Luori: Active Probing and Evaluation of Internet-wide IPv6 Fully Responsive Prefixes

Daguo Cheng*, Lin He*†, Chentian Wei*, Qilei Yin†, Boran Jin‡, Zhaoan Wang*, Xiaoteng Pan§, Sixu Zhou§, Ying Liu*†, Shenglin Zhang§, Fuchao Tan¶, Wenmao Liu¶
*Tsinghua University, †Zhongguancun Laboratory,
‡Beijing University of Posts and Telecommunications, §Nankai University, ¶NSFOCUS, Inc.

***Abstract*—With the large-scale deployment and application of IPv6, IPv6 network measurements will become increasingly important. However, a special type of IPv6 prefix called Fully Responsive Prefix (FRP) is having a significant impact on IPv6 measurement campaigns, which is defined as all addresses under a prefix responding to scans. Obviously, there cannot be a real responder behind each of these addresses. To reveal the current status and impact of Internet-wide IPv6 FRPs, we propose for the first time an active probing method for Internet-wide IPv6 FRPs, *Luori*, which transforms the active probing process under IPv6 huge prefix space (potential range of prefix presence) into a dynamic search process in a tree based on reinforcement learning, achieving efficient probing of arbitrary routing prefixes. The evaluation results show that *Luori* found 31.7K largest FRPs in a single Internet-wide probing with 11M budget, covering 1.5×10^{30} address space, which is $10^6\times$ that of existing methods. More importantly, after six months of Internet-wide probing, we have found 516K largest FRPs, which covers 1.3×10^{33} address space and 795 ASes, making it the largest publicly known FRP list. Based on this list, we screen out 20% of the addresses covered by FRPs from a well-known IPv6 active address dataset. Furthermore, we further analyze and find that the distribution of these FRPs is extensive and their implementation methods are diverse, which can provide beneficial references for the practical application of FRPs. We also make this list publicly available and maintain it long-term for use and study by relevant researchers.**

***Index Terms*—IPv6, fully responsive prefix, active probing**

I. Introduction

As the new generation of Internet protocol, IPv6 is rapidly developing and widely deployed globally. As of May 2024, about 40% of Google users can access its services via IPv6, up from less than 2% a decade ago [1]. This trend has also prompted numerous studies [2]–[12], one of the most typical of which is IPv6 network measurement. For example, in recent years, a large number of research efforts have been proposed to detect active IPv6 addresses [13]–[25], laying the foundation for understanding and optimizing IPv6 networks.

However, a special class of prefixes in IPv6 is having a huge impact on IPv6 network measurement: the *fully responsive prefix (FRP)* [19], [26], [27]. An FRP is defined as a prefix where all addresses respond to probes, such as ICMPv6 Echo Request. For example, a single /48 FRP can cover 2^{80} responsive addresses, but it is impossible for each of these addresses to correspond to a unique responder. Therefore, if FRPs are ignored in network measurement campaigns, many probing resources may fall into the "FRP traps", which not only wastes significant resources and time but also introduces an undesired bias and could even distort measurement conclusions. For instance, in IPv6 active address probing, if an FRP is not discovered in time, all probing resources may be gradually consumed by it, leading to meager rewards. In a dataset containing 24.5M active IPv6 addresses from a well-known hitlist service published on May 4, 2024 [26], [28], [29] (§IV-H), we find that there are 4.8M addresses under FRPs, accounting for 20% of the total. According to the 20%-50% hit rate of most of the currently active address probing methods [13]–[17], [20]–[24], this means there are nearly 9.6M to 24M probes that do not target real active addresses, resulting in wasted probing resources. Even worse, when we conduct other network measurement campaigns based on these addresses (e.g., port scanning [30]–[32] and topology discovery [33]–[35]), the measurement results may not accurately reflect the actual situation in the network and could be misleading.

Currently, research on FRPs is scarce. Efforts such as [17], [19], [28] have identified the presence of FRPs in active address probing and have proposed methods akin to brute-force probing to validate FRPs in probed active addresses (§II). Sattler et al. [27] have also studied similar prefixes (highly responsive prefixes) under IPv4 through brute-force probing. However, under IPv6, brute-force probing of Internet-wide FRPs is impractical, even with some high-speed tools like ZMap [36] and Masscan [37]. This is because IPv6's address space is 2^{96} times that of IPv4. Therefore, this paper aims to propose an efficient active probing method for Internet-wide FRPs and to perform long-term probing, thereby constructing and maintaining an Internet-wide FRP list (FRPlist) over time. On the one hand, the FRPlist can help researchers mitigate the impact of FRPs on their measurement campaigns and serve as a dataset to reduce unnecessary network burdens from redundant measurements. On the other hand, it can reveal the status and characteristics of abundant Internet-wide FRPs, enabling the exploration of their usage scenarios and implementation methods, and providing insights into their practical applications. However, achieving this goal is not straightforward, and we confront the following challenges:

- **Sparse distribution.** Despite the considerable number of FRPs, their distribution is still extremely sparse when compared to the vast IPv6 address space, presenting a significant

979-8-3503-5171-2/24/$31.00 ©2024 IEEE

challenge for their detection and discovery.

- **Diverse FRP patterns.** The implementation of FRPs can vary widely and may be of arbitrary lengths, e.g., the FRP patterns under different global routing prefixes may be different, which makes it difficult to analyze and extract their features.
- **Dynamic change.** FRP patterns may change dynamically over time, e.g., when the network environment changes. Continuously capturing FRP patterns to achieve long-term effective probing is necessary but challenging.

To tackle these challenges, we propose *Luori*, an efficient FRP active probing method based on reinforcement learning that can probe FRPs across any target prefix at Internet scale. The core idea is to represent the entire probing space as a tree, and then abstract the probing process as a dynamic search process in the tree. Specifically, we leverage the hierarchical nature of trees to represent the entire potential probing space, which is applicable to any FRP pattern, and to initially carve potential FRP patterns for each target prefix in the tree based on known FRPs. Then, based on the reinforcement learning idea, we propose an ϵ-Q strategy, which can continuously capture FRP patterns and adjust the direction of probing in the tree based on the probing feedback, thereby improving the probing efficiency and long-term probing effectiveness under the vast prefix space. To the best of our knowledge, this is the first study about Internet-wide IPv6 FRPs, whose contributions can be summarized as follows:

- We propose *Luori*, ***the first known Internet-wide FRP active probing method***, which transforms the active probing process under IPv6 huge prefix space into a dynamic search process in a tree based on reinforcement learning.
- We have implemented and deployed *Luori* to allow continuous probing of Internet-wide FRPs. The evaluation results show that *Luori* found 31.7K FRPs[1] in a single Internet-wide probing with 11M budget[2], covering 1.5×10^{30} address space[3], which is $10^6\times$ that of brute-force probing.
- As of now, *Luori* has been continuously probing Internet-wide for almost six months and has found 516K FRPs, which covers 1.3×10^{33} address space and 795 ASes, far exceeding the existing FRPlist, making it ***the largest publicly known FRPlist***. We also make the code and the FRPlist of this paper publicly available[4].
- For the well-known and commonly used active address dataset (e.g., Gasser's hitlist [26], [28], [29]), we find that 20% of its addresses belong to the FRPs we provide. Furthermore, our analysis reveals that the FRPs in the FRPlist mainly come from about 650 organizations in industries such as Cloud or CDN services, and network security. Meanwhile, the implementation of FRPs is diverse, in addition to the specific technical means of Cloud or CDN vendors (e.g., Cloudflare's addressing agility approach [27], [38]), most of them are based on the existing functionality implemented in Linux systems.

[1]number of FRPs after taking the largest (same as below), i.e., avoid co-existence of a prefix and its sub-prefixes

[2]number of FRPs planned for probing

[3]number of addresses covered by the FRPs

[4]https://frpv6.github.io/

II. Related Work

IPv6 FRPs. Murdock et al. [17] first mentioned such prefixes in IPv6 active address probing, naming them ***aliased prefixes*** and defining them as prefixes where all addresses within respond to scans and are considered responses from a single host. Subsequently, Zirngibl et al. [26] found that some CDN vendors' technical implementations also cause all addresses of a prefix to respond, but not from a single host, such as in Cloudflare's addressing agility approach [27], [38]. They then introduced the concept of FRPs, defining them as prefixes where all addresses respond, which encompasses aliased prefixes. Additionally, DET [19] calls FRPs as active prefixes. In this paper, we adopt the FRP concept.

FRP Probing. In IPv6 active address probing, Murdock et al. [17] and Gasser et al. [28] have respectively proposed FRP probing methods based on active addresses to mitigate their impact on probing results. Notably, the MAPD method proposed by Gasser et al. has become a common approach. Note that MAPD claims to be an aliased prefix probing method, but it does not determine that all responses under a prefix come from the same host, so it is still essentially an FRP probing method. In subsequent active address probing research based on MAPD [13], [14], [16], [19], [21], IPv6 FRPs are found to be commonly present in networks. The core of MAPD is to brute-force enumerate prefixes of all lengths contained in the active address as candidate FRPs (with a high probability of being FRP), and then detect whether they are real FRPs using the APD method. However, due to its brute-force nature, there are some limitations on candidate prefixes, i.e., just enumerate candidate prefixes with lengths greater than or equal to 64. Although the way MAPD generates candidate FRPs is brute-force, the detection method (APD) it proposes is still effective and has been widely used. In this paper, we also use the classical APD detection method to determine whether a prefix is FRP or not. APD determines whether a prefix is an FRP by generating a random address under each of the 16 sub-prefixes under the prefix. For example, for prefix 2001:5af3::/32, 16 addresses are generated: $2001:5af3:\{0-f\}***:****:****:****:****:****$, where nibble 9 iterates over the values from 0 to f, and $*$ stands for a random value. This prefix is considered an FRP when all 16 addresses are responsive.

III. *Luori* Design

A. Overview

The goal of *Luori* is to continuously and efficiently discover IPv6 FRPs at Internet scale. Fig. 1 shows the high-level overview of *Luori*. The *Search Tree Construction* module constructs a search tree for each target prefix, representing the entire probing space and potential FRP patterns under the target prefix. The *FRP Probing* module continuously generates candidate FRPs in the search tree and actively probes to determine if they are FRPs.

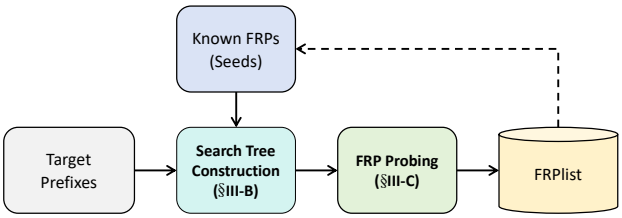

Fig. 1. High-level overview of *Luori*

Workflow. When conducting Internet-wide FRP probing, we first preprocess data on Internet-wide routing prefixes [39] and known FRPs [26], [28], [29] collected from publicly available sources. For known FRPs (seeds), we take the largest as the seed dataset for this paper. For Internet-wide routing prefixes, we directly obtain fully responsive routing prefixes (FRRPs) and non-fully responsive routing prefixes (NFRRPs, i.e., target prefixes) through APD. After the search tree construction module acquires all the target prefixes and seeds, it constructs a search tree for each target prefix separately. Next, the FRP probing module uses the search trees of all target prefixes as input and continuously discovers new FRPs under each search tree. Ultimately, *Luori* outputs all probed FRPs and forms the FRPlist, which also can be added to known FRPs for the next probe.

B. Search Tree Construction

As discussed, exhaustive enumeration is not suitable for discovering FRPs. Inspired by seed-based active IPv6 address discovery [13]–[22], we also employ the learned features from known FRPs (i.e., seeds) to guide the probing of unknown FRPs. Thus, accurately representing the patterns of the seeds is crucial. However, this is not easy because the FRP patterns are diverse and dynamically changing, e.g., the length of the pattern changes when a parent prefix of some FRPs is found to be FRP. Considering that the hierarchical nature of trees is well-suited for addressing the dynamic changes in FRP patterns, we propose constructing a tree for each target prefix to represent its entire potential probing space. Meanwhile, inspired by Monte Carlo tree search [40], [41] in reinforcement learning, we further transform this tree into a search tree, representing both nodes and their states to capture FRP patterns. Additionally, to initiate probing, we initially carved potential FRP patterns for Internet-wide all target prefixes in the tree based on seeds.

Since the FRP under target prefixes consists of both a global routing prefix (assigned by the Regional Internet Registry or the Internet Service Provider) and a local subnet identifier (SID, configured by the routing prefix owner), the FRP pattern is determined by SID. For target prefixes that contain seeds (*seeded prefixes*), we can construct the search tree directly. As shown in Fig. 2, when constructing a search tree based on the seeds of a seeded prefix, we first create the root node (i.e., the seeded prefixes), and then recursively perform *Expansion* (§III-B1) and *Backpropagation* (§III-B2) based on SID of each seed, thereby establishing a complete search tree. Its different layers represent prefixes of different lengths, and each node can represent a prefix (i.e. the path to reach node). Moreover, the states of each node are reflected by the value of the node (e.g. whether it is FRPs or not). All nodes and their states represent the FRP pattern of the target prefix, and the FRP pattern changes dynamically with the change of node states. However, seeds are limited, so most target prefixes do not have seeds (*unseeded prefixes*). For such prefixes, we propose an *inter-prefix search tree transfer strategy* for search tree construction (see §III-B3).

1) Expansion: We expand directly from the root node based on the SID of each seed. The depth of expansion is determined by the length of the SID. Specifically, the expansion consists of two parts: seed node expansion and potential node expansion. Seed node expansion refers to the direct expansion from top to bottom in the tree, following the left-to-right order of the seed's SID. This process forms a seed path in the tree, composed of seed nodes, representing a known successful search path (i.e., an FRP). Furthermore, the core of relying on seeds to guide is that the seeds pattern can initially represent the overall pattern of the target prefix, i.e., the probability of potential FRPs existing in seed region (adjacent to seed nodes) is higher, because they share a consistent pattern. Therefore, we also perform potential node expansion based on the individual seed nodes on the seed paths. Specifically, first expand all the potential nodes (i.e., $\in \{0-f\}$) in the same layer as the seed node, and then associate them with their parent nodes in the upper layer. As shown in Fig. 2, when the seeded prefix is 2001:5af3::/32 (i.e., the root), x = 32. If one of its seeds is 2001:5af3:1f90::/44 and assuming it is the first seed in the recursive process, its SID is "1f9". The expanded seed nodes are "1", "f" and "9", forming a seed path. The other expanded nodes are potential nodes. The depth of the expansion is up to layer 4. Each node in the tree represents a prefix. For example, the potential node "2" in layer 2 represents the prefix 2001:5af3:2000::/36, and the leaf seed node "9" in layer 4 represents the seed 2001:5af3:1f90::/44. However, it should be noted that the non-leaf seed node does not represent a seed, as it is simply a parent prefix of the seed, e.g., the prefix 2001:5af3:1f00::/40 represented by the seed node "f" in layer 3.

2) Backpropagation: The goal is to update the states (i.e., value information) on each node layer by layer in a reverse manner, starting from the leaf seed node until reaching the root node. This phase also mainly consists of two parts: known value backpropagation and potential value backpropagation. In known value backpropagation, since seeds are known FRPs, the known value (V_k) of all seed nodes on the seed path is increased by v (i.e., $V_k+ = v$), and the number of known visits (N_k) is also increased by 1 (i.e., $N_k+ = 1$). In potential value backpropagation, potential nodes receive value (V_p) from adjacent seed nodes. The V_p obtained by different potential nodes of a seed node gradually decrease from near to far (i.e., $V_p = v--$), because it is more likely to be an FRP when closer to the seed pattern. Meanwhile, the number of potential visits (N_p) to the potential node is also increased by 1 (i.e., $N_p+ = 1$). However, their V_p and N_p are not propagated upwards, because they have not been actually visited and have

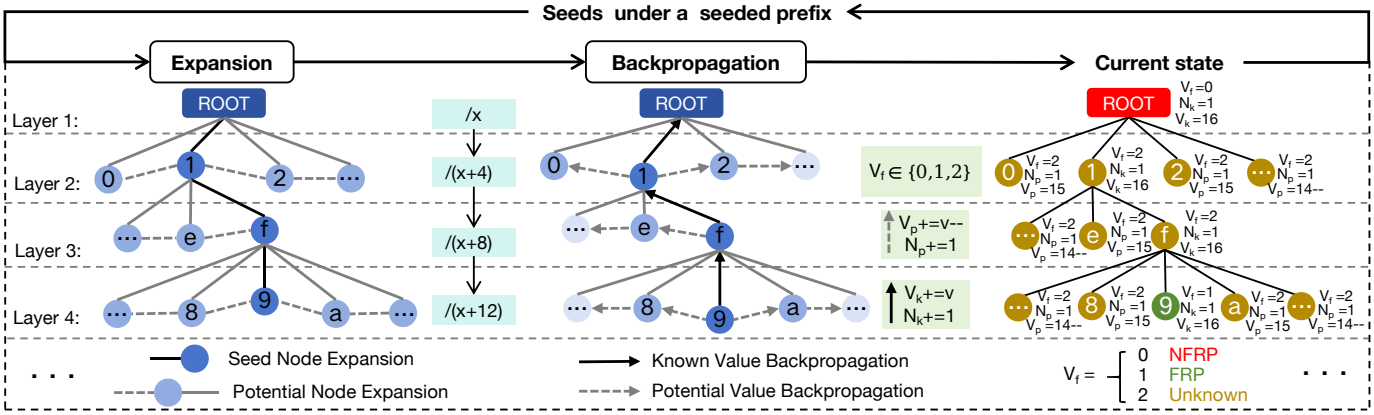

Fig. 2. Search Tree Construction, where $+=$ denotes accumulation and $v--$ denotes the V_p of the node at the omission decreases from a constant v

no known value. They just have a higher probability to be FRP nodes, and their true value has to be explored further. In addition, we also mark whether the prefixes represented by the node is FRP or not. A flag value (V_f) of 0 indicates a non-fully response prefix (NFRP), $V_f = 1$ indicates an FRP (e.g., a seed), and $V_f = 2$ represents unknown. Fig. 2 shows the current state of the search tree after expansion and backpropagation based on the first seed 2001:5af3:1f90::/44 when $v = 16$. The root node (i.e., seeded prefix) is predetermined to be a NFRRP, so $V_f = 0$. The seed node "9" in layer 4 (i.e., seed) has $V_f = 1$, and all other nodes are $V_f = 2$. Note that the V_f of seed nodes "1" and "f" is also 2, since a seed's parent prefix may also be FRP.

3) Inter-prefix Search Tree Transfer: As the above process is recursively executed, the nodes and states of this tree will undergo constant changes. The V_k, N_k, V_p and N_p of the nodes will continuously accumulate, and the V_f may also change. For example, the V_f of a potential node may change from 2 to 1 as a new seed is loaded. Once all the seeds under a seeded prefix are loaded into the search tree, this search tree represents the current FRP pattern of the seeded prefix and its entire potential probing space (since the tree can be continuously expanded), initiating subsequent FRP probing process. However, for unseeded prefixes, since they do not have any prior knowledge (i.e., seeds), the FRP patterns are completely unknown. Therefore, we also propose an **inter-prefix search tree transfer strategy**, whose goal is to transfer some potentially similar search trees (i.e., FRP patterns) from seeded prefixes for each unseeded prefixes. Otherwise, it is difficult to initiate probing.

The core of this strategy is that target prefixes with the same attributes (e.g., AS) have similar FRP patterns, i.e., the relevant attributes of target prefixes are used to transfer the search tree. First, based on network management experiences, prefixes with similar attributes are likely to have consistent configurations because they often have consistent use cases and are managed by the same network engineers. This experience is also applied in similar tasks, such as active address probing [21]. In addition, we also conducted tests based on the three attributes of the prefix belongs to the AS [39], organization [42] and business categories [43], [44] selected in this paper. We find that the information entropy of seeds under a single attribute decreases compared to the information entropy of all seeds, especially AS. Therefore, based on these three attributes, we design a hierarchical transfer strategy from fine-grained to coarse-grained. Specifically, for a unseeded prefix, we first obtain its AS and then match it with all the ASes to which the seeded prefix belongs. If successfully matched, search trees (excluding the root node) of all matched seeded prefixes are merged to create a new tree. During the merge process, nodes from different trees are combined, and the value information of the nodes is accumulated. Once the root node is set to the unseeded prefix, the search tree transfer is completed. If any search tree cannot be matched based on AS, then layer-by-layer matching will be performed based on the organization or business category to which the prefix belongs. Since the granularity of business categories is very coarse, eventually all target prefixes will get a search tree.

C. FRP Probing

The purpose of this module is to efficiently probe all target prefixes (including seeded and unseeded prefixes) based on search trees. However, since the search tree can be continuously expanded, its search space is huge. Meanwhile, the seed patterns may not fully represent the FRP patterns under the target prefix, e.g., they may be incomplete. Therefore, relying solely on the patterns of seeds for probing may result in a large number of FRPs not being found. To address this challenge, based on reinforcement learning idea, we propose an ϵ-Q strategy, which putting the limited probing resources on the more worthwhile branches to discover more new FRPs. And this tree is continuously updated (i.e., pattern) based on probing feedback to facilitate ϵ-Q to make optimal decisions in real-time. Fig. 3 illustrates the workflow of dynamic probing, which iteratively executes the *Selection* (III-C1), *Probing* (III-C2) and *Backpropagation* (III-C3) modules according to the budget of each target prefix (B).

1) Selection: The aim is to select the most valuable branch (i.e., a candidate FRP is generated) in the current state of the search tree. The recursive selection process starting from the root node is usually completed by a strategy, the core of which is to use the value information of the nodes in the tree's

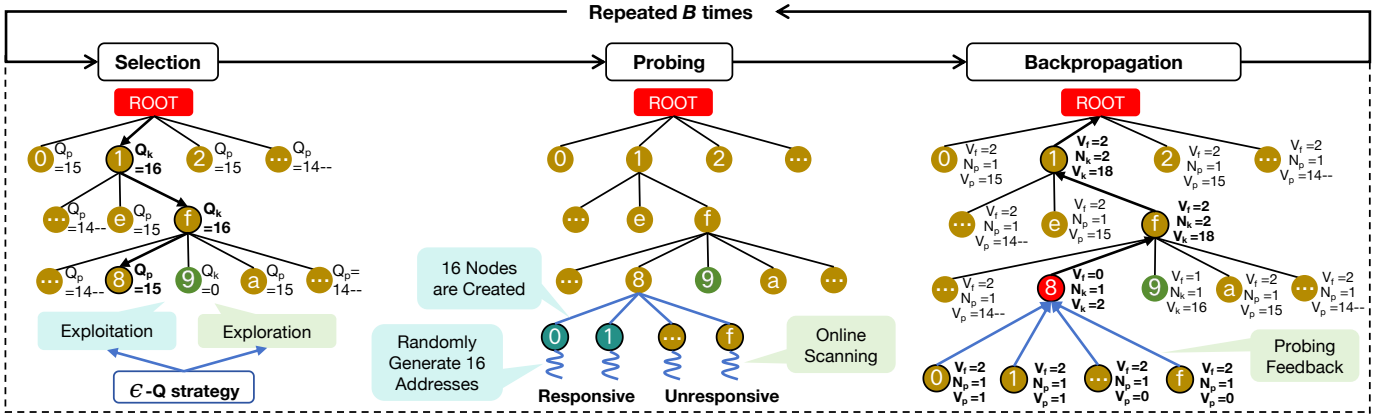

Fig. 3. FRP Probing (search tree in initial state is shown in Fig. 2)

current state to make the optimal choice. However, this optimal choice is not to always select the known better nodes (e.g., seed nodes "1" and "f"). Although we want to select along the seed path as much as possible (i.e., *exploiting* existing seed prior knowledge), it also needs to be able to select in other unknown paths (i.e., *exploring* the space where seeds do not exist), to avoid getting stuck in the seed pattern without any opportunity to discover more other FRP patterns. To achieve a balance between *exploitation* and *exploration*, we designed the selection strategy ϵ-Q in this paper based on the ϵ-Greedy algorithm [45], [46]:

$$a = \begin{cases} \arg\max_{a \in A_k(s)} Q_k(s,a) & P_k = 1-\epsilon, \\ \arg\max_{a \in A_p(s)} Q_p(s,a) & P_p = \epsilon, \end{cases} \tag{1}$$

where s and a represent the current state and action (i.e., selecting a node from $\{0-f\}$), respectively. $A_k(s)$ and $A_p(s)$ represent the set of known and potential nodes that can be selected in a single selection under state s, respectively. $A_k(s)$ includes the seed nodes and the newly probed FRP nodes (i.e., nodes on the new FRP paths) before state s. $A_k(s) \cup A_p(s) = \{0-f\}$. Specifically, this strategy consists of two components, corresponding to *exploitation* and *exploration*. In each selection, the action a with the highest known expected reward (Q_k) in the $A_k(s)$ is selected with probability $1-\epsilon$ (i.e., *exploitation*), and the action a with the highest potential expected reward (Q_p) in the $A_p(s)$ is selected with probability ϵ (i.e., *exploration*). The balance between *exploitation* and *exploration* can be adjusted by ϵ ($\in [0,1]$). Additionally, it can be seen that in *exploration*, we introduce Q_p to select from potential nodes, instead of randomly selecting. This is conducive to the full utilization of prior knowledge and can effectively increase the likelihood of selecting high-value nodes. The functions Q_k and Q_p are defined as follows:

$$Q_k(s,a) = \frac{V_k(s,a)}{N_k(s,a)} \times |V_f - 1| \quad a \in A_k(s), \tag{2}$$

$$Q_p(s,a) = \frac{V_p(s,a)}{N_p(s,a)} \times |V_f - 1| \quad a \in A_p(s), \tag{3}$$

Where $Q_k(s,a)$ and $Q_p(s,a)$ are the Q_k and Q_p of action a under state s. $V_k(s,a)$, $N_k(s,a)$ and V_f are the known value information for the known node corresponding to action a under state s. $V_p(s,a)$, $N_p(s,a)$ and V_f are the potential value information for the potential node corresponding to action a under state s. Their definitions are similar, with the former representing the average V_k or average V_p, while the latter introduces V_f to indicate whether the node is selectable or not. Specifically:

i) When $V_f = 1$, it indicates that the prefix represented by this node is already an FRP. Therefore, this node should not be selected, and this branch will not be probed, as its child nodes must also be FRP. At this point, based on the above function, it can be determined that $Q = 0$, indicates that the node will not be selected during the probing process.

ii) When $V_f = 0$ and 2, it means that the node is NFRP and unknown, and both of them will be possible to be selected. In this case, $|V_f - 1| = 1$, which does not affect the calculation of Q. Note that a node with $V_f = 0$ simply means that the node itself is an NFRP, but it does not mean that all its children are also NFRP. Therefore, it may still be selected, i.e., continue exploring this branch downwards. However, it is necessary to ensure that it is not the last node to be recursively selected.

Moreover, in our scenario, a larger prefix of an FRP may also be an FRP (i.e., the parent node of FRP). If we keep selecting leaf nodes for probing, it may lead to a lot of wasted probing resources. Therefore, in state s, the last recursively selected node is all nodes with $V_f = 2$ in the tree, not just the leaf nodes with $V_f = 2$. At the same time, this also exactly avoids selecting the prefix represented by nodes with $V_f = 0$. To realize the selection of non-leaf nodes, we design a simple but effective method. When there are more than n FRP among the child nodes of a parent node and no NFRP, we prioritize selecting the parent node to determine if it is an FRP, to avoid wasting probing resources. As shown in Fig. 3, it illustrates a possible selection path (the search tree is from Fig. 2). Typically, there is a bias toward *exploitation*, i.e., ϵ is set smaller, as selecting based on Q_k often leads to higher real rewards. Therefore, the node with the maximum Q_k among the known nodes in each layer will be selected with a higher probability, i.e., node "1", "f"

is selected. Meanwhile, since node "9" represents the seed prefix (i.e., $Q_k = 0$) and cannot be selected, the potential node "8" with the maximum Q_p is selected. Note that when there is more than one node corresponding to the maximum Q, it is sufficient to select randomly from them. Finally, when the target prefix is 2003:89af:0011:/48, a candidate FRP prefix 2003:89af:0011:1f80::/60 consisting of selected nodes is generated (i.e., prefix represented by node "8").

2) Probing: The aim is to expand the children of the selected node and detect whether the node is FRP or not. For the selected node, we first expand its child nodes, i.e., a total of 16 child nodes from 0 to f are created. Meanwhile, we also continue to randomly expand the 16 addresses based on the selected nodes and the expanded child nodes. Next, these 16 addresses are scanned online by sending ICMPv6 Echo Requests to determine whether they are responsive (IV-A3), as the states of the selected nodes and possible states of the child nodes are reflected by these 16 addresses, according to the APD detection method. As shown in Fig. 3, based on the selected "1f8", the 16 addresses obtained by the expansion are $2003:89af:0011:1f8\{0-f\}:****:****:****:****$. And it may be that just address $2003:89af:0011:1f8\{0-1\}:****:****:****:****$ is responsive.

3) Backpropagation: The purpose of this step is to update the value information of nodes (i.e., state of the tree) based on the response status of the 16 addresses in the probing, starting from the expanded 16 child nodes in reverse, layer by layer, until reaching the root node. It is important to note here that once a potential node has been selected and the probing has been executed, it will be transformed into a known node. Before performing the backpropagation, its value information will be transformed from V_p and N_p to V_k and N_k and initialized to 0. Based to the APD method, there will be three scenarios as follows:

- **Case 1:** when all 16 addresses have responses, the selected node is an FRP. At this time, consistent with the backpropagation in search tree construction (§III-B), the V_k and N_k of all nodes on the selected path will be increased by v and 1, respectively. And the corresponding V_p (i.e., $v--$) is propagated to the potential nodes. Furthermore, the V_f of the selected node is set to 1, and the expanded child nodes will be pruned.
- **Case 2:** when 16 addresses are partially responsive (e.g, only m address responses), the V_k and N_k of all nodes on the selected path will be increased by m and 1, respectively. The V_p and N_p of the responsive child nodes is added 1, and V_p of the unresponsive child nodes is 0 but N_p plus 1.
- **Case 3:** when all 16 addresses are unresponsive, the V_k of the selected node is 0 and does not back propagate, but the N_k of all nodes on the selected path will be increased by 1. Similarly, the V_p of all child nodes is 0 but N_p plus 1.

Furthermore, for Case 2 and Case 3, the V_f of the selected node is set to 0 and the V_f of its all expanded child nodes is set to 2. After the reverse value information update, the probing feedback is propagated to the whole prefix space to optimize the next probing. As shown in Fig. 3, based on the probing results (§III-C2), the V_k of the selected node "8" is updated to 2 and back propagated to the root node, and its V_f is updated to 0. The V_p and N_p of "0" and "1" in the expanded child nodes is updated to 1. The V_p of other expanded child nodes is 0. The V_f of all expanded child nodes is 2.

IV. Evaluation

A. Experimental Setup

1) Seed Datasets: To evaluate the performance of *Luori*, we first obtained an Internet-wide total of 211,037 announced IPv6 routing prefixes from the RouteViews project [39] and CAIDA [42]. The initial FRP detection for these prefixes by APD reveals 5302 FRRPs. Further, we select the latest FRPlist (May 4, 2023) provided by a well-known IPv6 hitlist service [26], [28], [29] as seeds, which contains 5.1M FRPs from active address probing. However, it is just 1.03M after taking the largest. Meanwhile, 4,703 of them belong to known 5302 FRRPs. Therefore, the final seed dataset used for evaluation contains 1.02M FRPs covering 1858 prefixes (i.e., seeded prefixes), and thus the number of unseeded prefixes is 203,877.

2) Evaluation Metrics: The goal of FRP probing is to discover as many of the FRPs as possible among a certain B. However, the number of FRPs does not accurately reflect the superiority of a method because different prefixes vary in size. For instance, if the largest FRP is not probed, many sub-prefixes of this FRP may be found, leading to an inflated count of FRPs. To fully evaluate the performance of *Luori*, We design the following four metrics:

- S_{FRP}: Address space (i.e., number of addresses) covered by the probed largest FRPs (avoid co-existence of a prefix and its sub-prefixes in probed FRPs), e.g., the value of S_{FRP} for a "/32" FRP is 2^{128-32}. S_{FRP} not only avoids the nuisance of varying sizes of different prefixes, but it is also in line with the original intention of studying FRPs, which is to have an impact on other measurement activities due to the vast address space covered by FRPs.
- N_{AS} *and* N_{RP}: Number of ASes and routing prefixes covered by the probed FRPs. Since FRPs are widely distributed across the Internet, it is also important to be able to probe FRPs under more ASes or routing prefixes.
- R_o: Ratio of the S_{FRP} where *Luori* and the baseline method overlap to their respective S_{FRP}, which can effectively evaluate the differences in probing results between *Luori* and the baseline method.

3) Implementation: Since there is no Internet-wide work on active probing of FRPs, *Luori* is compared to MAPD [28], a brute force probing method (§II). We have implemented prototypes of *Luori* and MAPD with Python. To improve the efficiency of *Luori*, we also implemented multiprocessing parallel execution of the search tree construction and FRP probing modules. For MAPD, we directly enumerate the sub-prefixes with lengths greater than or equal to 64 under each routing prefix as candidate FRPs for probing. Finally, they were all deployed on a server with 12 $\times$ Intel(R) Xeon(R) CPU E5-2650 v3 @ 2.30GHz, 128G RAM and 100Mbps

TABLE I
PERFORMANCES OF *Luori* AND MAPD (B IS THE BUDGET FOR EACH TARGET PREFIX AND #FRP IS THE NUMBER OF LARGEST PREFIXES PROBED)

Target Prefixes	seeded prefixes (Number: 1858)						unseeded prefixes (Number: 203,877)			
Metric	$B=200$		$B=400$		$B=600$		$B=20$		$B=50$	
	Luori	MAPD	***Luori***	MAPD	***Luori***	MAPD	***Luori***	MAPD	***Luori***	MAPD
S_{FRP}	1.1×10^{29}	3.4×10^{23}	4.9×10^{29}	6.8×10^{23}	9.5×10^{29}	1.0×10^{24}	1.7×10^{29}	1.6×10^{22}	5.4×10^{29}	4×10^{22}
N_{AS}	129	60	129	69	180	68	37	30	68	44
N_{RP}	695	554	713	612	1075	645	272	246	378	432
R_o	≈ 0	5.3%	≈ 0	7.4%	≈ 0	12.1%	≈ 0	10.4%	≈ 0	13.5%
#FRPs	8640	18,373	12,702	36,839	29,260	54,568	1126	876	2424	2166

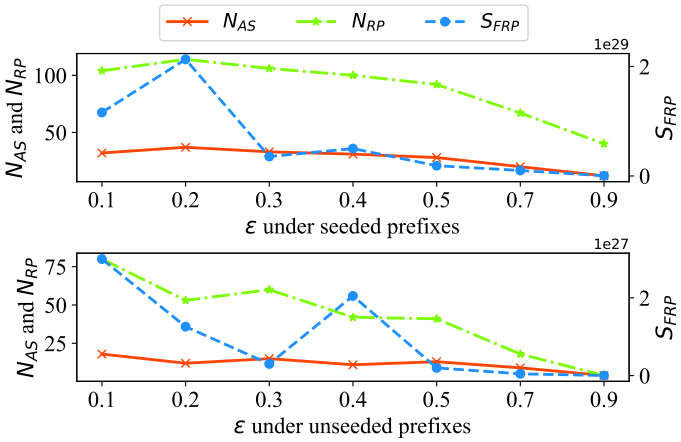

Fig. 4. Performance of *Luori* under different ϵ values

Bandwidth. In addition, this paper uses ICMPv6 probes to verify the liveness of IPv6 addresses (16 addresses under each candidate FRP). This is mainly because existing efforts [14], [19], [26] have shown that using ICMPv6 probes can discover more IPv6 active addresses, i.e., FRPs can be verified from candidate FRPs with a higher probability.

4) Hyper-parameters: In search tree construction, since the FRP is judged to pass 16 random addresses belonging to its sub-prefixes, v is set to 16. This also ensures that the Q_p can be propagated to any node. In FRP probing, n is set to 3 based on experience. Under seeded and unseeded prefix, ϵ is set to 0.2 and 0.1, respectively, which is discussed in §IV-C. In addition, during parallel execution, we set the maximum number of process pools to 200.

B. *Performance of* Luori

Table I shows the S_{FRP}, N_{AS}, N_{RP} and R_o for all methods under different B, including seeded prefixes and unseeded prefixes. Furthermore, to demonstrate that the number of FRPs (#FRPs) does not reflect the performance of a method, we also show the #FRPs for all methods. It is evident that *Luori* outperforms the brute-force probing method MAPD. Under all target prefixes and all B values, we observe that *Luori* probes the highest S_{FRP} and N_{AS}, which are 10^6-$10^7\times$ and 1.2-2.6$\times$ higher than MAPD, respectively. This is mainly due to the ability of *Luori* to probe FRPs of any length, and to prioritize the probing of larger FRPs based on feedback. This can also be proven from #FRP. Although the #FRPs of *Luori* is smaller than the MAPD, the probed FRPs are larger, which covers a larger address space. Meanwhile, *Luori* probed FRPs covering more AS, particularly under the seeded prefixes, mainly due to the design of the ϵ-Q strategy (exploration and exploitation) and the continuous adjustment of the probing direction based on the feedback to find FRPs in more space. N_{RP} of *Luori* is larger than MAPD for most of B, and smaller than MAPD only for unseeded prefixes with B=50. We explore the reason for this, which is mainly due to the bias of the seeds in the search tree, because the dataset of Hitlist mainly consists of /64 prefixes (see §IV-D for details). For the case of biased seeds, we believe it will gradually improve with long-term probing (i.e., continuous enrichment of the seeds dataset).

In addition, under all B values, the R_o of *Luori* is approximately 0, while the R_o of MAPD gradually increases with the increase of B. This indicates that some of the FRPs probed by MAPD are included in the probing results of *Luori*. *Luori* prioritizes probing larger FRPs within the budget, which means it might not have searched for smaller FRPs (e.g., /64 prefixes) when the budget is exhausted. MAPD can only brute-force probe fixed-length FRPs (i.e., enumerate /64 to /124 prefixes). Therefore, the R_o of MAPD is not particularly high. However, based on the trend that R_o increases as B grows, it can be inferred that when the budget is sufficient, the probing results of *Luori* will gradually include the probing results of MAPD. Finally, we also analyzed the efficiency of *Luori*. Specifically, during the search tree construction process, based on the seed dataset containing 1.02M FRPs (§IV-A1), *Luori* spends about 2.5 hours, which is an offline process. During the FRP probing process, when the B for seeded and unseeded prefixes are 400 and 20 respectively, probing a seeded prefix and an unseeded prefix on average takes approximately 3.6 seconds and 1.13 seconds, respectively. Overall, based on just one server (§IV-A3), *Luori* can complete a scan of all Internet-wide routing prefixes (211K) within 3 days (even faster by increasing the number of processes). The efficiency is perfectly acceptable for Internet-wide measurement campaigns. Note that the budgets for seeded and unseeded prefixes are different. First, the likelihood of finding FRPs under unseeded prefixes is lower, so spending excessive probing resources at once may yield little return. Second, this helps address measurement ethics concerns. Since there are a large number of unseeded prefixes (204K), a high budget could consume a significant amount of network resources.

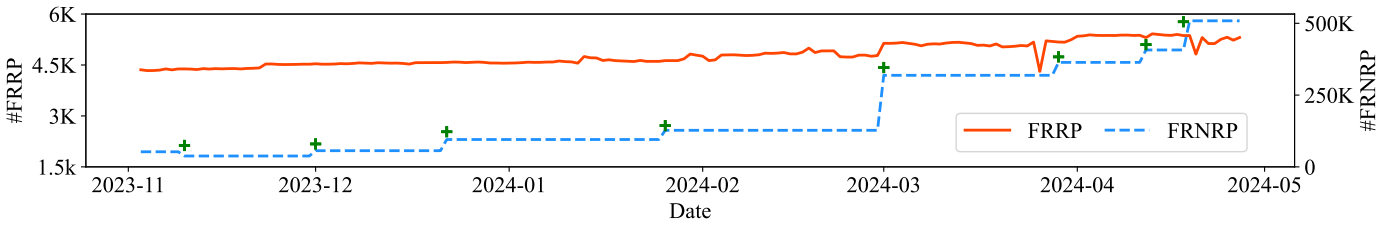

Fig. 5. Changes in the number of FRRPs and FRNRPs with iterative probing, where each "+" indicating the merging of newly probed FRNRPs into FRPlist

TABLE II
COMPARISON OF OUR ***Our FRPlist*** AND HITLIST'S FRPLIST (#FRPS=#FRRPS+#FRNRPS)

Dataset	Publication time	S_{FRP}	N_{AS}	N_{RP}	#FRPs	#FRRPs	#FRNRPs
Our FRPlist	April 30, 2024	1.3×10^{33}	795	9182	0.52M	7079	0.51M
Hitlist's FRPlist	May 4, 2024	1.2×10^{32}	629	6561	1.03M	4671	1.03M

C. Effect of Hyper-parameters

The constant ϵ ($\in [0, 1]$) is used to adjust the balance between *exploitation* and *exploration* in the ϵ-Q strategy. Generally, it is set to a small value, leaning towards utilizing existing knowledge for *exploitation*. Due to the difference in a priori knowledge between seeded and unseeded prefixes, to achieve optimal performance of *Luori*, we conducted experiments by sampling 10% from seeded and unseeded prefixes multiple times, and the budgets were set to 400 and 20, respectively. As shown in Fig. 4, under seeded and unseeded prefixes, when $\epsilon = 0.2$ and 0.1, respectively, S_{FRP}, N_{AS} and N_{RP} of *Luori* are achieved to their maximum values. Therefore, we set ϵ to be 0.2 and 0.1 under seeded and unseeded prefixes in this paper, respectively.

D. Performance of Luori for Long-Term Probing

As of now, we have accumulated an ***FRPlist*** containing 515,739 largest FRPs, which covers 1.3×10^{33} address space, equivalent to the address space covered by 7.1×10^{13} /64 prefixes. In addition, in this FRPlist, there are 7079 FRRPs (FRRPlist) and 508,660 fully responsive non-routing prefixes (FRNRPlist), which covers 795 ASes and 9182 routing prefixes in total. Fig. 5 shows the long-term iterative probing results of *Luori* from November 2023 to April 2024, i.e., the change in the number of FRRPs and FRNRPs in FRPlist, where each "+" indicating the merging of newly probed FRNRPs into FRPlist (taking the largest). The seeds for the first probe come from the FRPs found in the active address probing (about 52K), and the seeds in each subsequent probe are collected from the previous probe. During these six months, we can observe that the overall quantities of both FRRPs and FRNRPs are continuously increasing, with FRRP increasing from 4.5K to 5.5K, and FRNRP increasing from 52K to 500K (almost 10×). Notably, for FRNRPs, after the first probing, the quantity of FRNFRs actually decreased. This is mainly because *Luori* discovered that the larger prefix of the seeds is FRP, so the quantity decreased after merging. But it can be seen that this situation did not occur later, which thanks to *Luori*'s ability to prioritize probing larger FRPs. Additionally, the probing results of FRRPs come from traversing scans of Internet-wide all routing prefixes announced by BGP. Therefore, the continued growth of FRRPs is more evidence that FRPs will become more and more common in the future. This not only implies that the impact of FRP on IPv6 measurement campaigns will be even more significant, but further highlights the value of FRP's potential applications.

E. Comparison of Our FRPlist and Hitlist's FRPlist

To further analyze our FRPlist, we also compare it with the only publicly available latest FRPlist in the Hitlist service (§IV-A1), which was accumulated over time through MAPD during active address probing, with a longer accumulation time than ours. Note that we consider the Hitlist's FRPlist to be cumulative because the FRPlist under different times contains a large number of duplicate FRPs. Therefore, we chose its latest FRPlist for comparative analysis. As shown in the Table II, although the number of the largest FRPs in Hitlist's FRPlist is nearly double that of our FRPlist, our FRPlist covers 10× more address space. This is mainly because MAPD cannot probing the largest FRPs, i.e., Hitlist's FRPlist contains a large number of sub-prefixes of larger FRPs. Additionally, the number of ASes and routing prefixes it covers is also smaller than that of our FRPlist, which is mainly because *Luori* achieves Internet-wide active probing. Overall, Hitlist's FRPlist is sourced from active address probing, i.e., it is not Internet-wide probing. At the same time, the MAPD probing method it adopts has a large limitation.

As shown in Fig. 6, we further analyze the length distribution of FRPs in our FRPlist and Hitlist's FRPlist. For the FRRPs in both lists, most of their lengths are less than or equal to 48, especially /48 prefixes account for most of them. However, our FRPlist contains more larger FRPs (i.e., length≤47), which benefit from the results of our long-term probing of Internet-wide all routing prefixes. More obviously, the length distribution of FRNRPs in our FRPlist is diverse (/28-/80), and it also contains more larger FRPs (e.g., length≤64). However, the length of FRNRPs in the Hitlist's FRPlist is basically 64. This is mainly because *Luori* can probe FRPs of arbitrary length based on search trees and find the largest FRPs as preferentially as possible. However, when Hitlist collects FRPs based on active addresses by MAPD, which only detects prefixes that are smaller than /64 and contain more than 100 active addresses under them. The above shows why our FRPlist contains a larger address space and is

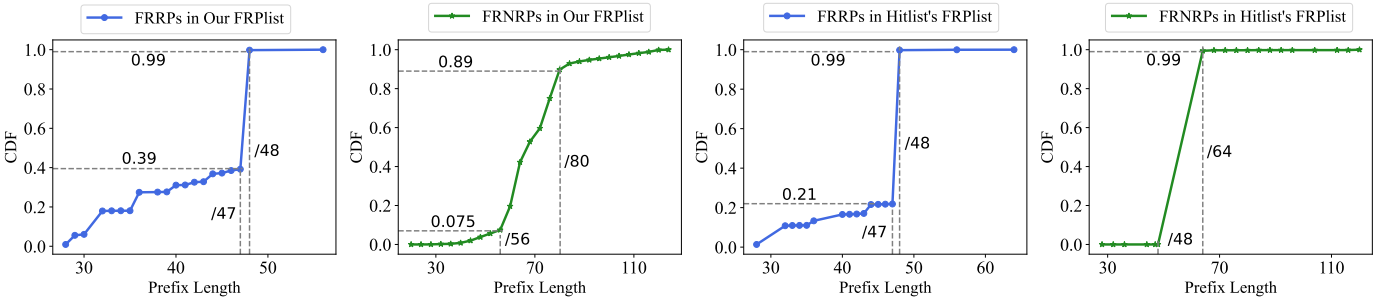

Fig. 6. CDF of the length of FRPs in ***Our FRPlist*** and Hitlist's FRPlist

more widely distributed. Meanwhile, this also demonstrates the effectiveness of *Luori* and the limitations of collecting FRPs based on IPv6 active address probing.

Key takeaways: By comparison with the only publicly available FRPlist from Hitlist, our FRPlist covers 10× more address space and also covers more Internet-wide ASes and routing prefixes. Therefore, our FRPlist should be **the largest publicly known FRPlist** to date.

F. Distribution of FRPs.

Our FRPlist demonstrates that IPv6 FRPs have become a common phenomenon. To further study their characteristics and possible usage scenarios, we investigate which ASes, organizations announce these FRPs. Table III shows the information related to the 10 ASes with the highest number of FRPs in FRPlist, FRRPlist and FRNRPlist, respectively, including the number of FRPs, the main open TCP/Ports and the organization to which the AS belongs. Firstly, it can be observed that there is a significant difference in the distribution of FRPs among ASes, with the majority concentrated in a few ASes. For example, the top 10 ASes in FRPlist account for 89% of the total FRPs. This indicates that there may be some specific usage scenarios for most Internet-wide FRPs.

To further investigate the reasons for the occurrence of FRPs in these ASes, we analyzed the organizations to which these ASes belonged. The results show that ASes with more FRPs are from large Cloud or CDN service providers, such as Imperva (Incapsula), Cloudflare, Amazon, and Akamai. We infer that FRPs may be a technical means that they need many network addresses to support their services efficiently. Couldflare's addressing agility approach [27], [38] is one of the relevant technical means, which utilizes FRPs to decouple IP addresses from domain names and services, improving addressing flexibility, i.e., authoritative name servers can select addresses in query responses from full prefixes. Therefore, all addresses within that prefix must respond. At the same time, we also found that the main open ports under these FRPs are 80/443 (i.e., HTTP/HTTPS), further confirming this point. We believe that other large Cloud or CDN service providers have implemented similar techniques, which is why there are a large number of FRPs under their ASes. In addition, the rest of the FRPs in FRPlist are mainly from about 650 different organizations [42], which also shows that FRPs have been widely used in practice. These organizations are mainly from Network Security, Online Music and Video Streaming Services, Internet Service Providers (ISPs), and so on [43], [44]. For Network Security, we found that Cloudflare Spectrum [27], [47]–[49] may be related to IPv6 FRPs, which are reverse TCP/UDP proxies used for DDoS protection. For Online Music and Video Streaming Services, this is one of the main applications of CDN networks that can provide high bandwidth and low latency services. For ISPs, we speculate that FRPs can be used as proxy services to assign random IPv6 addresses to users in real-time.

Key takeaways: These FRPs mainly come from 650 organizations in industries such as Cloud or CDN services, network security and ISPs, especially Cloud or CDN services playing a major role. This highlights their widespread adoption, especially in Cloud and CDN networks. This analysis can also help to further optimize *Luori*'s probing strategy to discover more FRPs, such as focusing on Cloud and CDN networks.

G. Email Verification

To further analyze the probed FRPs and verify their authenticity, we contacted network managers of 280 different ASes (mailboxes are accessible and contain more FRPs). So far, we have received responses from only 7 ASes, one of which was unaware of such prefixes. However, both confirm the accuracy of the probed FRPs. In one AS (not belonging to Cloudflare), FRPs were identified as an implementation of a specific CDN technology, with each FRP not corresponding to the same host. Conversely, in five other ASes, FRPs were implemented based on the existing functionality of Linux systems, with each FRP corresponding to the same host. The mentioned implementation methods are as follows: (1) *net.ipv6.ip_nonlocal_bind.* It is a configuration option in the Linux kernel. When set to 1, it allows binding to non-local IPv6 addresses, i.e., FRPs are implemented. (2) *ip -6 route add local PREFIX dev IFACE.* This is a command to add an IPv6 local route on a Linux system for a specific IPv6 prefix on a network interface (IFACE). Furthermore, we investigated that the IP_FREEBIND option in Linux also enables multiple IPv6 addresses to be bound to the same host [28]. They also share usage scenarios for these FRPs: (1) Spreading requests over many IP addresses to address IP-based rate limiting in large systems. (2) FRPs used as proxy services (e.g., intranet proxies) to assign random IPv6 addresses to users.

Key takeaways: We sent emails to 280 ASes. Although only 7 replies were received, they verified the authenticity of our probed FRPs and our analysis of FRP usage scenarios

TABLE III
TOP 10 ASES BASED ON THE NUMBER OF FRPS IN FRPLIST, FRRPLIST, AND FRNRPLIST, RESPECTIVELY.

FRPlist				FRRPlist				FRNRPlist			
AS	#FRPs	Org	TCP/Port	AS	#FRPs	Org	TCP/Port	AS	#FRPs	Org	TCP/Port
19551	165K	Imperva	80/443/...	16509	1260	Amazon	80/443	19551	164K	Imperva	80/443/...
13335	99K	Cloudflare	80/443	36183	1219	Akamai	-	13335	98.6K	Cloudflare	80/443
16509	97.9K	Amazon	80/443	13335	366	Cloudflare	80/443	16509	96.7K	Amazon	80/443
20940	36.3K	Akamai	80/443/53	19551	312	Imperva	80/443/...	20940	36.3K	Akamai	80/443/53
131662	26.1K	Denpa	22	204916	248	RACKTECH	80/53	131662	26.1K	Denpa	22
204916	18.4K	RACKTECH	80	36492	237	Google	-	204916	18.1K	RACKTECH	53/80
32934	9.64K	Facebook	-	210842	222	Rohmad Kumiadin	22	32934	9.64K	Facebook	-
46997	3.75K	Black Mesa	22	54113	171	Fastly	80/443	46997	3.75K	Black Mesa	22
28573	2.36K	Claro NXT	-	45609	167	Bharti Airtel	-	28573	2.35K	Claro NXT	-
36183	2.24K	Akamai	-	216157	163	SIXNET	22	36492	1.97K	Google	-

(§IV-F), e.g., load balancing of IP addresses in large systems, proxy services. In addition, the implementation of FRPs is diverse, except for CDNs, most of them are based on the existing functionality implemented in Linux systems (e.g., net.ipv6.ip_nonlocal_bind option), and each of their FRPs corresponds to the same host.

H. Impact of Our FRPlist

FRPlist is crucial for IPv6 network measurement campaigns, which can effectively reduce wasted probing resources and minimize biases in the results. The above analysis reinforces this view, which is mainly reflected in the fact that FRPs are widely distributed and cover a vast address space, but these addresses do not each correspond to a real responder behind them. Therefore, it is very meaningful to build an open-source FRPlist. Additionally, this helps prevent various researchers and organizations from conducting redundant probing processes, thus alleviating unnecessary strain on the network. Despite the Hitlist publishing previous collections of FRPlists from active address probing, they are not comprehensive and do not fully satisfy the requirements.

To prove this point, we utilized the latest active address dataset (May 4, 2024) from the Hitlist for our experiments. The dataset contains 24.5M active addresses, which have been filtered using the Hitlist's FRPlist. However, after comparing it to our FRPlist, we discovered that 4.8 million addresses still fell under FRPs, constituting approximately 20% of the total. This indicates a waste of around 20% of probing resources. Crucially, the IPv6 active address dataset in the Hitlist is a widely utilized resource in the realm of IPv6. It is frequently employed as a foundational dataset for subsequent research in areas like active address probing [13]–[16], [19]–[22], port probing [30], [31] and security analysis [31], [50]. However, when we are unaware of the presence of addresses under FRPs, this may lead to some unknown measurement bias or even wrong conclusions. As an intuitive example, when further asset probing is performed based on these addresses, we may find a much higher amount of assets than in the real network.

Key takeaways: IPv6 FRPs have resulted in a massive amount of fake active addresses (i.e., addresses under FRPs) in the network, e.g., the well-known and commonly used Hitlist's active address dataset contains 20% of such addresses, significantly impacting all measurement activities based on this dataset. Therefore, it is essential to actively probe Internet-wide FRPs to reveal this impact and to construct a long-maintained FRPlist for use by relevant researchers.

V. ETHICAL CONSIDERATIONS

To achieve Internet-wide probing of FRPs, we follow ethical conventions for network measurement [51], [52]. More importantly, *Luori*'s probing process is conducted iteratively (§III-C) to prevent the generation of a large volume of probing packets in a short period. We also cap the probing rate at 10 Mbps, significantly minimizing network impact. Furthermore, to avoid causing trouble to the probed network, we set up a web service on the default port of the probing host, which introduces our research and offers a way to halt the probing process if needed. Finally, we send only one packet per address during probing, rigorously implementing deduplication to prevent multiple probes of the same address.

VI. CONCLUSION

In this paper, we propose for the first time an active probing method for Internet-wide IPv6 FRPs, *Luori*, which realizes effective probing by representing diverse FRP patterns and the entire probing space as a search tree, and then based on reinforcement learning ideas, abstracts the probing process as a search process in the tree and achieves continuous optimization of the probing process. Experiments show that *Luori* can significantly discover 10^6-$10^7\times$ more address space (i.e., the number of addresses covered by FRPs) than the current brute-force probing method. More importantly, after six months of Internet-wide probing, we have constructed and open-sourced the largest known FRP list. Finally, based on this list, we not only evaluate its impact on IPv6 measurement campaigns but also analyze its usage scenarios and implementations, which provide a reference for the practical application of FRPs.

ACKNOWLEDGMENTS

We sincerely thank our shepherd Arvind Narayanan and the anonymous reviewers for their insightful comments and helpful feedback. This work is supported in part by the National Natural Science Foundation of China under Grant No. 62302253 and 61772307, the Zhongguancun Laboratory Project, the Beijing Natural Science Foundation under Grant No. 4222026, and NSFOCUS 20222910019. Lin He is the corresponding author of this paper.

REFERENCES

[1] Google, "Ipv6 adoption statistics," https://www.google.com/intl/en/ipv6/\statistics.html, 2024.

[2] Y. Liu, L. He, and G. Ren, "Gagms: A requirement-driven general address generation and management system," *Science China Information Sciences*, vol. 61, no. 9, pp. 092 109:1–092 109:15, 2018. [Online]. Available: https://doi.org/10.1007/s11432-017-9298-3

[3] 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 *Proceedings of the 30th Annual Network and Distributed System Security Symposium (NDSS)*. San Diego, California, USA: The Internet Society, 2023.

[4] Y. Liu, G. Ren, J. Wu, S. Zhang, L. He, and Y. Jia, "Building an ipv6 address generation and traceback system with nidtga in address driven network," *Science China Information Sciences*, vol. 58, no. 12, pp. 1–14, 2015. [Online]. Available: https://doi.org/10.1007/s11432-015-5461-0

[5] L. He, G. Ren, Y. Liu, and J. Yang, "Pavi: Bootstrapping accountability and privacy to ipv6 internet," *IEEE/ACM Transactions on Networking*, vol. 29, no. 2, pp. 695–708, 2021.

[6] E. Rye and D. Levin, "Ipv6 hitlists at scale: Be careful what you wish for," in *Proceedings of the ACM SIGCOMM 2023 Conference*, ser. ACM SIGCOMM '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 904–916. [Online]. Available: https://doi.org/10.1145/3603269.3604829

[7] 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.

[8] L. He, S. Wang, Y. Xu, P. Kuang, J. Cao, Y. Liu, X. Li, and S. Peng, "Enabling application-aware traffic engineering in ipv6 networks," *IEEE Network*, vol. 36, no. 2, pp. 42–49, 2022.

[9] E. C. Rye and R. Beverly, "Ipvseeyou: Exploiting leaked identifiers in ipv6 for street-level geolocation," in *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023, pp. 3129–3145.

[10] H. B. Tanveer, R. Singh, P. Pearce, and R. Nithyanand, "Glowing in the dark: Uncovering ipv6 address discovery and scanning strategies in the wild," in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023, pp. 6221–6237.

[11] L. He, G. Ren, Y. Liu, G. Song, J. E, J. Yang, and M. Xu, "Sav6: A novel inter-as source address validation protocol for ipv6 internet," *IEEE Network*, vol. 37, no. 5, pp. 64–70, 2023.

[12] L. Nie, L. He, G. Song, H. Gao, C. Li, Z. Wang, and J. Yang, "Towards a behavioral and privacy analysis of ecs for ipv6 dns resolvers," in *Proceedings of 18th IEEE Conference on Network and Service Management (CNSM)*, Thessaloniki, Greece, 2022, pp. 303–309.

[13] T. Yang, Z. Cai, B. Hou, and T. Zhou, "6forest: an ensemble learning-based approach to target generation for internet-wide ipv6 scanning," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 2022, pp. 1679–1688.

[14] 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.

[15] 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 *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 2021, pp. 1–10.

[16] 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.

[17] A. Murdock, F. Li, P. Bramsen, Z. Durumeric, and V. Paxson, "Target generation for internet-wide ipv6 scanning," in *Proceedings of the 2017 Internet Measurement Conference*, 2017, pp. 242–253.

[18] P. Foremski, D. Plonka, and A. Berger, "Entropy/ip: Uncovering structure in ipv6 addresses," in *Proceedings of the 2016 Internet Measurement Conference*, 2016, pp. 167–181.

[19] G. Song, J. Yang, Z. Wang, L. He, J. Lin, L. Pan, C. Duan, and X. Quan, "Det: Enabling efficient probing of ipv6 active addresses," *IEEE/ACM Transactions on Networking*, vol. 30, no. 4, pp. 1629–1643, 2022.

[20] 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 *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 2021, pp. 1–10.

[21] G. Song, J. Yang, L. He, Z. Wang, G. Li, C. Duan, Y. Liu, and Z. Sun, "Addrminer: A comprehensive global active ipv6 address discovery system," in *2022 USENIX Annual Technical Conference (USENIX ATC 22)*, 2022, pp. 309–326.

[22] 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.

[23] G. Song, L. He, F. Zhu, J. Lin, W. Zhang, L. Fan, C. Li, Z. Wang, and J. Yang, "Addrminer: A fast, efficient, and comprehensive global active ipv6 address detection system," *IEEE/ACM Transactions on Networking*, 2024.

[24] G. Song, L. He, Z. Wang, J. Yang, T. Jin, J. Liu, and G. Li, "Towards the construction of global ipv6 hitlist and efficient probing of ipv6 address space," in *2020 IEEE/ACM 28th International Symposium on Quality of Service (IWQoS)*, 2020, pp. 1–10.

[25] W. Zhang, G. Song, L. He, J. Lin, S. Wu, Z. Wang, C. Li, and J. Yang, "6vision: Image-encoding-based ipv6 target generation in few-seed scenarios," in *Proceedings of the 32nd IEEE International Conference on Network Protocols (ICNP)*, 2024.

[26] J. Zirngibl, L. Steger, P. Sattler, O. Gasser, and G. Carle, "Rusty clusters? dusting an ipv6 research foundation," in *Proceedings of the 2022 Internet Measurement Conference*. New York, NY, USA: ACM, 2022.

[27] P. Sattler, J. Zirngibl, M. Jonker, O. Gasser, G. Carle, and R. Holz, "Packed to the brim: Investigating the impact of highly responsive prefixes on internet-wide measurement campaigns," *Proceedings of the ACM on Networking*, vol. 1, no. CoNEXT3, pp. 1–21, 2023.

[28] 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 *Proceedings of the 2018 Internet Measurement Conference*. New York, NY, USA: ACM, 2018.

[29] L. Steger, L. Kuang, J. Zirngibl, G. Carle, and O. Gasser, "Target acquired? evaluating target generation algorithms for ipv6," in *Proceedings of the Network Traffic Measurement and Analysis Conference (TMA)*, Jun. 2023.

[30] A. Maghsoudlou, O. Gasser, and A. Feldmann, "Zeroing in on port 0 traffic in the wild," in *Passive and Active Measurement: 22nd International Conference, PAM 2021, Virtual Event, March 29–April 1, 2021, Proceedings 22*. Springer, 2021, pp. 547–563.

[31] P. Jose, S. J. Saidi, and O. Gasser, "Analyzing iot hosts in the ipv6 internet," 2023.

[32] G. Song, L. He, T. Zhao, Y. Luo, Y. Wu, L. Fan, C. Li, Z. Wang, and J. Yang, "Which doors are open: Reinforcement learning-based internet-wide port scanning," in *Proceedings of the 31st IEEE/ACM International Symposium on Quality of Service (IWQoS)*, Orlando, FL, USA, 2023, pp. 1–10.

[33] R. Beverly, "Yarrp'ing the internet: Randomized high-speed active topology discovery," in *Proceedings of the 2016 Internet Measurement Conference*, 2016, pp. 413–420.

[34] R. Beverly, R. Durairajan, D. Plonka, and J. P. Rohrer, "In the ip of the beholder: Strategies for active ipv6 topology discovery," in *Proceedings of the Internet Measurement Conference 2018*, 2018, pp. 308–321.

[35] X. Li, B. Liu, X. Zheng, H. Duan, Q. Li, and Y. Huang, "Fast ipv6 network periphery discovery and security implications," in *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2021, pp. 88–100.

[36] Z. Durumeric, E. Wustrow, and J. A. Halderman, "Zmap: fast internet-wide scanning and its security applications," in *22nd USENIX Security Symposium (USENIX Security 13)*, 2013, pp. 605–620.

[37] R. D. Graham, "Masscan: Mass ip port scanner," https://github.com/robertdavidgraham/masscan, 2023.

[38] M. Fayed, L. Bauer, V. Giotsas, S. Kerola, M. Majkowski, P. Odintsov, J. Sitnicki, T. Chung, D. Levin, A. Mislove *et al.*, "The ties that unbind: Decoupling ip from web services and sockets for robust addressing agility at cdn-scale," in *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, 2021, pp. 433–446.

[39] Routeviews, "University of oregon route views project," https://www.routeviews.org/routeviews/, 2024.

[40] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, "A survey of monte carlo tree search methods," *IEEE Transactions on Computational Intelligence and AI in games*, vol. 4, no. 1, pp. 1–43, 2012.

[41] M. Świechowski, K. Godlewski, B. Sawicki, and J. Mańdziuk, "Monte carlo tree search: A review of recent modifications and applications," *Artificial Intelligence Review*, vol. 56, no. 3, pp. 2497–2562, 2023.
[42] CAIDA, "Correspondence between as and organizations," https://publicdata.caida.org/datasets/as-organizations/, 2024.
[43] M. Ziv, L. Izhikevich, K. Ruth, K. Izhikevich, and Z. Durumeric, "Asdb: a system for classifying owners of autonomous systems," in *Proceedings of the 21st ACM Internet Measurement Conference*, 2021, pp. 703–719.
[44] ——, "Correspondence between as and business categories," https://asdb.stanford.edu/, 2023.
[45] C. Dann, Y. Mansour, M. Mohri, A. Sekhari, and K. Sridharan, "Guarantees for epsilon-greedy reinforcement learning with function approximation," in *International conference on machine learning*. PMLR, 2022, pp. 4666–4689.
[46] M. Tokic, "Adaptive ε-greedy exploration in reinforcement learning based on value differences," in *Annual conference on artificial intelligence*. Springer, 2010, pp. 203–210.
[47] Cloudflare, "Cloudflare spectrum," https://blog.cloudflare.com/spectrum/, 2024.
[48] ——, "Cloudflare spectrum product," https://www.cloudflare.com/products/cloudflare-spectrum/, 2024.
[49] ——, "Cloudflare spectrum network ports," https://developers.cloudflare.com/fundamentals/reference/network-ports/, 2024.
[50] M. S. Pour, C. Nader, K. Friday, and E. Bou-Harb, "A comprehensive survey of recent internet measurement techniques for cyber security," *Computers & Security*, vol. 128, p. 103123, 2023.
[51] E. Kenneally and D. Dittrich, "The menlo report: Ethical principles guiding information and communication technology research," *Available at SSRN 2445102*, 2012.
[52] C. Partridge and M. Allman, "Ethical considerations in network measurement papers," *Communications of the ACM*, vol. 59, no. 10, pp. 58–64, 2016.