IEEE INFOCOM 2025 - IEEE Conference on Computer Communications | 979-8-3315-4305-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/INFOCOM55648.2025.11044733

Pruning as Scanning: Towards Internet-wide IPv6 Network Periphery Discovery

Tao Yang, Ling Hu, Bingnan Hou, Zhenzhong Yang and Zhiping Cai
College of Computer Science and Technology, National University of Defense Technology, China
E-mails: {*yangtao97, linghu50, houbingnan19, zzy.nudt, zpcai*}*@nudt.edu.cn*

***Abstract*—IPv6 network peripheries serve as the last-hop routing devices connecting customers, making the rapid discovery of periphery addresses crucial for Internet security and network measurement. Traditional IPv6 scanning methods, based on target generation algorithms (TGAs), often fall short due to mismatches between the patterns identified by TGAs and those inherent in IPv6 periphery addresses. To address this, we introduce Pruning-as-Scanning (PaS), a novel IPv6 scanning approach designed for fast global IPv6 periphery address discovery. Unlike previous methods, PaS initiates scanning from the entire IPv6 address space without relying on seed datasets. It systematically probes IPv6 Internet prefixes from the shortest to the longest, similar to a hierarchical search, while discarding those irrelevant prefixes (and sub-prefixes), i.e., space pruning. This reduces the complexity of searching the entire IPv6 address space, allowing comprehensive measurement of IPv6 endpoint subnets where periphery devices are installed. We implemented an asynchronous scanner prototype based on this principle. Real-world tests demonstrated that PaS outperformed existing methods in both probing scale and the number of identified periphery addresses. Our approach achieved the first comprehensive, global-scale IPv6 network periphery discovery, uncovering over 712 million IPv6 periphery addresses using 33 billion probe packets, averaging 4 million addresses per hour at an uplink of 50 kpps, significantly expanding the IPv6 address corpus.**

***Index Terms*—IPv6, Internet-wide Scanning, Network Measurement, Network Security**

I. Introduction

IPv6 network peripheries are the last-hop routing devices connecting end-user subnets [1], [2], including various network entities. They can encompass smartphones, tablets, laptops, and other cellular network wireless devices, while *accessing the IPv6 Internet via LTE or 4G wireless connections where their 3GPP interfaces are extended to a LAN link*. Upon connecting to a wireless base station, those cellular hosts will shift from *host-only* to *router-and-host* mode, thereby enabling routing functionality [3], [4] Additionally, IPv6 peripheries can also be the modems, routers, switches, wireless access points, set-top boxes, VoIP phones, IoT devices, and other gateway equipments that are installed at a customer's location [5], [6]

IPv6 peripheries play a vital role in the architecture of the IPv6 Internet and hosts the essential functions for maintaining network operations, such as packet forwarding, traffic filtering [7], and system provisioning [5]. As a result, efficient discovery of the IPv6 periphery addresses has been proved to deliver significant benefits to numerous research areas. These areas include but are not limited to, network asset censuses [8], privacy leakage detection [9], IP address geolocation [10], cybersecurity intelligence reconnaissance [11], [12], and measuring remote networks through side-channel methods [13]. As such, there is a compelling incentive to advance research efforts in the IPv6 periphery discovery. However, the definitive truths in IPv6 networks, as discussed below, makes this task complicated.

Traditional TGA-based approaches cannot be compatible. Over the past decade, numerous remarkable works have been delicated in IPv6 target generation algorithms (TGAs) for searching the active addresses [14]–[21] Nevertheless, when applied to the IPv6 periphery discovery, those traditional TGA-based approaches could commonly encounter challenges. The fundamental issue lies in the distinct addressing patterns of IPv6 peripheries while compared to the clustered distributions of active IPv6 addresses assumed by TGAs, as shown in Fig. 1. In short, TGA-based approaches consider the addresses similar to a known active address (e.g., with merely few discrepancies of nibbles) more likely to respond when probed. We will show that IPv6 periphery addresses prevalently exhibit "irregular" last 64 bits (namely Interface Identifiers), and discovering these IPv6 periphery addresses can significantly enrich the IPv6 address corpus (see § V-D). These discrepancies of address patterns render traditional TGA-based approaches, which rely on generating potential targets from known active addresses (seeds), less effective for IPv6 periphery discovery.

Existing topology-based approaches lack scalability. Considering that IPv6 peripheries constitute the last-hop routing connections to end hosts, some researchers try to locate periphery addresses based on network topology insights [1], [2]. Specifically, they utilized the scanning tools like ZMap [22] or Yarrp [23] for exhaustively dispatching probe packets towards endpoint subnets, so as to provoking the indirect responses from the periphery device maintaining those subnets. From these responses, researchers can then extract the periphery addresses. However, the vast IPv6 address space, which encompasses 2^{128} unique addresses and 2^{64} endpoint subnets of the longest prefix /64 [24], [25]. The computational resource required to scan each possible subnet is prohibitive, potentially spanning millions of years at an uplink of 100,000 packets per second.

At first glance, we can neither employ the seed addresses to narrow the search scope like the TGA-based approaches, nor afford the time and bandwidth resources consumed by brute-force scanning strategies. This raises the question: *Does conducting the Internet-wide IPv6 periphery discovery hold* *viability?* Our answer is positive.

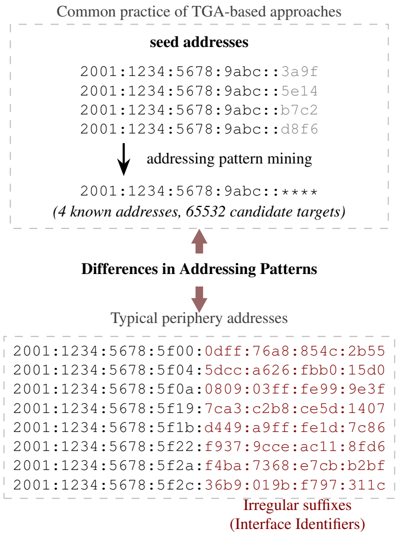
Common practice of TGA-based approaches
seed addresses
2001:1234:5678:9abc::3a9f
2001:1234:5678:9abc::5e14
2001:1234:5678:9abc::b7c2
2001:1234:5678:9abc::d8f6
addressing pattern mining
2001:1234:5678:9abc::****
(4 known addresses, 65532 candidate targets)
Differences in Addressing Patterns
Typical periphery addresses
2001:1234:5678:5f00:0dff:76a8:854c:2b55
2001:1234:5678:5f04:5dcc:a626:fbb0:15d0
2001:1234:5678:5f0a:0809:03ff:fe99:9e3f
2001:1234:5678:5f19:7ca3:c2b8:ce5d:1407
2001:1234:5678:5f1b:d449:a9ff:fe1d:7c86
2001:1234:5678:5f22:f937:9cce:ac11:8fd6
2001:1234:5678:5f2a:f4ba:7368:e7cb:b2bf
2001:1234:5678:5f2c:36b9:019b:f797:311c
Irregular suffixes
(Interface Identifiers)

Fig. 1: Irregular address patterns of IPv6 peripheries are incompatible the common practices of TGA-based approaches.

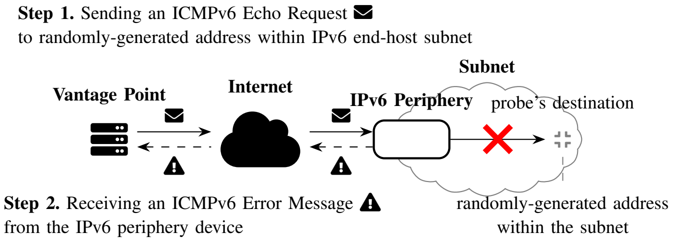
Step 1. Sending an ICMPv6 Echo Request
to randomly-generated address within IPv6 end-host subnet
Subnet
Vantage Point
Internet
IPv6 Periphery
probe's destination
Step 2. Receiving an ICMPv6 Error Message
from the IPv6 periphery device
randomly-generated address
within the subnet

Fig. 2: Process of IPv6 periphery discovery. The prober at the *vantage point* sends an ICMP Echo Request to a randomly-generated target within the endpoint subnet. As the IPv6 periphery device cannot forward the probe packet to its intended destination, it drops the packet and responds with an ICMPv6 error message to vantage point, thus exposing its own address.

To this end, we propose Pruning-as-Scanning (abbr., PaS), an innovative approach to comprehensively discover the IPv6 network peripheries on a global scale. At its core, PaS simply needs to send packets to random targets in end-point subnets, observe the IPv6 periphery addresses (sources) by receiving indirectly responding packets as shown in Fig. 2 - and no more than that. However, the key challenge solved by PaS is how to effectively dispatch the probes to those subnets across the vast address space (nearly 2^{128}). We will show that endpoint subnets are distributed very sparsely, and a considerable amount of IPv6 address space remains unallocated to end-users (see § V-A), rendering attempts at probing these unused networks in vain.

The implementation of PaS is straightforward and intuitive: step-by-step examination of the networks from short IPv6 prefixes (e.g., every /32s) to long IPv6 prefixes (e.g., every /64s), while simultaneously discarding those network prefixes (and subprefixes) which remain unallocated to end-users. This pruning strategy can conserve a significant amount of probing resources, thus making it feasible to exhaustively probe all the IPv6 endpoint subnets with negligible dead ends. To achieve appropriate pruning, PaS creatively introduces the concept of the "fingerprint of prefix". See § III-C for how PaS uses the "fingerprint of prefix" as an indicator to reduce the search scope of Internet-wide IPv6 periphery discovery, and therby make the impossible possible.

The main contributions of the paper could be summarized as follows:

- We introduce *fingerprint of prefix*, a novel metric for ascertaining the status of an IPv6 prefix, specifically whether a given IPv6 prefix is allocated to end-user networks or remains unused. Additionally, we offer a theoretical analysis demonstrating the low likelihood of this metric being incorrect.
- We propose Pruning-as-Scanning (PaS), a new IPv6 scanning technique for periphery discovery. By taking our *fingerprint of prefix* as pruning indicator, PaS can exhaustively probe all the IPv6 endpoint subnets with minimal omissions, thus enabling it to enumerate the IPv6 network peripheries across the world. To the best of our knowledge, PaS is the first to accomplish this task.
- Using PaS's principle, we developed an asynchronous IPv6 scanner prototype that can rotate the header fields of each IPv6 probe packet and disperse probing traffic.
- We performed a thorough Internet-wide active measurement of IPv6 peripheries, using 33 billion probe packets over a 10-day duration. As a result, PaS discovered 712 million unique addresses, significantly expanding the IPv6 address pool. Comparison results demonstrated that PaS outperformed all existing approaches in both the breadth of its probing scale (from limited to global) and the number of identified periphery addresses (from seed-assisted 64M to self-supported 712M).

II. Background and Related Work

A. Addressing and Routing in IPv6

IPv6 addresses are 128 bits. Each consecutive 4 bits have a hexadecimal representation called a nibble. Within an IPv6 unicast address, the 128 bits are conceptually segmented into two parts: the upper 64 bits are utilized for routing, namely *network prefix*, while the lower 64 bits describe a specific host, namely *interface identifier* (IID).

In IPv6, any network can be represented into a "prefix". The text representation of IPv6 address prefixes is similar to the way IPv4 address prefixes are written in Classless Inter-Domain Routing (CIDR) notation [24]. For example, the 48-bit prefix `200112345678` (hexadecimal) can be written into `2001:1234:5678::/48`.

All IPv6 routing protocols are designed to support prefixes of any length. Hardware and software of routing and forwarding should impose no rules, but implement "longest-match-first" on prefixes of any valid length in its Forwarding Information Base (FIB) [26]. This clearly dictates:

Observation 1: *IPv6 packets destined for any addresses within a same prefix will be routed to the same next-hop router*[1].

The addressing boundary of end-users subnets has been defined as `/64`, due to the necessity of reserving a 64-bit capacity for Interface Identifiers (IIDs). Thus, this explicitly indicates:

Observation 2: *A single probe is sufficient for each network of a /64 prefix as probing multiple sub-prefixes within the same /64 is highly unlikely to yield different periphery devices.*

It's worth noting that, although probing a /64 network's periphery seems challenging without pre-known active addresses to trace the last-hop to the destination, the implementation in our measurement is straightforward, as illustrated in Fig. 2: Since any destinations within a /64 will be directed to its periphery device (if one exists), we simply need to generate targets within a /64 with randomized IIDs and send a probe destined for these targets to accomplish probing of this end-user subnet. Because, the periphery device cannot route the probe packet to its destination, which is almost certainly inactive[2]. As a result, periphery device will respond with ICMPv6 error messages[3], inadvertently exposing its addresses [27].

B. IPv6 Target Generation

Due to the vast address space, IPv6 scanning cannot be realized in a brute-force manner like IPv4 scanning. Currently, IPv6 scanning solutions commonly focus on target generation approaches that use known IPv6 addresses to deduce new active addresses.

Ullrich et al. [14] introduced a pattern-based recursive algorithm designed to incrementally add more seeds for scanning with each iteration across a flexible address range. Entropy/IP [15] utilizes the structural information of seeds to create target addresses for scanning. 6Gen [16] employs an agglomerative hierarchical clustering (AHC) algorithm, using each seed as a cluster center to create target addresses while ensuring maximum seed density and minimum scale. Conversely, 6Tree [17] adopts a divisive hierarchical clustering (DHC) algorithm to construct a spatial tree based on the seeds' structure, which segments the IPv6 address space into manageable nodes/subspaces. 6Gen and 6Tree then scan target addresses within each node for discovery of active IPv6 addresses. Based on the above insights, numerous machine learning models have been trained on seed addresses in order to generate candidate addresses for active measurements, using reinforcement learning [19], generative adversarial networks [28], graph theory [29], ensemble learning [20], and bidirect hierarchical clustering [21].

It can be easily summed up that these TGA approaches share a fundamental process of address pattern mining, which implicitly introduces the assumption: targets sharing the same address pattern as known active IPv6 addresses have a higher likelihood of responding to probes. However, this hypothesis is challenged by the significant distinctions among IPv6 periphery addresses which we will show.

C. IPv6 Periphery Discovery

IPv6 periphery devices are the last-hop routing infrastructure connecting end-users. To discover these devices and their addresses, Edgy [1] traces one random target within specific prefixes, such as /56 or /60, to identify the last hops in the route path. Li et al. [2] explored IPv6 periphery discovery by leveraging the behavior of some routers to respond to packets addressed to non-existent devices within their subnet with ICMP Destination Unreachable messages. Their tool, XMap, dispatches probes to each *constituent* subnet of an IPv6 network (e.g., every /64 within a given /32 network), in order to extract the last-hop in-path router from these ICMP packets

These methods prove effective in suitable contexts, such as when the probing scope is confined (e.g., within a single /32 address block) or when the complete range of prefixes delegated to end-user subnets is known. However, when extending these practices of periphery discovery to broader scales, such as the entire Internet, their strategies of exhaustively probing every *potential* subnet faces significant obstacles due to the immense breadth of the IPv6 address space. Hence, we are thus motivated to solve this challenge.

III. Pruning-as-Scanning

Pruning-as-Scanning (PaS) aims to provoke indirect responses from last-hop IPv6 periphery devices for given prefix networks. Its core idea is to gradually identify the unused prefixes and incrementally focus the probing efforts on those prefixes assigned to end-user networks. As shown in Fig. 3, its workflow can be divided into an initialization stage, followed by active probing that proceeds in rounds. Results from one round of probing are used to tune the probing scope of subsequent rounds, a process called *Pruning*.

A. Initialization

To ensure its generalizability, Pruning-as-Scanning operates without the additional prerequisites required by other approaches, such as seed information.

Initially, PaS employs the RIPEstat data API [30] to retrieve all prefixes that IANA and RIRs have allocated and assigned directly since January 1, 1970 (Unix Time 0). They encompasses all the IPv6 addresses potentially used by now, denoted to *IPv6 address space*, which is utilized for excluding improbable addresses so as to preliminarily reduce the search scope (**Round 0**). This data are readily available, making the experiments of PaS easily replicated by anyone.

B. Address Space Pruning

After initialization stage, the remaining address space is still prohibitively vast, encompassing over 2^{115} individual addresses as will be shown in Fig. 4. Even though we need to probe merely once per /64 using the targets with randomized IIDs (see § II-A and Fig. 2), the search scope among the entire *address space* is still spanned to a total of 2^{51} potential targets ($2^{51} = \frac{2^{111}}{2^{64}}$) at the time of writing this paper.

[1] More precisely, a group of next-hop routers, due to the load balancing.

[2] Probability of hitting an active addresses is negligible in a /64.

[3] Consistently referred to as *indirect responses* throughout the paper.

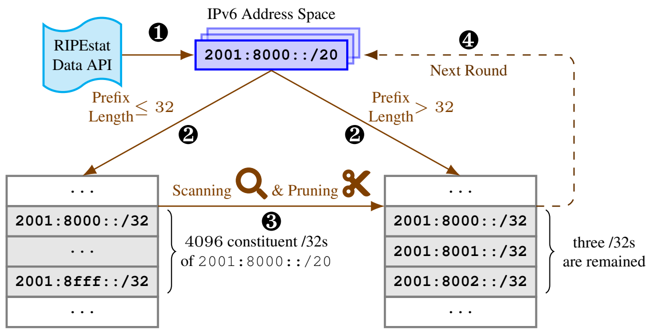

Fig. 3: The workflow of Pruning-as-Scanning is illustrated by **Round 0** (❶), and **Round 1** (❷ & ❸). Note that the prefix 2001:8000::/20 initially encompasses $2^{12} = 4096$ constituent /32s, but after applying the pruning, only three /32s remain. Subsequent rounds repeat the steps ❷ to ❹ until completion.

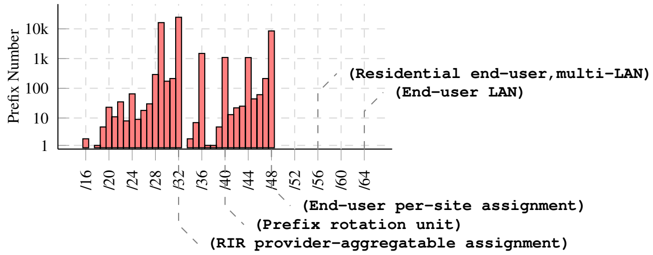

Fig. 4: So far, IANA and RIRs have allocated or directly assigned 54,161 IPv6 prefixes, ranging from /16 to /48 (after aggregation), spanning roughly 2^{115} individual IPv6 addresses.

Given the massive size of the IPv6 address space, PaS will not probe each /64 subnet in a brute-force manner. Rather, it examines the constituent prefixes of the networks from large to small, adhering to the hierarchical addressing principle of IPv6 [24]. Specifically, PaS will perform active probing iteratively. In each round, all the constituent prefixes at a specific length (e.g., /32, /40, and so on) within the search scope of that round will be systematically scanned. Then, the probing results can be used, on one hand, to extract the desired IPv6 periphery addresses from the indirect responses; on the other hand, they will be utilized to assess the necessity for further scanning of the specific prefixes at finer granularity, thereby reducing the search scope of the next-round probing.

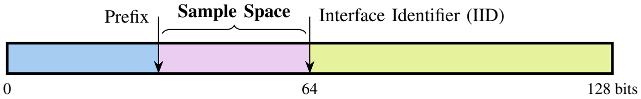

Fig. 5: For a probed prefix, its PaS destinations consist of: the fixed prefix, the randomized IID, and the middle bits used for sampling.

To facilitate the understanding of above process, we take the practical steps in our experiments for illustration:

- **Round 1.** Given with n_1 /32 networks in the current search scope, PaS conducts comprehensive probing of all /32s and reserves longer prefixes for subsequent rounds. For each /32, we randomly select ρ_1 /64s and append random Interface Identifiers (IIDs) to generate $n_1 \times \rho_1 \times 1$ target addresses, as shown in Fig. 5. We employ a ZMap-variant scanner incorporating our unique modifications (see § IV), to probe each target with the maximum Hop Limit (255) and gather responses.
  Why /32? It corresponds to the universally standard of Provider Aggregatable (PA) address blocks assigned by Regional Internet registries (RIRs) [31], for the purpose of effective aggregation and routing, thereby minimizing the size of the global internet routing tables and enhancing router performance, as shown in Fig. 4.
- **Round 2.** Using results of prior round, we eliminate those /32s that do not necessitate the further probing (a.k.a, redundant /32 prefixes) - a process called *pruning*. The remaining /32s, along with the prefixes longer than /32 in the IPv6 addres space, form the next search scope. Assuming n_2 /40s in the defined scope, we again sample ρ_2 /64s with appending random IIDs to generate $n_2 \times \rho_2 \times 1$ targets for probing as same as prior round.
  Why /40? Existing studies have shown that the domestic networks usually allocate subscriber addresses from a typical /40 block pool for prefix rotation [32]. This demonstrates that sub-prefix assignments for end-users often originate from the same /40 block (see Fig. 4).
- **Round 3.** Likewise, the redundant /40s would be discarded, while the remaining /40s and longer prefixes further narrow the search space. given n_3 /48 network in the next search space, we generate the random-IID targets with ρ_3 randomly-selected /64s for probing, and then receive the indirect responses.
  Why /48? It is the shortest prefix that can be allocated to end-sites [33], the locations where users and devices establish direct connections. Ordinarily, an IPv6 periphery devices will not maintained a endpoint subnet exceeding /48.
- **Round 4.** Similarly, we would drop out those /48s identified as redundancy, and take the remaining /48s (w.r.t., totaling n_4 /56 networks) as the inputs for the next round. Each /56 network will be probed with ρ_4 random IID probes, consuming a total of $n_4 \times \rho_4$ probing packets.
  Why /56? A typical residential customer should ideally receive at least one /56 or /64 prefix from their ISP for optimal network performance and address allocation [33].
- **Round 5.** At last, after elimination of irrelevant /56 prefixes, the final search space consists of the remaining /56s. Here, we probe random-IID targets for their constituent /64 networks, totaling n_5, and capture the indirect responses.

The described method cumulatively consumed a sum of $n_1 \times \rho_1 + n_2 \times \rho_2 + n_3 \times \rho_3 + n_4 \times \rho_4 + n_5 \times 1$ probes, far less than the prohibitive resource demand considered initially. Accordingly, we will further prove that this comprehensive IPv6 scanning, even without initial seed data like prior works [1], [2], is both economically viable and can be completed within days.

However, the pivotal aspect of implementing the aforementioned strategy lies in accurately assessing whether a prefix is redundant - the exact issue we aim to tackle in the § III-C.

C. Fingerprint of Prefix

The core idea of PaS for support the search space pruning is the fingerprints of prefixes.

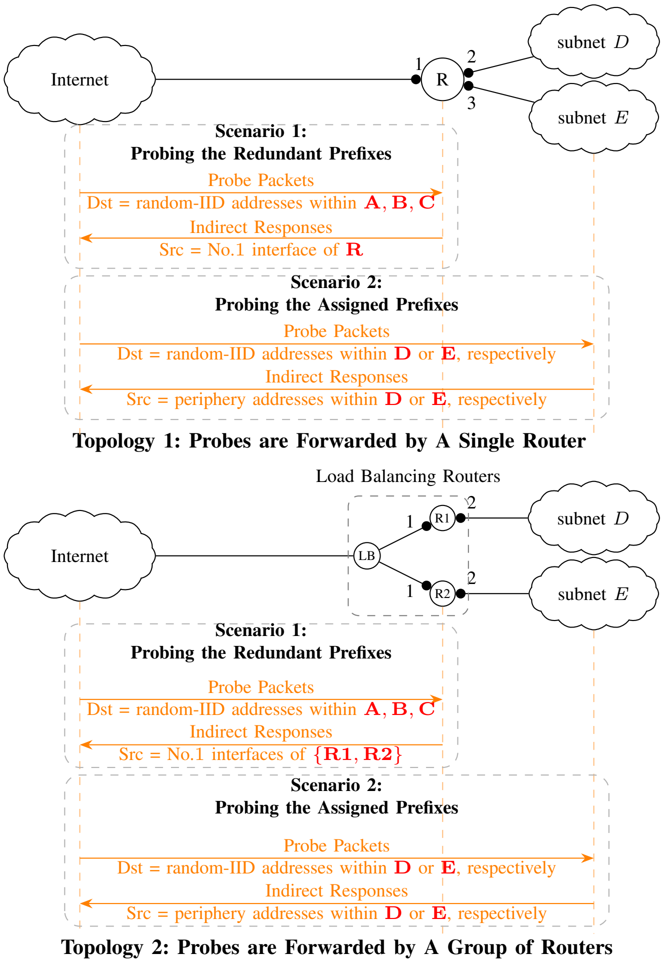
Internet
R
1
2
3
subnet D
subnet E
Scenario 1:
Probing the Redundant Prefixes
Probe Packets
Dst = random-IID addresses within **A**, **B**, **C**
Indirect Responses
Src = No.1 interface of **R**
Scenario 2:
Probing the Assigned Prefixes
Probe Packets
Dst = random-IID addresses within **D** or **E**, respectively
Indirect Responses
Src = periphery addresses within **D** or **E**, respectively
Topology 1: Probes are Forwarded by A Single Router
Load Balancing Routers
Internet
LB
R1
R2
1
2
1
2
subnet D
subnet E
Scenario 1:
Probing the Redundant Prefixes
Probe Packets
Dst = random-IID addresses within **A**, **B**, **C**
Indirect Responses
Src = No.1 interfaces of {**R1**, **R2**}
Scenario 2:
Probing the Assigned Prefixes
Probe Packets
Dst = random-IID addresses within **D** or **E**, respectively
Indirect Responses
Src = periphery addresses within **D** or **E**, respectively
Topology 2: Probes are Forwarded by A Group of Routers

Fig. 6: Two typical topologies illustrate the forwarding process of PaS's probe packets to destination subnets in IPv6 Internet.

To facilitate the illustration of this concept, let's consider a typical IPv6 forwarding model, i.e., the **topology 1** in Fig. 6. This topology comprises the global Internet, two distinct subnets (denoted to D and E), and the routing device that establishes the connectivity between them: we assume that, without loss of generality, the probes destined for networks/prefixes A, B, C, D, and E were routed to router R (or alternatively, to the router group $\{R1, R2\}$ caused by the load balancer illustrated by **topology 2** in Fig. 6) via the Internet. These routers can forward the corresponding probe packets to the subnets of D and E. However, The probe packets destinated for networks A, B, C cannot be routed because their prefixes are not yet assigned to the real-world subnets. As a result, the vantage point of our PaS will receive the indirect responses from the perphery devices within subnets D and E, as well as from No.1 interfaces of those routers [27].

Intuitively, subsequent probing efforts should focus on networks/prefixes D and E, given that they are already assigned the real-world subnets where the desired IPv6 peripheries likely reside. In contrast, persistently probing the prefixes A, B and C would be futile. We are thus motivated to remove those redundant prefixes early.

It can be readily noticed that probing the redundant prefixes (e.g., A, B, C) will merely elicit responses from the intermediate router R or router group $\{R1, R2\}$, making the indirect responses homogeneous. That is, these responses typically share the identical source addresses (w.r.t., **topology 1**) or groups of source addresses (w.r.t., **topology 2**). Conversely, for a prefix assigned to the real-world subnet, its indirect responses would showcase the source addresses significantly different from those of other prefixes. In essence, the collection of source addresses in indirect responses from sufficient probe packets can act as a distinctive identifier for a prefix, or in a simpler term, its *fingerprint*. Therefore, the basic insight utilized by PaS for removal of redundant prefixes is shown as follows:

Observation 3: *Prefixes that share the identical fingerprint with others should be considered redundant, thereby excluded from the subsequent probing.*

Accordingly, the algorithm for the process of eliminating redundant prefixes can be strategically summarized in Alg. 1. *In a nutshell*, prefixes displaying unique fingerprints warrant the next-round probing. For example, the fingerprints of prefixes D and E each occur only once throughout, whereas A, B, and C share an identical fingerprint and are therefore removed.

Algorithm 1 Redundant Prefix Elimination (Pruning)

```
Require: $M$, the $\langle$probing prefix, source addresses$\rangle$ pairs of
    indirect responses; $P$, set of probing prefixes.
Ensure: prefixes $N$ remained for next-round probing
initialize the fingerprint $f_p \leftarrow \emptyset, \forall p \in P$
for $p, s \in M$ do
    $f_p \leftarrow f_p \cup \{s\}$ # building fingerprint for each prefix
end for
initialize the frequency $f_p.freq \leftarrow 0, \forall p \in P$
for $p \in P$ do
    $f_p.freq \leftarrow f_p.freq + 1$ # calculating the frequency for
    each fingerprint
end for
$N = \emptyset$
for $p \in P$ do
    if $f_p.freq = 1$ then
        $N \leftarrow N \cup \{p\}$ # remaining the unique-fingerprint prefixes
    end if
end for
```

D. How many probes are sufficient?

A critical concern that remains unsolved is *how many probes are sufficient for probing a prefix?* Thoroughly examining each /64 subnet within a prefix can indeed provide answers, but this approach would not lead to any reduction in probing redundancy. Rather, we opt to sample a few /64 subnets within each prefix in order to infer its fingerprint. The following method is utilized to determine the quantity of probing packets sent to a prefix, balancing accuracy and efficiency.

To prevent wasteful practices on IPv6 address resources, the prefixes in use should have high HD-ratio[4] [34]. For the prefix of a length $56 - l$ (e.g., $l = 0$ for a /56) and a HD-ratio γ, we can calculate the number of /56s (efficiency measurement unit) using $n = 2^{\gamma l}$, where $\gamma = \frac{\log n}{\log 2^l} \leq 1$. The likelihood that an arbitrarily-selected address falls within these assigned endpoint subnets is denoted by $\sigma = \frac{n}{2^l} = \frac{1}{2^{(1-\gamma)l}}$. Should we aim to randomly sample k targets to deduce the fingerprint of this prefix, there's a worst-case scenario where all our probes are dispatched to the unassigned portion of the address space. The probability of this happening is given by $(1-\sigma)^k$, which needs to be less than a predetermined threshold: $(1-\sigma)^k < e^{-\theta}$. Thus, the number of probes k must satisfy $k > \frac{-\theta}{\ln(1-\sigma)}$. Direct computation here poses a challenge, thus we approximate the $\ln(1-\sigma)$ using $\ln(1-\sigma) = -\sum_{i=1}^{\infty} \frac{x^i}{i}$, leading to $k > \frac{\theta}{\sigma + \frac{\sigma^2}{2} + \frac{\sigma^3}{3} + \cdots}$. For simplification, we adopt $k = \frac{\theta}{\sigma}$ to meet above requirement, thus deriving that $k = \theta(2^{(1-\gamma)})^l$.

According to standard for IPv6 address allocation where the HD Ratio threshold based on /56 assignments shifts from 0.8 to 0.94 [31], we have conservatively set γ at 0.75 to ensure sufficient probing. Details on the probe quantity for scanning a specific prefix size are provided in Tab. I. Furthermore, We will demonstrate that this probe allocation strategy effectively balances efficiency with reduced probing omissions (see § V-C).

TABLE I: Number of Probes at Prefix Sizes

Prefix Size	/32	/40	/48	/56	/64
Number of Probes	64θ	16θ	4θ	θ	1

IV. The New IPv6 Scanner

Currently, there is not yet a tool designed for scanning the Internet-wide IPv6 peripheries. To address this gap, we introduce an new IPv6 scanner meticulously redeveloped from the foundations of ZMap [22] and XMap [2]. It has been enhanced with following innovative modules.

The first *packet generation* module: 1) populates the Interface Identifiers of destination addresses with random bits to provoke indirect responses from last-hop routers (peripheries), and 2) modifies the IPv6 traffic class, flow label, and ICMPv6 checksum field for each probe packet to maximize engagement with the load balancing router's multipath functionality [35]. By doing so, our PaS probes can reach more topologies and routes, enhancing the precision of prefix fingerprinting.

The second *traffic dispersion* module is uniquely capable of effectively spreading its probing traffic across the entire IPv6 address space, distinguishing it from prior tools like XMap, which can only permute the address space of one individual IPv6 prefix.

This scanner is fully compatible with UNIX-like platforms, and has been deployed in the real-world experiments. More details of its implementation will be disclosed in our future works.

V. Real-world Evaluation

In May 2024, we deployed Pruning-as-Scanning from a well-connected server located in Canada, equipped with 8 CPU cores and 32 GB of RAM. Utilizing an uplink rate of 50 kpps, we dispatched approximately 33 billion probes, thereby achieving a thorough measurement of the global IPv6 network peripheries.

A. Global Measurement Metadata

We specifically disclose our experimental raw data to enhance credibility.

To minimize the error probability $e^{-\theta}$ of misclassifying a prefix as redundant, the parameter was set to $\theta = 16$ throughout the measurements. As a result, we allocated a varying number of probes to the prefixes of different lengths: 1024 probes for a /32, 256 probes for a /40, 64 probes for a /48, 16 probes for a /56, and one probes for a /64, consistent with Tab. I.

TABLE II: Raw Results of Pruning-as-Scanning in Rounds

Round	Consumed Probes	Probing Prefixes	Indirect Responses	UF* Prefixes	Periphery Addresses
1 (/32)	537.4 M	524.8 k	7579.6 k	7.4 k	43.9 k
2 (/40)	498.3 M	1946.6 k	128.8 M	59.7 k	1111.7 k
3 (/48)	979.6 M	15.3 M	224.9 M	2692.5 k	10.9 M
4 (/56)	11028.5 M	689.3 M	7984.9 M	78.6 M	319.9 M
5 (/64)	20128.6 M	20128.7 M	7783.6 M	-	442.4 M

$*$: Unique-Fingerprint

The raw results from each round of measurement are presented in Tab. II. Specifically, we initialize the entire IPv6 address space using 54,161 individual IPv6 prefixes (after aggregation) obtained from the RIPEstat API, with their prefix lengths distributed as illustrated in Fig. 4. From this, we extracted a total of 524,845 /32 prefixes.

In **Round 1**, we utilized 537,441,280 probes (524,845 ×1024) to measure the fingerprints of the /32s. This effort yielded 7,579,589 indirect responses (ICMPv6 error messages) and identified 43,911 unique last-hop addresses (IPv6 periphery addresses). From these results, we determined that 7,426 /32s, which were then selected for the next round of measurements, had unique fingerprints.

In **Round 2**, a total of 498,340,096 probes (1,946,641 ×256) were used to probe all /40s. It is noteworthy that the /40

[4] The host-density ratio (HD-Ratio) measures overall IPv6 address utilization efficiency for a given prefix, which can be calculated as:

$$\text{HD-ratio} = \frac{\log \text{number of allocate objects}}{\log \text{maximum number of allocatable objects}}$$

where objects are IPv6 site addresses assigned on a given size (typically /56). HD-ratio is widely adopted as the threshold for justifying the allocation of additional address space.

prefixes investigated exceeded the number of constituent /40s derived from /32 prefixes in the previous round. Because we additionally incorporated the prefixes ranging from /33 to /40 across the entire IPv6 address space. Consequently, this round produced 128,835,349 indirect responses and identified 1,111,696 IPv6 periphery addresses. Following our algorithm, we successfully identified 59,671 /40 prefixes with unique fingerprints.

Similarly, a total of 979,695,232 probes of **Round 3** had been sent to 15,307,738 /48 prefixes, directing 64 probes at each. This measurement obtained 224,991,126 indirect responses, from which we extracted 10,854,249 IPv6 periphery addresses. Moreover, a substantial number of these, precisely 2,692,516 /48 prefixes, were verified as possessing unique fingerprints. Following this, 689,282,560 constituent /56s were derived from those unique-fingerprint /48s and used for further probing with 11,028,520,960 probe packets (689,282,560 ×16) in **Round 4**. As a result, 7,984,997,530 indirect responses were captured, from which we isolated 319,948,597 IPv6 periphery addresses and comfirmed 78,637,440 unique-fingerprint /56s.

In last round (**Round 5**), we exhaustively scanned every /64 subnet of those unique-fingerprint /56s with a total of 20,128,698,639 probes. This effort eventually produced 7,783,686,619 indirect responses and 442,354,713 IPv6 periphery addresses. With the /64 addressing boundary reached [24], [25], further investigation into smaller subnets is unnecessary, eliminating the need to probe /64s with unique fingerprints for finer granularity.

In summary, utilizing Pruning-as-Scanning, the measurement campaign sent a total of 33,172,696,207 probe packets over 200 hours, ultimately identifying an unprecedented 712,126,848 unique IPv6 periphery addresses worldwide.

B. Performance Comparison

Given the absence of establishe approaches for Internet-wide IPv6 periphery discovery, comparing performance with our PaS lacks a clear baseline. As such, We naturally use the scanning of random /64 subnets as the only baseline method for global-scale measurement. Furthermore, to address potential concerns about the fairness of those *related but not globally scalable* approaches, we also reproduce their experiments using the original setups. This is, however, for reference purposes only.

The details of the comparison experiments are shown below, all at an uplink of 50k packets per second (pps):

- **Global-scale Baseline.** We used the same 33.17 billion probes as PaS during random periphery scanning of the entire IPv6 address space. This baseline resulted in 1.02 billion indirect responses but only 1.16 million unique periphery addresses.
- **Scale-limited Approach: XMap.** We replicated XMap's probing campaign within 15 IPv6 blocks as described in its original paper [2]. This effort utilized 64.42 billion probes, yielding 29.46 billion indirect responses, from which we extracted 52.47 million unique periphery addresses.
- **Scale-limited Approach: Edgy.** Edgy requires BGP-informed and hitlist-informed seeds to function effectively [1]. We integrated these data sets to reproduce Edgy's experiments. As the traceroute-like tool Yarrp adopted by Edgy will provides indirect responses from both last-hop devices and intermediate routers, the ratio of indirect responses to consumed probes are relatively higher. As a result, Edgy's entire measurement consumed 13.48 billion probes, received 5.80 billion indirect responses, and identified 64.61 million unique last-hop addresses (IPv6 network peripheries).

To sum up, PaS clearly outperforms all the approaches in comparison, as shown in Tab. III: Despite these established methods having the head-start advantages with the aid of prior knowledge, such as known IPv6 blocks for XMap or seed datasets for Edgy, they fall significantly short in real-world performance compared to our PaS method. In particular, PaS can identify approximately 11× more IPv6 periphery addresses than the scale-limited state-of-the-art method (Edgy) and further offer more than 600× improvement over the baseline in global-scale measurements.

TABLE III: Performance Comparison of PaS and Baselines

Approach	Scale	Probes	Indirect Responses	Periphery Addresses
PaS	**Global**	**33.17B**	**16.13B**	**712.12M**
Random	Global	33.17B	1.02B	1.16M
XMap	Limited	64.42B	29.46B	52.47M
Edgy	Limited	13.48B	5.80B	64.61M

C. How Many of Missed IPv6 Peripheries in Pruning?

To answer this question, we arbitrarily selected a number of discarded prefixes at various granularities (i.e., 10 redundant /32s, 100 redundant /40s, and 1000 redundant /48s) and performing brute-force scans of each /64 subnet to identify new (i.e., potentially missed) IPv6 periphery addresses. This practice is taken for two reasons: first, exhaustively scanning every /64 subnet within the entire IPv6 address space to establish a concrete ground truth is impractical; second, we have previously established a theoretical error boundary for misclassifying a prefix as redundant. The limited-scale experiments are sufficient to cross-validate our theory.

The results are succinctly summarized as follows: no additional periphery addresses were found in the brute-force scan of discarded /32 and /40 prefixes, and those never-before-seen IPv6 periphery addresses were merely extracted from indirect responses of few (< 5) discarded /48 prefixes. Given the prevalent prefix rotation in the IPv6 Internet [9], [32], [36], we believe this is more likely due to address reassignment rather than an omission in the pruning process. Therefore, the missed IPv6 periphery addresses are expected to be negligible in pruning.

D. How Does PaS Enrich the IPv6 Corpus?

The discoveries made by PaS have not only broadened the IPv6 address corpus but have also significantly enhanced our IPv6 insights. To showcase these advancements, we meticulously compared PaS's findings on the following metrics, with existing publicly accessible IPv6 address data, including the HMap6 dataset [37], the AddrMiner dataset [38], and the IPv6 Hitlist Service [39] (abbreviated as Hitlist). All of these datasets were gathered using TGA-based approaches.

1) Address Space Coverage [+]: As per convention [21], [40], we utilize the number of /64 and /80 segments involved by PaS's discoveries and other datasets as the metric for comprehensive evaluation of their IPv6 address space coverage. As shown in Tab. V, PaS's IPv6 (periphery) addresses encompass about 561 million /64s and 681 million /80s, which are respectively $\approx 42\times$ and $\approx 43\times$ of the state-of-the-art works (IPv6 Hitlist Service). This demonstrates the exceptional capacity of PaS to achieve both breadth (coverage of networks) and depth (quality of individual addresses) in IPv6 measurements.

PaS's comparative advancement stems from the inherent limitations of TGA-based approaches, i.e., they can hardly discover IPv6 addresses with patterns that differ from known seeds, as have explained in Fig. 1. While some techniques attempt to borrow patterns from known seeds to explore seedless IPv6 regions, they remain constrained by this foundation.

2) Valid Autonomous Systems [+]: The number of involved Autonomous Systems (ASes) is crucial for evaluating the quality of IPv6 address sets. However, using the absolute number of ASes may inflate performance estimates due to extremely uneven distribution of IPv6 addresses. For example, AddrMiner dataset appears to encompass 21,904 ASes, but 88.28% of its IPv6 addresses originate from its Top-1 AS. Additionally, 9,674 ASes have < 5 addresses each.

To mitigate this potential distortion in performance evaluation, it is imperative to exclude those invalid autonomous systems. Naturally an AS can be considered *valid* only if it hosts a sufficient number of addresses, empirically defined as ≥ 1000. Consequently, results show that PaS's discoveries encompassed the highest number of *valid* ASes, at 1170. In comparison, 766 and 523 *valid* ASes were identified in HMap6 dataset and IPv6 Hitlist, respectively, with AddrMiner dataset featuring the lowest with only 255 *valid* ASes.

Tab. VI provides the Top-10 ASes associated with each address sets. Notably, AddrMiner dataset is predominantly concentrated within a single autonomous system, with 88.29% of addresses residing in AS16509. This pronounced homogeneity in AS distribution compromises the dataset's quality. In contrast, a volume of addresses discovered by PaS are evenly distributed across various ASes, including many not covered by existing address datasets (e.g., Comcast Cable Communications, an American telecommunications company).

TABLE IV: Typical IPv6 Address IID Categories

Category	IID Examples	Comments
EUI64	0250:56ff:fe89:49be	embed MAC address 00:50:56:89:49:be and then flip 7th bit
Embed-IPv4	0012:0122:0126:0072	embed IPv4 address 12.122.126.72
Low-byte	0000:0000:0000:f1b7	all zeros except the lower bytes
Pattern-bytes	0021:2222:0001:0001	more than two bytes of zeros
Randomized	10de:51e8:eb66:7583	pseudorandom

3) EUI-64 Addresses [+]: The lower 64 bits of an IPv6 address constitute the Interface Identifier (IID), and an EUI-64 address incorporates a 64-bit Extended Unique Identifier (EUI-64) derived from the Media Access Control (MAC) address of the network interface card (NIC). Analyzing the IID of IPv6 addresses and quickly collecting EUI-64 addresses play a crucial role in revealing security risks and privacy concerns [9], [10], [36], [41].

Drawing from previous practices [20], [40], we categorized IPv6 addresses into five typical groups based on their Interface Identifiers (IIDs) : EUI-64, Embedded-IPv4, Low-byte, Byte-pattern, and Random as shown in Tab. IV. Although certain datasets can provide the specific types of IPv6 addresses, such as AddrMiner's with Random IID and HMap6's with Low-byte IID, none encompass the entire spectrum of IPv6 IID categories, particularly lacking in EUI-64 addresses, as shown in Fig. 7.

In a notable contrast, the dataset from PaS showcases a comprehensive collection of IPv6 addresses across all IID types, including 53 million with Embedded-IPv4 IIDs, 18 million with Low-byte IIDs, 82 million with Byte-pattern IIDs, and 342 million with random IIDs. Most significantly, it contains 214 million EUI-64 addresses, accounting for 30% of its findings. This quantity surpasses the aggregate of IPv6 addresses in the datasets we compared it to, thus significantly enriching our IPv6 corpus with its diversity and volume.

4) Aliased Addresses [–]: It's crucial to promptly eliminate the aliased address in IPv6 measurements. Probing within an aliased IPv6 prefix[5] generates disproportionately high responses, appearing as though they come from a volume of devices when, in fact, they originate from a single host. This significantly distorts the results from the ground truth. Fortunately, PaS collects only indirect responses during probing, effectively excluding aliased addresses from the results. In contrast, probing an aliased prefix typically receives a direct response from the destinations. Therefore, TGA-based approaches, which focus on extracting sources of direct responses, are more susceptible to aliased prefixes.

Experimental results support the above illustration: Using publicly available aliased and non-aliased prefixes, we found that $< 0.01\%$ of PaS's discoveries are suspected to be aliased addresses. In comparison, the Hitlist [39] shows a ratio of 0.06%, and HMap6 dataset [37] show an increase to 0.96%. Surprisingly, 24.25% of AddrMiner data [38] are identified as aliased addresses, meaning a quarter of its results should have been excluded. This highlights the skewing effects of aliased prefixes on IPv6 measurement and further demonstrates that our PaS not only provides more IPv6 addresses but also maintains a remarkably low alias rate.

VI. Future Works and Ethical Considerations

Admittedly, there are still some limitations of our work while the global-scale IPv6 network periphery scan was ac-

[5] An IPv6 prefix, under which every IP address replies to queries, commonly used in CDNs [18].

Address	IPv6 Address Space Coverage			Valid	EUI-64 Addresses		Aliased Addresses	
Dataset	Addresses	/64s	/80s	ASes	Number	Ratio	Num	Ratio
PaS	**712,126,848**	**561,609,635**	**681,632,549**	**1170**	**214,543,742**	**30.12%**	**36,271**	$< \mathbf{0.01}\%$
HMap6*	42,433,279	7,659,590	8,065,717	523	579,609	1.36%	411,368	0.96%
AddrMiner*	74,348,374	4,257,745	5,786,815	255	946,104	1.27%	18,031,066	24.25%
Hitlist*	24,508,023	13,285,750	16,182,869	766	1,687,538	6.88%	13,556	0.06%

*: Datasets are acquired on May 31, 2024.

TABLE V: Overall Comparison of Pruning-as-Scanning with Existing IPv6 Address Datasets

PaS	Autonomous System Name - Country	Ratio
AS55836	Reliance Jio Infocomm Limited - India	21.12%
AS45609	Bharti Airtel Ltd. - India	13.22%
AS9808	CMCC - China	12.86%
AS7922	Comcast - United States	7.07%
AS4134	China Telecom Backbone - China	5.22%
AS38266	Vodafone Idea Ltd - India	3.39%
AS4837	China Unicom Backbone - China	2.34%
AS45271	Idea Cellular Limited - India	2.09%
AS131445	Advance Wireless Network - Thailand	2.08%
AS24445	CMCC (Henan) - China	1.90%

AddrMiner	Autonomous System Name - Country	Ratio
AS16509	Amazon.com, Inc. - United States	88.29%
AS12322	Free SAS - France	2.15%
AS36183	Akamai Technologies, Inc. - United States	0.81%
AS20940	Akamai International B.V. - Netherlands	0.52%
AS36492	Google, LLC - United States	0.42%
AS13335	Cloudflare, Inc. - United States	0.35%
AS14061	DigitalOcean, LLC - United States	0.31%
AS20773	Host Europe GmbH - Germany	0.25%
AS3320	Deutsche Telekom AG - Germany	0.23%
AS63949	Akamai Connected Cloud - Global	0.21%

HMap6	Autonomous System Name - Country	Ratio
AS60781	LeaseWeb Netherlands B.V. - Netherlands	13.37%
AS18779	EGIHosting - United States	6.59%
AS30633	LeaseWeb USA, Inc. - United States	5.43%
AS46723	Unlabeled LLC - United States	2.58%
AS34665	Petersburg Internet Network ltd. - Russia	2.39%
AS400522	Rootcloud LLC - United States	2.37%
AS29632	Netassist Limited - Gibraltar	2.09%
AS398559	Tunbroker LLC - United States	1.93%
AS57043	HOSTKEY B.V. - Netherlands	1.82%
AS399975	TFIRE - United States	1.79%

Hitlist	Autonomous System Name - Country	Ratio
AS13335	Cloudflare, Inc. - United States	14.39%
AS12322	Free SAS - France	9.84%
AS45609	Bharti Airtel Ltd. - India	4.65%
AS47583	Hostinger International Limited - Cyprus	4.09%
AS55836	Reliance Jio Infocomm Limited - India	2.80%
AS5607	SKY UK Limited - United Kingdom	2.46%
AS17676	SoftBank Corp. - Japan	1.71%
AS4134	China Telecom Backbone - China	1.49%
AS44812	IP SERVER LLC - Russian Federation	1.44%
AS7738	Vítal - Brazil	1.43%

TABLE VI: Top 10 Autonomous Systems on the IPv6 Address Datasets. Note that majority of the AddrMiner addresses are located in the AS16509.

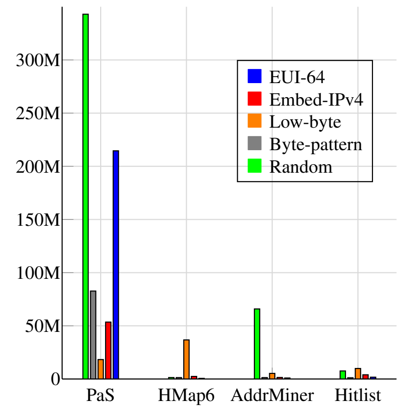

Fig. 7: Address Categorization with Typical IIDs on IPv6 Address Datasets.

complished. We look forward to improving it in the future from following aspects:

- **Reducing RAM Footprint.** The PaS itself does not inherently require a large RAM footprint. However, the current prefix identification algorithm, although effective, demands substantial RAM for storing fingerprints, due to its reliance on simplicity and counter-based tracking. Future efforts will focus on integrating more efficient data structures, like hash tables and bitmaps, to decrease memory overhead.
- **Broadening Probing Protocols.** We implement the PaS method using the ICMPv6 Echo Request protocol, similar to ping. Our method focus on indirect responses (ICMPv6 error messages) rather than on the probe protocol itself, allowing us to explore how various protocols, like TCP-SYN or UDP, affect efficiency in future.
- **Enhancing Security Analysis.** Given the IPv6 network periphery's potential to host critical network assets, the scarcity of comprehensive security research in this area motivates us to bridge this knowledge gap.

Adhering to ethical Internet use, we have incorporated the following ethical considerations in our measurements:

- **Self-restraint.** We limited the probing rate to an uplink bandwidth of 50 kpps throughout, as agreed with our VPS providers. We probe each /64 prefix once only in a random sequence to minimize network impact.
- **Complaint Handling.** Network administrators seeking privacy can easily contact us via the email addresses listed in the WHOIS database or the reverse DNS records. We have promptly excluded network entities within at least five ASes from our results upon receiving complaints.
- **Responsible Data Disclosure.** We will release the unique-fingerprint /32, /40, and /48 prefixes identified by Pruning-as-Scanning. However, in order to prevent the cybercriminal abuse, the finer-grained prefixes and periphery addresses will not be made public to the community.

VII. Conclusion

This paper introduces an innovative approach, Pruning-as-Scanning (PaS), which is the first to efficiently and thoroughly discover IPv6 periphery addresses on an Internet-wide scale with negligible omissions. PaS pioneeringly leverages the fingerprints of prefixes to significantly narrow the search scope during IPv6 scanning, unlike existing methods that require a priori information such as seed datasets. Moreover, we have implemented an asynchronous scanner prototype based on PaS's principle. Real-world tests demonstrated that our approach outperformed all baseline methods in terms of probing scale (from limited to global) and the sheer volume of periphery addresses (from seed-assisted 64M to self-supported 712M). Furthermore, the discoveries made by PaS remarkably enrich the IPv6 corpus, filling various gaps that existing IPv6 address datasets exhibit.

Acknowledgment

The authors would like to greatly appreciate the anonymous reviewers for their insightful comments. This work is supported by the National Natural Science Foundation of China (No. 62472434 and No. U22B2005) and the Science and Technology Innovation Program of Hunan Province (No. 2022RC3061). Tao Yang and Ling Hu contributed equally to this work. The corresponding author is Zhiping Cai.

References

[1] E. C. Rye and R. Beverly, “Discovering the IPv6 Network Periphery,” in *Proc. PAM*. Springer, 2020, pp. 3–18.

[2] X. Li, B. Liu, X. Zheng, H. Duan, Q. Li, and Y. Huang, “Fast IPv6 Network Periphery Discovery and Security Implications,” in *Proc. DSN*. IEEE, 2021, pp. 88–100.

[3] J. Korhonen, J. Arkko, T. Savolainen, and S. Krishnan, “IPv6 for Third Generation Partnership Project (3GPP) Cellular Hosts,” Internet Requests for Comments, RFC Editor, RFC 7066, November 2013.

[4] C. Byrne, D. Drown, and A. Vizdal, “Extending an IPv6 /64 Prefix from a Third Generation Partnership Project (3GPP) Mobile Interface to a LAN Link,” Internet Requests for Comments, RFC Editor, RFC 7278, June 2014.

[5] H. Singh, W. Beebee, C. Donley, and B. Stark, “Basic Requirements for IPv6 Customer Edge Routers,” Internet Requests for Comments, RFC Editor, RFC 7084, November 2013.

[6] J. Woodyatt, “Recommended Simple Security Capabilities in Customer Premises Equipment (CPE) for Providing Residential IPv6 Internet Service,” Internet Requests for Comments, RFC Editor, RFC 6092, January 2011.

[7] E. Davies and J. Mohacsi, “Recommendations for Filtering ICMPv6 Messages in Firewalls,” Internet Requests for Comments, RFC Editor, RFC 4890, May 2007, http://www.rfc-editor.org/rfc/rfc4890.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4890.txt

[8] D. Plonka and A. Berger, “Temporal and Spatial Classification of Active IPv6 Addresses,” in *Proc. IMC*. ACM, 2015, pp. 509–522.

[9] E. Rye, R. Beverly, and K. C. Claffy, “Follow the Scent: Defeating IPv6 Prefix Rotation Privacy,” in *Proc. IMC*. ACM, 2021, pp. 739–752.

[10] E. C. Rye and R. Beverly, “IPvSeeYou: Exploiting Leaked Identifiers in IPv6 for Street-level Geolocation,” in *Proceeding of the IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023, pp. 3129–3145.

[11] F. Gont and T. Chown, “Network Reconnaissance in IPv6 Networks,” Internet Requests for Comments, RFC Editor, RFC 7707, March 2016.

[12] J. Czyz, M. Luckie, M. Allman, M. Bailey *et al.*, “Don’t Forget to Lock the Back Door! A Characterization of IPv6 Network Security Policy,” in *Proc. NDSS*, 2016.

[13] L. Pan, J. Yang, L. He, Z. Wang, L. Nie, G. Song, and Y. Liu, “Your Router is My Prober: Measuring IPv6 Networks via ICMP Rate Limiting Side Channels,” in *Proc. NDSS*, 2023.

[14] J. Ullrich, P. Kieseberg, K. Krombholz, and E. Weippl, “On Reconnaissance with IPv6: A Pattern-based Scanning Approach,” in *Proc. International Conference on Availability, Reliability and Security*. IEEE, pp. 186–192.

[15] P. Foremski, D. Plonka, and A. Berger, “Entropy/IP: Uncovering Structure in IPv6 Addresses,” in *Proc. IMC*. ACM, 2016, pp. 167–181.

[16] A. Murdock, F. Li, P. Bramsen, Z. Durumeric, and V. Paxson, “Target Generation for Internet-wide IPv6 Scanning,” in *Proc. IMC*. ACM, 2017, pp. 242–253.

[17] 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, pp. 31–46, 2019.

[18] O. Gasser, Q. Scheitle, P. Foremski, Q. Lone, M. Korczynski, S. D. Strowes, L. Hendriks, and G. Carle, “Clusters in the Expanse: Understanding and Unbiasing IPv6 Hitlists,” in *Proc. IMC*, ACM. New York, NY, USA: ACM, 2018.

[19] 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. INFOCOM*. IEEE, 2021, pp. 1–10.

[20] 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. INFOCOM*. IEEE, 2022, pp. 1679–1688.

[21] B. Hou, T. Yang, Z. Cai, K. Wu, and T. Zhou, “Search in the Expanse: Towards Active and Global IPv6 Hitlists,” in *Proc. INFOCOM*. IEEE, 2023.

[22] Z. Durumeric, E. Wustrow, and J. A. Halderman, “ZMap: Fast Internet-wide Scanning and Its Security Applications,” in *Proc. USENIX Security*, vol. 8, 2013, pp. 47–53.

[23] R. Beverly, R. Durairajan, D. Plonka, and J. P. Rohrer, “In the IP of the Beholder: Strategies for Active IPv6 Topology Discovery,” in *Proc. IMC*. ACM, 2018, pp. 308–321.

[24] R. Hinden and S. Deering, “IP Version 6 Addressing Architecture,” Internet Requests for Comments, RFC Editor, RFC 4291, February 2006. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4291.txt

[25] B. Carpenter, T. Chown, F. Gont, S. Jiang, A. Petrescu, and A. Yourtchenko, “Analysis of the 64-bit Boundary in IPv6 Addressing,” Internet Requests for Comments, RFC Editor, RFC 7421, January 2015.

[26] M. Boucadair, A. Petrescu, and F. Baker, “IPv6 Prefix Length Recommendation for Forwarding,” Internet Requests for Comments, RFC Editor, BCP 198, July 2015.

[27] A. Conta, S. Deering, and M. Gupta, “Internet Control Message Protocol (ICMPv6) for the Internet Protocol version 6 (IPv6) Specification,” Internet Requests for Comments, RFC Editor, RFC 4443, March 2006. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4443.txt

[28] 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. INFOCOM*. IEEE, 2021, pp. 1–10.

[29] 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, p. 108666, 2022.

[30] R. N. C. Centre, “Allocation history,” https://stat.ripe.net/docs/02.data-api/allocation-history.html, accessed: 2024-05-31.

[31] A. APNIC and N. RIPE, “IPv6 Address Allocation and Assignment Policy,” 2020. [Online]. Available: https://www.ripe.net/publications/docs/ripe-738

[32] R. Padmanabhan, J. P. Rula, P. Richter, S. D. Strowes, and A. Dainotti, “DynamIPs: Analyzing Address Assignment Practices in IPv4 and IPv6,” in *Proc. CoNEXT*. ACM, 2020, pp. 55–70.

[33] T. Narten, G. Huston, and L. Roberts, “IPv6 Address Assignment to End Sites,” Internet Requests for Comments, RFC Editor, BCP 157, March 2011.

[34] G. Huston, “Considerations on the IPv6 Host Density Metric,” Internet Requests for Comments, RFC Editor, RFC 4692, October 2006.

[35] R. Almeida, R. Teixeira, D. Veitch, C. Diot *et al.*, “Classification of Load Balancing in the Internet,” in *Proc. INFOCOM*. IEEE, 2020, pp. 1987–1996.

[36] E. Rye and D. Levin, “IPv6 Hitlists at Scale: Be Careful What You Wish For,” in *Proc. SIGCOMM*, 2023, pp. 904–916.

[37] H. Bingnan, “HMap6 Hitlist,” https://github.com/hbn1987/6Scan, accessed: 2024-05-31.

[38] G. Song, “AddrMiner IPv6 Hitlist,” https://addrminer.github.io/IPv6_hitlist.github.io/#hitlist-a, accessed: 2024-06-04.

[39] O. Gasser, Q. Scheitle, P. Foremski, Q. Lone, M. Korczynski, S. D. Strowes, L. Hendriks, and G. Carle, “IPv6 Hitlist Service,” https://ipv6hitlist.github.io/, accessed: 2024-05-31.

[40] 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.

[41] S. J. Saidi, O. Gasser, and G. Smaragdakis, “One Bad Apple Can Spoil Your IPv6 Privacy,” *ACM SIGCOMM Computer Communication Review*, vol. 52, no. 2, pp. 10–19, 2022.