Discovering the IPv6 Network Periphery

Erik C. Rye[0000−0002−8151−8252] and Robert Beverly[0000−0002−5005−7350]

Naval Postgraduate School, Monterey, CA
rye@cmand.org, rbeverly@nps.edu

Abstract. We consider the problem of discovering the IPv6 network periphery, i.e., the last hop router connecting endhosts in the IPv6 Internet. Finding the IPv6 periphery using active probing is challenging due to the IPv6 address space size, wide variety of provider addressing and subnetting schemes, and incomplete topology traces. As such, existing topology mapping systems can miss the large footprint of the IPv6 periphery, disadvantaging applications ranging from IPv6 census studies to geolocation and network resilience. We introduce "edgy," an approach to explicitly discover the IPv6 network periphery, and use it to find > 64M IPv6 periphery router addresses and > 87M links to these last hops – several orders of magnitude more than in currently available IPv6 topologies. Further, only 0.2% of edgy's discovered addresses are known to existing IPv6 hitlists.

Keywords: IPv6 · Topology · Discovery · Reconnaissance · Security

1 Introduction

Among the unique properties inherent to IPv6's large address space size are ephemeral and dynamic addressing, allocation sparsity and diversity, and a lack of address translation. These well-known properties complicate efforts to map the infrastructure topology of the IPv6 Internet. While previous research has tackled problems of target selection, speed, and response rate-limiting in active IPv6 topology probing [7], the IPv6 *periphery* – last hop routed infrastructure connecting end hosts – is challenging to discover, and difficult to discern.

Discovery of the IPv6 periphery is important not only to the completeness of network topology mapping, but provides a crucial supporting basis for many applications. For instance, IPv6 adoption [12,34,27], census [26], and reliability and outage studies [21] all depend in part on a complete and accurate map of the IPv6 topology inclusive of the periphery, while understanding provider address allocation policies and utilization also requires completeness [15,29]. Similarly, work on IPv4 to IPv6 network congruence [13,20] and IPv6 geolocation [5] can utilize IPv6 topologies. Further, our work illuminates potential security and privacy vulnerabilities inherent in the way today's IPv6 periphery is deployed [11,31].

We present "edgy," a new technique to explicitly discover the IPv6 periphery. In contrast to IPv6 scanning [23,17], passive collection [26], or hitlists [16,14], which, by construction, target endhosts, edgy is specifically designed to find last hop routers and subnetworks in the IPv6 Internet. Our contributions include:

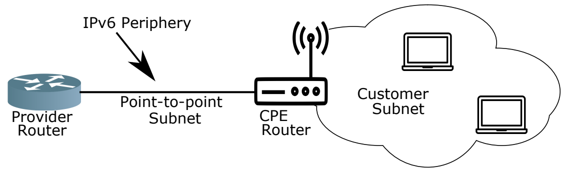

Fig. 1. Common IPv6 architecture: an IPv6 subnet is assigned to the link between the provider and last hop CPE routers. There is no NAT or private addressing; a separate distinct routed IPv6 subnet is assigned to devices attached to the last hop CPE.

1. Edgy, an algorithm to discover, identify, and enumerate the IPv6 periphery.
2. Active measurement using edgy to find 64.8M last hop router addresses and 87.1M edges to these last hops from a single vantage.
3. Discovery of periphery addresses that are 99.8% disjoint from current IPv6 hitlists [16] and orders of magnitude larger than existing IPv6 topology snapshots [8], suggesting that edgy is complementary to these prior approaches.
4. Discovery of 16M EUI-64 last hop addresses, suggesting a potential vulnerability to security and privacy.

2 Background and Related Work

In this work, we define the "periphery" not to be servers or clients, but rather the last hop router connecting network endhosts. Whereas significant prior work has developed techniques for IPv6 endhost discovery [23,17,16], comparatively little work has explored the IPv6 periphery.

The large address space in IPv6 removes the need for address translation; thus, while many IPv4 hosts are connected via NATs [32], the IPv6 periphery typically extends into customer premises. Indeed, in IPv6, the Customer Premises Equipment (CPE) is a router, implying that in conjunction with the rapid increase in IPv6 adoption [12,34], the IPv6 periphery is considerably larger than in IPv4, especially for residential networks.

Figure 1 shows an example of the IPv6 periphery we attempt to discover. Here, the point-to-point subnet between the provider and the CPE is assigned a public IPv6 prefix; the subnet on the other side of the CPE (e.g., in the customer's home) is also a publicly-routed prefix. While this example shows a common residential IPv6 architecture, similar designs exist in the enterprise.

Consider an IPv6 traceroute to a random address within a provider's globally advertised BGP prefix, such as is routinely performed by existing production topology mapping systems [18]. The traceroute (Figure 2): i) is unlikely to hit the prefix allocated to a customer CPE or her network; ii) is even less likely to reach a host within the customer's network; and iii) does not illuminate the scope, characteristics, or breadth of subnets within the prefix. When a traceroute does not reach its target destination it is ambiguous: does the last responsive hop belong to the core of the network, or the periphery?

Passive techniques suffer similar problems in revealing the network periphery. For instance, BGP, by design aggregates routes such that the aggregate visible

```
traceroute to 2a03:4980:2b6:9624:8643:b70f:adae:4f40
    . . .
 5  2001:7f8:1::a502:4904:1  16.862 ms
 6  2a03:4980::6:0:2  25.948 ms
 7  2a03:4980::b:0:5  39.560 ms
 8  *
 9  *
```

Fig. 2. Randomly chosen trace targets are unlikely to discover subnets within a prefix, or to elicit a response. It is thus ambiguous whether hop 7 is a periphery address in this example, even though the trace reaches into the destination's /32.

in a looking glass does not reveal the subnets within. And, while there has been significant prior work in characterizing the IPv6 address space, these primarily focus on endhosts. For example, Plonka and Berger examine and analyze the addresses and behaviors of IPv6 clients connecting to a large CDN [26]. However, this passive collection of client requests alone does not reveal the network periphery on the path to those clients.

3 Methodology

Our work seeks to perform active probing in a way that elicits responses from the last hop IPv6 periphery, rather than network core infrastructure, servers or other endhosts. Enumerating last hop router addresses, e.g., CPE, and inferring networks beyond the last hops are the principal goals of edgy.

Edgy is divided into an initialization stage, followed by active probing that proceeds in rounds. Results from one round of probing are used to guide probing in subsequent rounds. This section describes edgy; the complete algorithm is given in Appendix A.

3.1 Edgy

Because of the massive size of the IPv6 address space, edgy relies on an input set of "seed traces" to focus and guide its discovery. Thus, the ability of edgy to discover the network periphery depends strongly on the input seed traces it uses. In §3.2 we describe two specific realistic seed traces we utilize: i) BGP-informed; and ii) hitlist-informed.

Algorithm 1 describes edgy's initialization stage. Edgy iterates through the input seed and examines the last responsive hop in each trace, regardless of whether a sequence of same last IP responses or loops occur. It maintains the set of targets that, when used as the traceroute destination, had a given last hop. Edgy then finds *unique* last hops – those that were only discovered by probing destinations that reside within a single /48 prefix. The intuition is to find candidate /48 prefixes that are likely to be subnetted, and hence contain periphery routers. By contrast, if there are two or more probes to targets in different /48s that elicit the same last hop, those /48s are less likely to be subnetted, or traces to targets in these /48s are unresponsive beyond the middle of the network. In either case, edgy terminates exploration of these target /48s rather than continuing to probe them.

These candidate target /48 prefixes are fed to Algorithm 2 which probes targets within the input prefixes at progressively finer granularities until a stopping condition (a discovery threshold η) is reached. A random Interface IDentifier (IID) (the 64 least significant bits in an IPv6 address) for each target subnet is used as the trace destination. Figure 3 depicts an illustration of edgy's first round behavior targeting an example /48 belonging to Cox Communications.

The first subnet discovery round probes different /56 prefixes and serves as a coarse filter to determine which candidate /48s exhibit an appreciable amount of subnetting and merit further probing. /56s are used initially as [28] recommends this as a potential subnet size for residential customers; therefore, if a /48 is allocated entirely to residential customers with /56s, the initial probing round should discover all of the /56 allocations. We note, however, that these prefix delegation boundaries are not mandatory, that it is impossible to know *a priori* what prefix delegation strategy a provider has chosen, and that networks can be subdivided in a non-uniform manner for allocations to customers. If the number of distinct last hops found during a probing round exceeds the threshold η, we further subdivide responsive prefixes for additional probing in the next round. The choice and sensitivity of η are discussed in [30].

It has been shown that aliased networks are common in the IPv6 Internet, where every address within a prefix is responsive despite no actual host being present. We remove last hops equal to the probe target, as well as networks and addresses present in the publicly curated list of aliases from Gasser et al. [16]. In addition, we remove replies from non-routable prefixes – we observe site- and link-local addresses that fall into this category – as well as IPv4-in-IPv6 addresses and replies that appear to be spoofed.

After removing aliases and bogus replies, target /48s that generate $> \eta$ unique last hop addresses proceed to the second round of probing. In the second round, edgy sends probes to addresses within each /60 of the target /48. Figure 4 depicts an illustration of edgy's second round behavior, again for the same Cox Communications /48. Target /48 networks that generate $> \eta$ unique last hop addresses (exclusive of aliases) move to the next round. The third probing round sends probes to a random IPv6 address in each /62 of the target networks. Finally, target /48s that exhibit subnetting beyond the /60 level (as evidenced by four unique last hops for each /62 within any /60), are probed at /64 granularity.

Note that, during testing, we initially explored other periphery discovery mechanisms. For instance, intuitively, a binary-tree discovery process that bisects prefixes and probes each half would programmatically explore subnets. Unfortunately, such an efficient approach performs poorly as providers do not allocate subnets uniformly. In this case, a core router can falsely appear as the common last hop for destinations in a common prefix, even when significant subnetting is present. Additionally, the third round of probing was added to limit

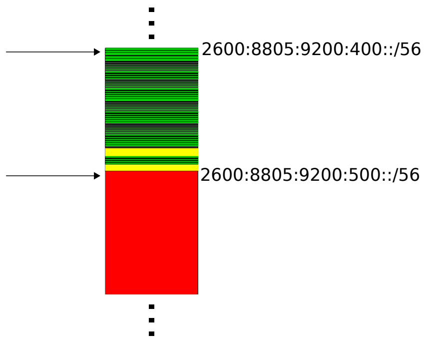

Fig. 3. A portion of a target /48 (2600:8805:9200::/48) is shown; colors correspond to the true delegated customer subnet sizes that edgy discovers. Green represents /64, yellow /60, and red /56. In the first probing round, edgy sends probes to each /56 in a target /48 (represented by arrows).

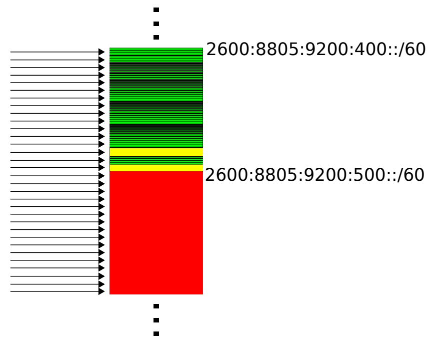

Fig. 4. In the second round, probes are sent to each /60 in the target /48. New addresses are discovered in the upper half of this portion of the target address space where subnet allocation is finer- grained, but not in the lower half. Many operators mix allocation sizes within the same /48.

time spent probing target networks at the /64 granularity to those proven to subnet within the final nybble of the network prefix.

3.2 Edgy Input

Edgy takes as input a seed set of traces. These seed traces are created from running traceroutes to corresponding seed targets. We consider two realistic potential seed target lists: BGP-informed and hitlist-informed. The BGP-informed targets assume no prior knowledge other than global BGP advertisements. Since BGP routes are readily available from looking glasses, this scenario is easily replicated by anyone and models what CAIDA uses to inform their probing. In our experiments, we utilize publicly available BGP-informed seed traces collected as part of an August 2018 effort to uniformly probe every /48 in the IPv6 Internet [29,9]. Herein, we term this trace seed data as the *BGP-informed* seed.

Second, we consider a target set informed by prior knowledge in the form of passive traces, server logs, or hitlists. In our experiments, we utilize a publicly available IPv6 hitlist [16] that was used to generate a seed set of hitlist-informed traces [7]. Herein, we term this trace seed the *hitlist-informed seed*.

3.3 Limitations

There are several potential complications that edgy may encounter, and corresponding limitations of our approach and evaluation. First, during probing, we depend on receiving a response from the penultimate traceroute hop along the data path to a destination. However, the last responsive hop may instead be a different router due to filtering, loss, or rate-limiting, i.e., if the last hop remains anonymous. This case does not cause false inferences of periphery addresses, but instead causes edgy to terminate probing of a prefix prematurely.

Second, we do not have ground-truth in order to determine whether the periphery we discover is indeed the last hop before a destination endhost. While various, and at times conflicting, guidance exists regarding the size of delegated prefixes [19,25,10] discovery of unique /64s is strongly indicative of discovering the periphery. Additionally, the periphery addresses we find are frequently formed using EUI-64 addresses where we can infer the device type based on the encoded MAC address (see §4.5). These MAC addresses specifically point to CPE. Further, we examine several metrics of "edginess" to better understand the results in §4.3. In particular, we determine whether traces enter their target network and, if so, quantify how far inside the target network they reach. We also analyze the last hop addresses edgy discovers in order to understand how many also appear as intermediate hops to different targets. As intermediate hops, such addresses are unlikely to exist in the IPv6 periphery.

3.4 Probing

Probing consists of sending hop-limited ICMPv6 packets; we used the high-speed randomized yarrp topology prober [6] due to the large number of traces required during edgy's exploration, as well as to minimize the potential for ICMPv6 rate limiting (which is mandated and common in IPv6 [7]).

We use ICMPv6 probes as these packets are designed for diagnostics and therefore are less intrusive than UDP probes. Further, we send at a conservative rate while yarrp, by design, randomizes its probing in order to minimize network impact. Last, we follow best established practices for performing active topology probing: we coordinated with the network administrators of the vantage point prior to our experiments and hosted an informative web page on the vantage point itself describing the experiment and providing opt-out instructions. We received no opt-out requests during this work.

4 Results

From Sept. to Oct. 2019 we ran edgy from a well-connected server in Lausanne, Switzerland. Edgy used yarrp at less than 10kpps with the neighborhood TTL setting to reduce load on routers within five hops of the vantage point.

4.1 BGP-Informed Seed Results

Initializing edgy with the BGP-informed seed data yielded 130,447 candidate /48 prefixes. Following Algorithm 2, edgy traced to a random IID in each of the 256 constituent /56 subnets in each /48s (a total of 33,394,432 distinct traces).

This first round of probing 33.4M targets discovered 4.6M unique, non-aliased last hop IPv6 addresses residing in 33,831 distinct /48 prefixes (Table 1). Often,

Table 1. BGP and Hitlist-Informed Routable Address Discovery by Round

	BGP-Informed				Hitlist-Informed			
Round	**Prefixes Probed**	**Unique Last Hops**	**Unique Last Hop /48s**	**Cum. Unique Last Hops**	**Prefixes Probed**	**Unique Last Hops**	**Unique Last Hop /48s**	**Cum. Unique Last Hops**
1 (/56)	130,447	4,619,692	33,831	4,619,692	111,670	9,217,137	89,268	9,217,137
2 (/60)	34,520	12,228,916	26,082	13,410,601	67,107	11,021,329	74,302	11,365,910
3 (/62)	12,014	14,770,061	11,675	24,832,391	4,462	5,428,992	19,942	15,569,221
4 (/64)	2,641	15,326,298	7,833	37,169,357	1,531	15,340,591	32,718	29,248,703

the last hop address is not contained within the target /48 prefix but in a different /48 prefix belonging to the same Autonomous System (AS). Further, probing different target /48 prefixes in round one resulted in last hops within the same /48 (but different than the target /48). This phenomenon of a many-to-one relationship between the target prefix and the last hop prefix persists across rounds as the probing granularity increases.

The density of discovered last hop addresses across target prefixes is non-uniform: nearly 75% of the targeted /48 prefixes produce 16 or fewer distinct last hops. The prefixes in which the last hops reside is also highly non-uniform. Of the 33,831 /48s in which last hop addresses reside, 11,064 were responsible for only a single last hop address. This is likely indicative of a /48 allocation to an end site. On the other end of the spectrum, a single /48 (`2001:1970:4000::/48`) contained over 200,000 unique last hop addresses. `2001:1970:4000::/48` was the last hop prefix in traces to 1,008 distinct /48 target prefixes, the most extreme example of many target /48s mapping to a single last hop prefix.

Because a /48 prefix entirely subnetted into /52s should exhibit 16 distinct last hops, we choose $\eta = 16$ empirically as a baseline indication of more granular subnetting. The choice and sensitivity of η are discussed in detail in [30].

34,520 of the input 130,447 /48 target prefixes passed the η threshold in round one. Each of these /48 prefixes were then probed at a /60 granularity (4,096 probes to each /48). Edgy discovers significantly more unique non-aliased last hop addresses in this round, ∼12.2M, as the probing is focused on known address-producing target subnetworks identified in the first round.

To select target /48s for round three, we use $\eta = 256$ as an indicator of subnetting at a granularity finer than /56. 12,014 /48s meet this criteria, and were used as targets for probing at the /62 granularity (∼196.8M traces).

Round three, while probing $< 10\%$ of the input target seed prefixes, is focused on those with fine-grained subnetting and helps to expose subnetting strategies. As the IETF now discourages, but does not forbid, /64 or more-specific subnetting [25], we are interested in the prevalence of fine-grained subnetting, but must balance inferring this delegation behavior with probing load. Because subnetting generally occurs on nybble boundaries [25], by probing /62s, we are able to detect when target prefixes are subnetted beyond /60s, which is an indication that perhaps the operator is allocating /64 subnets. The /62 probing round produced ∼14.7M unique last hop addresses.

The final round is designed to enumerate last hop addresses for /64 subnets. Edgy selects any prefix with $\eta = 4$ prefix-unique last hops within a /60 (because we probe each /62, each /60 contains four targets). We surmise that four prefix-

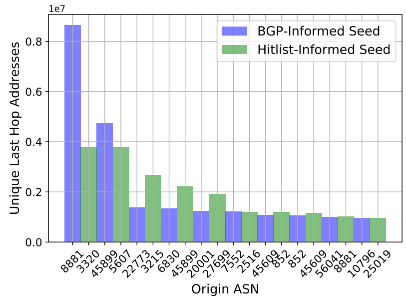

Fig. 5. Top 10 Last Hop ASN

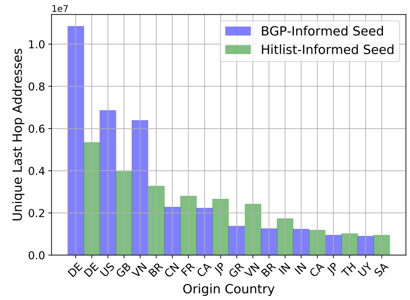

Fig. 6. Top 10 Last Hop Country

unique last hops is an indication that either the operator subnets at the /62 level, or is assigning /64 networks to their customers. The final /64 probing round discovered 15.3M distinct IPv6 addresses through exhaustive probing of 2,641 /48 target prefixes that met the η threshold to be in round four.

Cumulatively, edgy discovers more than 37M distinct IPv6 last hop addresses using the BGP-informed seed. Table 1 quantifies discovery across probing rounds. 3,989 ASs are represented in the last hop addresses, corresponding to 143 countries, as reported by Team Cymru's IP to ASN service [33]. Figures 5 and 6 summarize the ASes and countries that produced the largest number of periphery last hop addresses.

4.2 Hitlist-Informed Seed Results

We replicate the experiment described in §4.1 seeded with the hitlist-informed seed traces (from [7]). Table 1 shows the per-round results for both the BGP-informed and hitlist-informed seeds. Algorithm 1 on this input seed yielded 111,670 target /48 prefixes, about 20k fewer than the BGP-informed seed. However, the initial /56 probing round discovered nearly twice as many unique last hop addresses. The hitlist-informed seed led to almost double the number of target prefixes in the /60 round as compared to the BGP-informed seed, but discovered nearly 1M fewer last hops. As a result, only 4,462 /48 target prefixes were probed in the /62 probing round, discovering 5.4M last hops from 19,942 /48 prefixes. 1,531 target /48s were exhaustively probed at the /64 granularity in the fourth round, about 1% of the input hitlist seed prefixes. The /64 probing round discovered over 15M unique last hops, indicating that the 1,500 target /48s each contributed about 10,000 unique addresses on average. We attribute the differences between the BGP-informed and hitlist-informed seed data results to differences in how the original source data was collected. For example, the BGP-informed seed data was derived from a uniform sweep of the advertised IPv6 space, while the hitlist-informed seed data derived from a measurement campaign aimed at networks known to be dense in customers.

In total, periphery on the hitlist-informed seed discovers over 29M unique last hop router addresses. Nearly half of those addresses are found in the /64 probing round, during which edgy exhaustively probes all of the /64s in 1,531 /48 target prefixes. This suggests that a small number of prefixes have fine-grained subnetting, and that substantial periphery topology can be gained by probing a carefully selected set of target prefixes. Figures 5 and 6 display the top ten ASes and countries from which we obtain last hops; for the hitlist-informed seed, 141 countries and 3,578 ASNs contribute to the total.

4.3 Edginess Metrics

To better understand the extent to which edgy discovers IPv6 periphery infrastructure, we introduce three metrics of "edginess." The first coarse metric is simply the fraction of traces with a last hop within the same AS as the probe destination. Clearly, this condition does not imply that the last hop is truly an interface of the periphery router. However, it provides a rudimentary measure of whether traces are reaching the target network's AS. In contrast, a trace to a non-existent network will be dropped at an earlier hop in a default-free network.

We compare edgy's results against a day's worth of CAIDA's IPv6 Ark traceroute results from 105 different vantage points on Oct 1, 2019 [8]. Across nearly 17M traceroutes performed on that day, 1.7M (10%) produced a response from the target destination. However, of those 1.7M traceroutes that reached the destination, 86.2% were from probing the `::1` address, while 13.3% came from destinations known to be aliased, i.e., a fake reply. Unsurprisingly, fewer than 0.5% of the probes to random targets reached the destination.

40.2% of the CAIDA traces elicit a response from a last hop address that belongs to a BGP prefix originated by the same AS as the destination. In contrast, 87.1% of edgy's traces reach the target AS. While these results cannot be directly compared – edgy performs two orders of magnitude more traces than CAIDA; see §4.7 – it does demonstrate that the probing performed by edgy is in fact largely reaching the target network, if not the periphery.

Our second edginess metric is a more granular measure of how deep into the target network, and hence how close to the periphery, traces traverse. For each trace, we find the number of most significant bits (MSBs) that match between the target and the last hop response, i.e., the netmask of the most specific IPv6 prefix that encompasses the target and last hop. As before, this metric does not provide a definitive measure of reaching the periphery. Indeed, we empirically observe many networks that use very different IPv6 prefixes for the last hop point-to-point subnetwork as compared to the customer's prefix. However, the basis of this metric is that hierarchical routing implies more matching MSBs the closer the trace gets to the target.

Figure 7 shows the distribution of matching bits across the traceroutes from both CAIDA and edgy. Whereas the median size of the matching prefix is a /13 for CAIDA, it is nearly a /32 for edgy. The target and last hop share the same /48 for more than 5% of the edgy traces, but just 2% of the CAIDA traces. Thus, again, we see edgy's probing reaching more of the network periphery.

Finally, we quantify how many of our last hop addresses appear *only* as periphery addresses in our traces, and therefore do not appear as an intermediate hop in traceroutes to other target addresses. In the BGP-informed seed's first

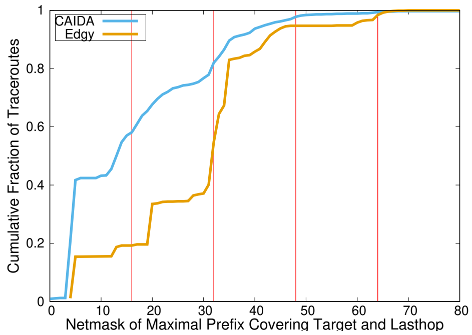

Fig. 7. Size of prefix encompassing both target and last hop IPv6 addresses

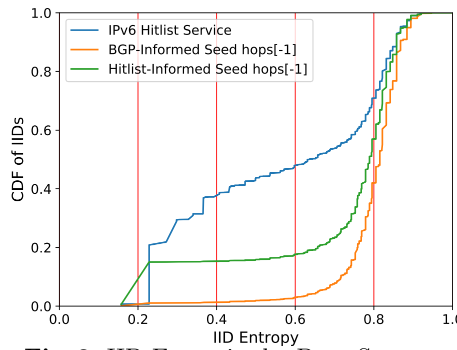

Fig. 8. IID Entropies by Data Source

round, 0.9% of discovered last hop addresses to a target appear as an intermediate hop to another target. In the second round, the same is true of 21% of last hops, 23% in the third round, and 4% in the fourth probing round. However, closer examination indicates that these numbers, particularly in the second and third round, are skewed by providers that frequently cycle periphery prefixes. For example, in the second round, 1.6M of the 2.5M addresses seen both as a last and an intermediate hop are located in ASN8881, which we observe cycling customer prefixes on a daily basis [30]. This often causes traces to appear to "bounce" between two (or more) different addresses toward the end of a trace. Sorting by the time the response was received shows that a single IPv6 address was responsible for high hop count responses until after a distinct point at which a second address becomes responsive. This erroneously causes the address that was not responsible for the highest hop count response to appear as if it were an intermediate hop for the target.

We also observe a second class of IPv6 address that appears both as a last hop and an intermediate hop to other targets. These addresses appear as the last hop for a large number of target networks that are most likely unallocated by the provider; these addresses typically have low entropy IID (e.g., ::1 or ::2) and are likely provider infrastructure. These last hop addresses also appear on the path to addresses that appear to be CPE, based on the high entropy or EUI-64 last hop returned when they are an intermediate hop.

4.4 Consolidated Results and Seed Data Comparison

Although both probing campaigns began with approximately the same number of target /48 prefixes in the first probing round (130,447 and 111,670 in the BGP and hitlist-informed seeds, respectively), only 9,684 /48s are common between the two data sets. The number of target prefixes in common decreases at each round, reaching 177 in /64 probing round. Only ∼1.6M (2.5%) last hop IPv6 addresses are present in both data sets. These results demonstrate edgy's sensitivity to seed input, and suggest that additional seed sources may aid discovery.

Of the top ten ASNs, only four are common between the two data sets – ASNs 852, 8881, 45899, and 45609. Of the top ten countries, however, six are common: Germany, Vietnam, Canada, Brazil, India, and Japan, with Germany ranking first in both. While the US is the second-leading producer of last hop addresses in the BGP-informed seed data with ∼6.9M unique last hops, it is fourteenth in the hitlist-informed data with only 357,877 addresses. Finally, we consider the last hop provider type using CAIDA's AS type classification [3]. By this classification, edgy's results come overwhelmingly from transit/access networks (99.9%) rather than content or enterprise ASes. This matches our intent for edgy to focus on IPv6 periphery discovery.

4.5 EUI-64 Addresses

Previous studies, e.g., [16,7] identified the presence of many EUI-64 addresses in IPv6 traceroutes, where the host identifier in the IPv6 address is a deterministic function of the interface's Media Access Control (MAC) address. Our study similarly found a significant fraction of EUI-64 addresses, despite the introduction of privacy extensions for Stateless Address Autoconfiguration (SLAAC) addresses in 2007 [24]. We discover slightly more than 16M EUI-64 last hop addresses, identifiable from the `ff:fe` at byte positions 4 and 5 in an IID, using the BGP-informed seed data, or approximately 42% of the total last hops. However, only 5.4M (34%) of the MAC addresses in these 16M last hops are unique.

The discrepancy between unique EUI-64 last hop addresses and MAC addresses appears to have two root causes. The first is delegated prefix rotation. Although 3.5M of the 5.4M unique MAC addresses observed appear in only one last hop address, 1.9M appear multiple times. Of these, the vast majority appear in only several addresses in the same /48, suggesting that the provider periodically rotates the remaining 16 bits of the network address portion [1,30,31]. We observe some providers rotating the prefix delegated to their customers on a daily basis, and further examination of forced prefix cycling is a topic of future work. The second cause behind the disparity between number of MAC addresses and EUI-64 last hop addresses is due to what we believe is MAC address reuse.

For instance, the MAC address `58:02:03:04:05:06` occurs in more than 266k BGP-informed seed last hop addresses in 76 /48s allocated to providers throughout Asia and Africa. Because our probing took place over a period of several weeks, we believe it is unlikely that a combination of provider prefix rotation and mobility substantially contributed to these; its simple incremental pattern in bytes 2 through 6 further suggest it is likely a hard-coded MAC address assigned to every model of a certain device. Support forums indicate that some models of Huawei LTE router [4,2] use `58:02:03:04:05` as an arbitrary MAC address for their LTE WAN interface.

4.6 Comparison to the IPv6 Hitlist Service

We compare our results to an open-source, frequently updated hitlist [16]. In mid-October 2019, the hitlist provides approximately 3.2M addresses responsive to ICMPv6, and TCP and UDP probes on ports 80 and 443.

Both the structure and magnitude of the addresses we discover differentiate our work from [16], which is unsurprising given our focus on finding addresses at the network periphery. Unlike our results, the addresses in the hitlist are less likely to be EUI-64 addresses. Only ∼441,000 EUI-64 addresses (with ∼338,000 unique MAC addresses) appear in the hitlist, representing approximately 14% of the total responsive addresses. Figure 8 plots the normalized Shannon entropies of the IIDs of addresses in our datasets compared with addresses in the IPv6 hitlist service. We see that the IPv6 hitlist contains a far greater proportion of low-entropy IIDs addresses than the last hop addresses edgy discovers. As periphery devices, particularly CPE in residential ISPs, are unlikely to be statically assigned a small constant IID and instead generate a high-entropy address via SLAAC, this reinforces edgy's discovery of a different portion of the IPv6 Internet than prior work. Further emphasizing the complementary nature of edgy's probing, only 0.2% of the addresses we discover appear in this hitlist, indicating that edgy discovers different topology. Finally, while the last hops edgy discovers overwhelmingly (99.9%) reside in access networks (§4.4), CAIDA's AS-type classifier categorizes 1.8M of the hitlist's IPv6 addresses as residing in access/transit networks, 1.2M in content networks, and 48k in enterprise networks.

4.7 Comparison with CAIDA IPv6 Topology Mapping

We again examine a day's worth of CAIDA's IPv6 Ark traceroute results from 105 different vantage points on Oct 1, 2019 [8], to understand edgy's complementary value. Because edgy sends nearly two orders of magnitude more probes (544M vs 8.5M), these are not directly comparable; however, we note that edgy discovers 64.8M non-aliased, routable last hop addresses that CAIDA does not. CAIDA finds 163,952 unique, non-aliased, routable last hop addresses. However, despite focusing on only target networks that are dense in last hops, edgy still discovers $\sim 25\%$ of the last hop addresses that CAIDA does. Edgy similarly finds 87.1M links to the last hop address that CAIDA does not, but discovers 54,024 of the 365,822 edges that contain only routable addresses from CAIDA's probing. Edgy's discovery of ∼37M unique periphery last hops from ∼544M targets probed in the BGP-informed seed yields 0.068 unique last hops per target, while the Ark traceroutes discover 0.019 unique last hops per target.

4.8 Comparison with Seed Data Source

Edgy, by design, extends topology discovery methodologies and is complementary to existing topology mapping campaigns. However, because we believe edgy provides increased address discovery over existing mapping systems, we compare the results obtained with edgy to the trace seeds used as input to edgy.

The BGP-informed seed source consists of traces conducted in August, 2018 to every /48 in the routed IPv6 Internet conducted from CAIDA's Archipelago [9]. These traces to ∼711M unique targets produce ∼5.8M unique last edges and ∼5.4M unique last hops after removing non-routable addresses. By contrast, edgy discovers ∼59.5M unique final edges and ∼37.1M unique IPv6 last hops by probing to ∼545M targets when seeded with the BGP-informed data. Thus, edgy significantly expands the discovered topology of an input seed.

Likewise, edgy discovers significantly more last hop addresses and edges than the hitlist-informed seed. The hitlist-informed seed discovers 434,560 unique last hops and 656,849 unique final edges, while edgy, informed by this data, discovers ∼29.2M unique last hops and ∼32.0M final edges.

5 Conclusions and Future Work

We introduce edgy, an algorithm to discover previously unknown portions of the IPv6 Internet, namely, the IPv6 periphery. Edgy extends and augments existing IPv6 discovery mapping systems, and the last hop periphery addresses that it discovers are nearly entirely disjoint from previous topology mapping campaigns. Because of privacy concerns involved with EUI-64 addresses and the ephemeral nature of many addresses, we are not releasing the periphery addresses edgy discovers at this time; however, we expect our results to be reproducible.

Several topics are planned for future work. First, we observe service providers that cycle their customers' periphery prefix periodically. This rotation leads to high levels of address discovery for these providers, but, based on examining IID reuse, over counts the number of actual device interfaces present. We plan to: i) discover which networks implement high-frequency prefix rotation; ii) quantify the rates at which new prefixes are issued; and iii) determine whether the prefix issuing mechanism is deterministic and predictable. Second, we discover large numbers of EUI-64 IPv6 addresses more than a decade after the introduction of SLAAC privacy extensions [24]. Because edgy discovers periphery devices like CPE, quantifying device types present in networks may be possible by cross-referencing the models providers issue to customers, and through correlation with protocols that leak model information [22]. Third, we wish to obtain more ground truth information on the IPv6 periphery as well as explicit validation of our results and algorithm. Fourth, we plan to improve edgy's efficiency by training it with historical data and leveraging multiple vantage points. For instance, periphery networks that exhibit frequent customer prefix cycling may need to be probed on a regular basis, while those with stable last hops may be re-probed infrequently. Finally, because of the ephemeral nature of some of the addresses we discover, we intend to couple other measurements tightly with address discovery. For example, to further elucidate these addresses' value, we will send ICMPv6 Echo Requests and capture service banners immediately after receiving probe responses.

Acknowledgments

We thank Jeremy Martin, Thomas Krenc, and Ricky Mok for early feedback, John Heidemann for shepherding, Mike Monahan and Will van Gulik for measurement infrastructure, and the anonymous reviewers for insightful critique. This work supported in part by NSF grant CNS-1855614. Views and conclusions are those of the authors and should not be interpreted as representing the official policies or position of the U.S. government or the NSF.

References

1. Zwangstrennung (forced ip address change) (2018), https://de.wikipedia.org/wiki/Zwangstrennung
2. Huawei lte cpe b315 (mts 8212ft) - discussion (2019), http://4pda.ru/forum/index.php?showtopic=700481&st=3580
3. The CAIDA UCSD AS Classification Dataset (2019), http://www.caida.org/data/as-classification
4. Speedport ii lte router status (2020), https://telekomhilft.telekom.de/riokc95758/attachments/riokc95758/552/327892/1/routerstatus.pdf
5. Berger, A., Weaver, N., Beverly, R., Campbell, L.: Internet Nameserver IPv4 and IPv6 Address Relationships. In: Proceedings of ACM Internet Measurement Conference (IMC) (2013)
6. Beverly, R.: Yarrp'ing the Internet: Randomized High-Speed Active Topology Discovery. In: Proceedings of ACM Internet Measurement Conference (IMC) (Nov 2016)
7. Beverly, R., Durairajan, R., Plonka, D., Rohrer, J.P.: In the IP of the Beholder: Strategies for Active IPv6 Topology Discovery. In: Proceedings of ACM Internet Measurement Conference (IMC) (Nov 2018)
8. CAIDA: The CAIDA UCSD IPv6 Topology Dataset (2018), http://www.caida.org/data/active/ipv6_allpref_topology_dataset.xml
9. CAIDA: The CAIDA UCSD IPv6 Routed /48 Topology Dataset (2019), https://www.caida.org/data/active/ipv6_routed_48_topology_dataset.xml
10. Chittimaneni, K., Chown, T., Howard, L., Kuarsingh, V., Pouffary, Y., Vyncke, E.: Enterprise IPv6 Deployment Guidelines. RFC 7381 (Informational) (Oct 2014), https://www.rfc-editor.org/rfc/rfc7381.txt
11. Czyz, J., Luckie, M., Allman, M., Bailey, M.: Don't Forget to Lock the Back Door! A Characterization of IPv6 Network Security Policy. In: Network and Distributed Systems Security (NDSS) (2016)
12. Czyz, J., Allman, M., Zhang, J., Iekel-Johnson, S., Osterweil, E., Bailey, M.: Measuring IPv6 Adoption. SIGCOMM Comput. Commun. Rev. **44**(4) (Aug 2014)
13. Dhamdhere, A., Luckie, M., Huffaker, B., claffy, k., Elmokashfi, A., Aben, E.: Measuring the Deployment of IPv6: Topology, Routing and Performance. In: Proceedings of ACM Internet Measurement Conference (IMC) (2012)
14. Fan, X., Heidemann, J.: Selecting Representative IP Addresses for Internet Topology Studies. In: Proceedings of ACM Internet Measurement Conference (IMC) (2010)
15. Foremski, P., Plonka, D., Berger, A.: Entropy/IP: Uncovering Structure in IPv6 Addresses. In: Proceedings of ACM Internet Measurement Conference (IMC) (2016)

16. Gasser, O., Scheitle, Q., Foremski, P., Lone, Q., Korczyński, M., Strowes, S.D., Hendriks, L., Carle, G.: Clusters in the Expanse: Understanding and Unbiasing IPv6 Hitlists. In: Proceedings of ACM Internet Measurement Conference (IMC) (2018)
17. Gont, F., Chown, T.: Network Reconnaissance in IPv6 Networks. RFC 7707 (Informational) (Mar 2016), http://www.ietf.org/rfc/rfc7707.txt
18. Hyun, Y., k. claffy: Archipelago measurement infrastructure (2018), http://www.caida.org/projects/ark/
19. IAB, IESG: Recommendations on IPv6 Address Allocations to Sites. RFC 3177 (Informational) (Sep 2001), http://www.ietf.org/rfc/rfc3177.txt
20. Livadariu, I., Ferlin, S., Alay, Ö., Dreibholz, T., Dhamdhere, A., Elmokashfi, A.: Leveraging the IPv4/IPv6 identity duality by using multi-path transport. In: 2015 IEEE Conference on Computer Communications Workshops (2015)
21. Luckie, M., Beverly, R.: The Impact of Router Outages on the AS-level Internet. In: Proceedings of ACM SIGCOMM (2017)
22. Martin, J., Rye, E.C., Beverly, R.: Decomposition of MAC Address Structure for Granular Device Inference . In: Proceedings of the Annual Computer Security Applications Conference (ACSAC) (Dec 2016)
23. Murdock, A., Li, F., Bramsen, P., Durumeric, Z., Paxson, V.: Target Generation for Internet-wide IPv6 Scanning. In: Proceedings of ACM Internet Measurement Conference (IMC) (2017)
24. Narten, T., Draves, R., Krishnan, S.: Privacy Extensions for Stateless Address Autoconfiguration in IPv6. RFC 4941 (Sep 2007), http://www.ietf.org/rfc/rfc4941.txt
25. Narten, T., Huston, G., Roberts, L.: IPv6 Address Assignment to End Sites. RFC 6177 (Best Current Practice) (Mar 2011), http://www.ietf.org/rfc/rfc6177.txt
26. Plonka, D., Berger, A.: Temporal and Spatial Classification of Active IPv6 Addresses. In: Proceedings of ACM Internet Measurement Conference (IMC) (2015)
27. Pujol, E., Richter, P., Feldmann, A.: Understanding the share of IPv6 traffic in a dual-stack ISP. In: Passive and Active Measurement (PAM) (2017)
28. RIPE: Best Current Operational Practice for Operators: IPv6 Prefix Assignment for End-Users - Persistent vs Non-Persistent, and What Size to Choose (2017), https://www.ripe.net/publications/docs/ripe-690
29. Rohrer, J.P., LaFever, B., Beverly, R.: Empirical Study of Router IPv6 Interface Address Distributions. IEEE Internet Computing (Aug 2016)
30. Rye, E.C., Beverly, R.: Discovering the IPv6 Network Periphery (2020), https://arxiv.org/abs/2001.08684
31. Rye, E.C., Martin, J., Beverly, R.: EUI-64 Considered Harmful (2019), https://arxiv.org/pdf/1902.08968.pdf
32. Srisuresh, P., Holdrege, M.: IP Network Address Translator (NAT) Terminology and Considerations. RFC 2663 (Informational) (Aug 1999), http://www.ietf.org/rfc/rfc2663.txt
33. Team Cymru: IP to ASN mapping (2019), https://www.team-cymru.org/IP-ASN-mapping.html
34. Zander, S., Wang, X.: Are We There Yet? IPv6 in Australia and China. ACM Trans. Internet Technol. **18**(3) (Feb 2018)

Appendix A Algorithm Details

Algorithm 1 Discover_Init($seed\_traces$)

```
density = []
targets = []
for (hops, dst) ∈ seed_traces do
    dst48 ← dst & (2^48 − 1 ≫ 80)
    LH ← hops[−1]
    density[LH] ← density[LH] ∪ dst48
for LH ∈ density do
    if |LH| = 1 then
        targets ← density[LH]
for prefix ∈ targets do
    Discover(prefix)
```

Algorithm 2 Discover($prefix$)

```
masks = {56, 60, 62, 64}
LH ← {}
t ← rand(0, 2^64)
for n ∈ masks do
    for i ← {0 . . . 2^(n−48) − 1} do
        hops ← yarrp(prefix + (i ≪ (128 − n)) + t)
        LH ← hops[−1]
    if |LH| ≤ η or n = 64 then
        break
```