble of the address. This a very clear structure which has easy to detect patterns, ideal for discovery by TGAs. Addresses from this AS were first added to the hitlist via CT logs and the Bitnodes dataset [4] and their share drastically increased with the first usage of TGAs in the 2022 paper [5]. The high percentage of addresses from AS12322 in most candidate sets also explains their bias towards the ISP network category. This further stresses the need to filter certain categories for use cases where addresses from the respective categories should not be targeted. Furthermore, also addresses from specific ASes should be filtered, as their inclusion in the seed set can introduce biases towards those networks far beyond their presence in the seed set.

Number of covered ASes. Next to the distribution of the ASes contained in the candidate sets, their absolute number is equally relevant. TGAs should generate new targets which represent the active part of the IPv6 Internet, which cannot be achieved if only a small number of ASes is covered in their candidate sets. Most candidate sets cover substantially fewer total ASes than the respective seed sets, especially when using the full hitlist as input. Only in very specific circumstances, when the seed set already contains very few ASes (such as for Educational or NSP), some candidate sets cover more ASes. Very low coverage rates compared to the seed set means that algorithms discover ASes from very specific origins which cannot represent the IPv6 Internet. Even when combined, the candidate sets of all algorithms only cover 75 % of the ASes in the seed set. While the combined candidate sets include 684 ASes which have not been covered by the seed set, 4875 ASes from the seed set are not included. The exact number of newly covered ASes can be seen in Table III.

It is therefore imperative to use additional address sources in order to achieve a higher AS coverage if measurements should cover a representative subset of the IPv6 Internet. It also raises the questions if current TGAs adequately address the need for a balanced candidate set across a large number of ASes.

Ratio of aliased prefix. Aliased prefixes, as defined in Section II, are excluded in our scans as they do not add any valuable information, but instead introduce a bias to the results. It is therefore an important measure of quality for a candidate set to contain few addresses from aliased prefixes. As described in Section IV-B, we conducted APD ourselves and with the aliased prefixes published by the *IPv6 Hitlist* service. We compared the unfiltered sets with non-aliased versions and found that most algorithms have a negligible rate of addresses from aliased prefixes, thereby not impacting the algorithms candidate set quality when filtered. Two exceptions are 6GCVAE and Entropy/IP, which generate up to almost 50 % aliased addresses for some categorized input as well as the full input. This decreases the usable size of their candidate sets substantially, which should be kept in mind before scanning. The exact rate of aliased prefixes can be found in Appendix Table III.

Although the rate of aliased prefixes in most candidate sets were relatively low, which means that only little unnecessary scanning overhead would be introduced, we still stress the need for APD.

Protocol responses. Depending on the use case for the generated addresses, it can be crucial to discover targets with a high response rate to a certain protocol. Figure 8 shows the response rate to the different protocols per candidate set. All candidate sets have the highest response rate to ICMP probes, which is to be expected due to the prevalence in the seed set. Moreover, unlike in IPv4, ICMP in IPv6 can not simply be fully blocked due to its important functionality in stateless address autoconfiguration [32]. Responses to other protocols are much less frequent for all candidate sets. Especially the response rate for HTTP and HTTPS is very similar to the share of non-ISP addresses in the responsive portion of the candidate sets. Looking at Figure 6, we can see that the responses to the candidate set of 6VecLM have the lowest share of ISP addresses and the highest number of responses to HTTP and HTTPS. Entropy/IP and 6GCVAE on the other hand, have

more than 95 % ISP addresses in their respective responses and the lowest share of protocol responses other than ICMP. The per-protocol response rate for the candidate sets generated with categorized inputs show a very strong correlation with the per-category response rates of the hitlist, see Figure 3. With CDN input addresses, all candidate sets receive between 30 and 65 % HTTP and HTTPS responses, whereas with ISP input, no candidate set generates more than 3 % response rate for any protocol besides ICMP.

This shows that input and algorithm should be chosen carefully depending on the desired protocol responses for the use case.

C. GFW Filtering

As described in Figure 1, the responses to our DNS probes are post-processed with a *GFW Filter*. Our probes contain `AAAA` queries for `www.google.com` and frequently receive responses with addresses from the *Teredo* prefix in their answer section. These responses do not originate from legitimate hosts, but are instead likely injected by the GFW for reasons of censorship [5]. `google.com` is on the list of censored domains in China [33] and no web service for `www.google.com` is reachable at the returned addresses. Following the description of Zirngibl *et al.* [5], we filter all responses containing addresses from the *Teredo* prefix in their answer section and do not count them as responsive in our work.

In both our scans, however, we see a substantial change in the format of the injections, as we receive responses containing addresses from Facebook's network in their answer section. This change in behavior indicates that the GFW tries to adapt the IPv6 injections to the format of their IPv4 injections, which contain addresses from a fixed pool of IPv4 addresses, including addresses from Facebook and Twitter [34]. Until recently, every returned Teredo address encoded a corresponding address from the IPv4 pool in the last 32 bits of the address, whereas now, a separate pool of IPv6 addresses from similar networks such as Facebook is being returned. We argue that, at this moment and for probes querying `www.google.com`, filtering out responses containing addresses from Facebook is sufficient, as our scans show similar response rates to DNS probes as the *IPv6 Hitlist* service. To the best of our knowledge, this new behavior of the GFW has not been documented before. DNS scans targeting IPv6 addresses need to take this behavior into account and adjust the filtering pipeline accordingly. Since we, however, expect that the GFW can change its behavior in unpredictable ways, we chose not to adapt the filter of the *IPv6 Hitlist* service to this new type of injection, and instead changed the domain name which is used in the regular query probes. The new domain name is not censored by the GFW and does not trigger any injections, which we expect to remain the same in the future. We argue that this yields more stable and usable results for researchers using the *IPv6 Hitlist*.

VI. Discussion and Conclusion

In this work, we have highlighted the dependency of IPv6 measurements on their targets. We have shown that address collections such as the *IPv6 Hitlist* service contain multiple types of networks, including ISPs, CDNs, and NSPs, with different behavior. While addresses from CDN networks respond to HTTP and HTTPS at a rate of around 50 % and are responsive for a median time span of more than 200 days, ISP addresses are mostly only available for a single scan and only respond to ICMP for 97 % of the addresses. Furthermore, we evaluated the behavior of different Target Generation Algorithms, using the full and categorized versions of the hitlist as input. We demonstrated that the input has a strong influence on various metrics, such as the number of generated and responsive addresses, protocol responses, and addresses origin. All but one candidate sets generated from uncategorized input show a very strong bias towards ISP networks, which in turn have a strong bias towards single ASes and generally have a response rate below 10 % for any protocol other than ICMP. Output from categorized seed sets consists of addresses from the respective input category, exhibiting behavior similar to the addresses from the respective categories in the hitlist. However, we learned that most TGAs are complex tools and the majority of published tool chains are trained and optimized on a specific input. While we tried to adapt the algorithms and parameters to suite our use cases and scenarios, we were not able to reach published rates of responsive addresses. Furthermore, algorithms with dynamic scanning capabilities are not suited for all use cases, as the adherence to scanning rates, blocklists and detection of aliased prefixes cannot be achieved without modifications. Our work provides a detailed comparison under different circumstances to allow for a selection of suitable TGAs and a more focused analysis and optimization in the future. As an example, a scan application which requires large numbers of targets and does not have tight restraints on scan budgets, should opt for an algorithm such as 6Graph or 6Tree, as they generate the largest candidate sets. Scenarios, on the other hand, which dictate efficient scanning with a high response rate and do not require modifications to the candidate set before scanning, are best suited for dynamic algorithms, as they reach the highest response rates.

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

Acknowledgments

This work was partially funded by the German Federal Ministry of Education and Research under the projects PRIMEnet (16KIS1370), 6G-life (16KISK002) and 6G-ANNA (16KISK107).

References

[1] Google, *Google IPv6 Statistics*, https://www.google.com/intl/en/ipv6/statistics.html.

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

[3] D. Adrian, Z. Durumeric, G. Singh, and J. A. Halderman, "Zippier ZMap: Internet-Wide Scanning at 10 Gbps," in *WOOT*, 2014.

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

[5] J. Zirngibl, L. Steger, P. Sattler, O. Gasser, and G. Carle, "Rusty clusters? dusting an IPv6 research foundation," in *Proc. ACM Int. Measurement Conference (IMC)*, 2022. DOI: 10.1145/3517745.3561440.

[6] T. Cui, G. Gou, and G. Xiong, "6GCVAE: Gated Convolutional Variational Autoencoder for IPv6 Target Generation," in *Advances in Knowledge Discovery and Data Mining*, 2020. DOI: 10.1007/978-3-030-47426-3_47.

[7] T. Cui, G. Gou, G. Xiong, C. Liu, P. Fu, and Z. Li, "6GAN: IPv6 Multi-Pattern Target Generation via Generative Adversarial Nets with Reinforcement Learning," in *Proc. IEEE Int. Conference on Computer Communications (INFOCOM)*, 2021. DOI: 10.1109/infocom42981.2021.9488912.

[8] T. Yang, B. Hou, Z. Cai, K. Wu, T. Zhou, and C. Wang, "6Graph: A graph-theoretic approach to address pattern mining for Internet-wide IPv6 scanning," *Computer Networks*, vol. 203, Feb. 2022. DOI: 10.1016/j.comnet.2021.108666.

[9] P. Foremski, D. Plonka, and A. Berger, "Entropy/IP: Uncovering Structure in IPv6 Addresses," in *Proc. ACM Int. Measurement Conference (IMC)*, 2016. DOI: 10.1145/2987443.2987445.

[10] L. Steger, L. Kuang, J. Zirngibl, G. Carle, and O. Gasser, *Data and analysis at tum university library*, Dataset, 2023. DOI: 10.14459/2023mp1709953.

[11] Z. Liu, Y. Xiong, X. Liu, W. Xie, and P. Zhu, "6Tree: Efficient dynamic discovery of active addresses in the IPv6 address space," *Computer Networks*, vol. 155, May 2019. DOI: 10.1016/j.comnet.2019.03.010.

[12] G. Song, J. Yang, Z. Wang, L. He, J. Lin, L. Pan, C. Duan, and X. Quan, "DET: Enabling Efficient Probing of IPv6 Active Addresses," *IEEE/ACM Transactions on Networking*, vol. 30, no. 4, Aug. 2022. DOI: 10.1109/tnet.2022.3145040.

[13] T. Cui, G. Xiong, G. Gou, J. Shi, and W. Xia, "6VecLM: Language Modeling in Vector Space for IPv6 Target Generation," in *Machine Learning and Knowledge Discovery in Databases: Applied Data Science Track*, 2021. DOI: 10.1007/978-3-030-67667-4_12.

[14] B. Hou, Z. Cai, K. Wu, J. Su, and Y. Xiong, "6Hit: A Reinforcement Learning-based Approach to Target Generation for Internet-wide IPv6 Scanning," in *Proc. IEEE Int. Conference on Computer Communications (INFOCOM)*, 2021. DOI: 10.1109/infocom42981.2021.9488794.

[15] T. Yang, Z. Cai, B. Hou, and T. Zhou, "6Forest: An Ensemble Learning-based Approach to Target Generation for Internet-wide IPv6 Scanning," in *Proc. IEEE Int. Conference on Computer Communications (INFOCOM)*, 2022. DOI: 10.1109/infocom48880.2022.9796925.

[16] B. Hou, Z. Cai, K. Wu, T. Yang, and T. Zhou, "6Scan: A High-Efficiency Dynamic Internet-Wide IPv6 Scanner With Regional Encoding," *IEEE/ACM Transactions on Networking*, 2023. DOI: 10.1109/tnet.2023.3233953.

[17] *AS Classification*, https://catalog.caida.org/dataset/as_classification.

[18] M. Ziv, L. Izhikevich, K. Ruth, K. Izhikevich, and Z. Durumeric, "ASdb," in *Proc. ACM Int. Measurement Conference (IMC)*, 2021. DOI: 10.1145/3487552.3487853.

[19] F. Gont and T. Chown, *Network Reconnaissance in IPv6 Networks*, RFC 7707, Mar. 2016. DOI: 10.17487/RFC7707. [Online]. Available: https://www.rfc-editor.org/info/rfc7707.

[20] J. Ullrich, P. Kieseberg, K. Krombholz, and E. Weippl, "On Reconnaissance with IPv6: A Pattern-Based Scanning Approach," in *10th International Conference on Availability, Reliability and Security*, 2015. DOI: 10.1109/ares.2015.48.

[21] A. Murdock, F. Li, P. Bramsen, Z. Durumeric, and V. Paxson, "Target generation for internet-wide IPv6 scanning," in *Proc. ACM Int. Measurement Conference (IMC)*, 2017. DOI: 10.1145/3131365.3131405.

[22] E. C. Rye and R. Beverly, "Discovering the IPv6 Network Periphery," in *Passive and Active Measurement*, 2020. DOI: 10.1007/978-3-030-44081-7_1.

[23] X. Li, B. Liu, X. Zheng, H. Duan, Q. Li, and Y. Huang, "Fast IPv6 Network Periphery Discovery and Security Implications," in *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2021. DOI: 10.1109/dsn48987.2021.00025.

[24] R. Beverly, R. Durairajan, D. Plonka, and J. P. Rohrer, "In the IP of the Beholder: Strategies for Active IPv6 Topology Discovery," in *Proc. ACM Int. Measurement Conference (IMC)*, 2018. DOI: 10.1145/3278532.3278559.

[25] "University of Oregon Route Views Project." (), [Online]. Available: http://www.routeviews.org/routeviews/.

[26] *PeeringDB*, https://catalog.caida.org/dataset/peeringdb.

[27] D. Dittrich, E. Kenneally, *et al.*, "The Menlo Report: Ethical principles guiding information and communication technology research," *US Department of Homeland Security*, 2012.

[28] C. Partridge and M. Allman, "Addressing Ethical Considerations in Network Measurement Papers," *Communications of the ACM*, vol. 59, no. 10, Oct. 2016.

[29] J. Zirngibl, P. Buschmann, P. Sattler, B. Jaeger, J. Aulbach, and G. Carle, "It's over 9000: analyzing early QUIC deployments with the standardization on the horizon," in *Proc. ACM Int. Measurement Conference (IMC)*, 2021. DOI: 10.1145/3487552.3487826.

[30] E. Rye, R. Beverly, and K. C. Claffy, "Follow the Scent: Defeating IPv6 Prefix Rotation Privacy," in *Proceedings of the 21st ACM Internet Measurement Conference*, 2021. DOI: 10.1145/3487552.3487829.

[31] S. J. Saidi, O. Gasser, and G. Smaragdakis, "One Bad Apple Can Spoil Your IPv6 Privacy," *ACM SIGCOMM Computer Communication Review*, vol. 52, 2 Jun. 2022. DOI: 10.1145/3544912.3544915.

[32] S. Thomson, T. Narten, and T. Jinmei, *IPv6 Stateless Address Autoconfiguration*, RFC 4862 (Draft Standard), RFC, RFC Editor, Sep. 2007. DOI: 10.17487/RFC4862. [Online]. Available: https://www.rfc-editor.org/rfc/rfc4862.txt.

[33] N. Hoang, A. Niaki, J. Dalek, J. Knockel, P. Lin, B. Marczak, M. Crete-Nishihata, P. Gill, and M. Polychronakis, "How Great is the Great Firewall? Measuring China's DNS Censorship," in *30th USENIX Security Symposium*, 2021.

[34] Anonymous, A. A. Niaki, N. P. Hoang, P. Gill, and A. Houmansadr, "Triplet Censors: Demystifying Great Firewall's DNS Censorship Behavior," in *10th USENIX Workshop on Free and Open Communications on the Internet (FOCI 20)*, 2020.

Table III: Appendix: Overview of different metrics for the algorithm per categorized input.

		6Forest	6GAN	6GCVAE	6Graph	6Hit	6Scan	6Tree	6VecLM	DET	Entropy
Number of candidate addresses	Content	1.94 M	486.59 k	3.28 M	34.85 M	10.02 M	8.94 M	11.08 M	77.82 k	8.56 M	5.87 M
	NSP	2.09 M	521.17 k	2.60 M	30.95 M	9.71 M	9.48 M	9.16 M	66.36 k	1.63 M	5.52 M
	Educational	1.44 M	316.24 k	699.80 k	1.98 M	24.02 M	9.88 M	11.05 M	83.54 k	1.06 M	4.48 M
	Non-Profit	710.61 k	125.30 k	284.16 k	295.57 k	19.57 M	9.97 M	7.89 M	0	5.87 M	3.79 M
	ISP	3.33 M	409.52 k	845.24 k	24.57 M	8.37 M	7.91 M	11.17 M	18.24 k	7.79 M	5.98 M
	Full	1.84 M	486.22 k	2.00 M	106.12 M	17.92 M	5.87 M	35.39 M	48.55 k	8.30 M	5.63 M
Generation factor	Content	1.11	0.28	1.88	19.97	5.74	5.13	6.35	0.04	4.90	3.36
	NSP	2.18	0.54	2.72	32.29	10.13	9.89	9.55	0.07	1.70	5.76
	Educational	11.48	2.53	5.60	15.82	192.22	79.09	88.38	0.67	8.48	35.83
	Non-Profit	22.01	3.88	8.80	9.15	605.98	308.69	244.39	0.00	181.82	117.42
	ISP	1.21	0.15	0.31	8.91	3.03	2.87	4.05	0.01	2.82	2.17
	Full	0.28	0.08	0.31	16.38	2.77	0.91	5.46	0.01	1.28	0.87
Number of responsive addresses	Content	173.90 k	13.42 k	13.97 k	443.44 k	230.62 k	491.22 k	416.75 k	3.87 k	360.54 k	7.62 k
	NSP	128.27 k	3.66 k	15.31 k	527.19 k	552.35 k	884.09 k	1.15 M	5.68 k	381.85 k	15.86 k
	Educational	19.37 k	2.62 k	585	22.04 k	99.82 k	37.73 k	106.52 k	1.23 k	745	2.59 k
	Non-Profit	38.91 k	8.50 k	2.61 k	15.12 k	2.86 M	946.16 k	1.58 M	0	355.65 k	13.66 k
	ISP	1.53 M	55.22 k	179.00 k	3.27 M	3.03 M	3.50 M	3.45 M	2.06 k	2.89 M	1.49 M
	Full	494.21 k	41.36 k	111.48 k	4.74 M	3.31 M	2.01 M	4.71 M	3.81 k	1.28 M	59.25 k
Relative response rate	Content	8.96 %	2.76 %	0.43 %	1.27 %	2.30 %	5.49 %	3.76 %	4.97 %	4.21 %	0.13 %
	NSP	6.15 %	0.70 %	0.59 %	1.70 %	5.69 %	9.33 %	12.54 %	8.56 %	23.46 %	0.29 %
	Educational	1.35 %	0.83 %	0.08 %	1.11 %	0.42 %	0.38 %	0.96 %	1.47 %	0.07 %	0.06 %
	Non-Profit	5.48 %	6.79 %	0.92 %	5.12 %	14.64 %	9.49 %	19.99 %	0 %	6.06 %	0.36 %
	ISP	45.88 %	13.48 %	21.18 %	13.29 %	36.21 %	44.28 %	30.89 %	11.29 %	37.07 %	24.97 %
	Full	26.85 %	8.51 %	5.58 %	4.46 %	18.46 %	34.31 %	13.32 %	7.84 %	15.46 %	1.05 %
Aliased prefix ratio	Content	1.75 %	6.14 %	35.14 %	0.21 %	0.38 %	0.15 %	0.12 %	0.37 %	0.31 %	40.94 %
	NSP	1.29 %	11.18 %	34.53 %	0.14 %	0.21 %	0.18 %	1.43 %	0.11 %	1.84 %	44.21 %
	Educational	0.66 %	14.92 %	17.81 %	0.73 %	4.26 %	0.44 %	0.07 %	0.08 %	0.63 %	50.06 %
	Non-Profit	1.78 %	2.22 %	13.89 %	2.53 %	0.02 %	0.13 %	21.02 %	0 %	11.39 %	41.67 %
	ISP	0.68 %	4.12 %	31.47 %	0.28 %	0.17 %	0.09 %	0.04 %	2.82 %	0.63 %	22.13 %
	Full	2.26 %	12.77 %	42.98 %	0.30 %	0.28 %	0.21 %	0.08 %	0.71 %	0.99 %	43.30 %
Candidate ASes	Content	1.03 k	1.22 k	5.14 k	4.03 k	969	884	1.08 k	675	2.64 k	3.13 k
	NSP	2.59 k	2.98 k	5.21 k	7.16 k	1.87 k	1.74 k	2.00 k	1.34 k	1.31 k	6.32 k
	Educational	819	1.33 k	980	2.36 k	493	473	627	572	469	2.50 k
	Non-Profit	430	377	215	1.50 k	263	264	323	0	147	3.29 k
	ISP	2.10 k	1.77 k	3.86 k	10.08 k	3.26 k	2.82 k	4.25 k	1.47 k	10.02 k	6.06 k
	Full	4.39 k	4.04 k	6.19 k	20.99 k	10.22 k	10.43 k	16.82 k	3.87 k	19.65 k	7.64 k
Responsive ASes	Content	186	121	1.24 k	1.04 k	668	654	803	207	1.05 k	62
	NSP	466	97	1.23 k	2.06 k	1.40 k	1.41 k	1.59 k	557	357	292
	Educational	230	101	250	538	334	354	429	145	57	212
	Non-Profit	205	62	60	311	179	194	222	0	44	576
	ISP	240	169	881	3.65 k	2.13 k	1.76 k	3.15 k	322	3.82 k	220
	Full	618	336	814	10.97 k	6.52 k	5.52 k	10.94 k	844	7.16 k	252
Coverage of seed ASes	Content	17.77 %	11.56 %	118.43 %	99.14 %	63.80 %	62.46 %	76.70 %	19.77 %	100.76 %	5.92 %
	NSP	23.57 %	4.91 %	62.11 %	104.05 %	70.71 %	71.52 %	80.58 %	28.17 %	18.06 %	14.77 %
	Educational	38.40 %	16.86 %	41.74 %	89.82 %	55.76 %	59.10 %	71.62 %	24.21 %	9.52 %	35.39 %
	Non-Profit	65.92 %	19.94 %	19.29 %	100.00 %	57.56 %	62.38 %	71.38 %	0 %	14.15 %	185.21 %
	ISP	5.66 %	3.99 %	20.79 %	86.10 %	50.31 %	41.58 %	74.28 %	7.60 %	90.09 %	5.19 %
	Full	3.63 %	1.97 %	4.78 %	64.48 %	38.34 %	32.42 %	64.31 %	4.96 %	42.10 %	1.48 %
Number of newly covered ASes	Content	27	8	1.14 k	268	20	14	23	0	296	44
	NSP	52	32	1.05 k	531	66	49	25	0	70	255
	Educational	32	15	231	128	12	24	15	0	10	191
	Non-Profit	65	7	57	109	15	18	10	0	13	555
	ISP	18	6	688	639	26	13	10	0	1.18 k	161
	Full	55	3	100	359	24	6	16	0	286	50