

基于特征融合的IPv6地址空间动态探测算法

王泽昊¹, 李冬², 于俊清^{1,2}, 陈俊鑫¹, 刘陶然¹, 税帅策¹

(1. 华中科技大学网络与信息化办公室, 湖北 武汉 430074;

2. 华中科技大学网络空间安全学院, 湖北 武汉 430074)

摘要: IPv6巨大的地址空间使得传统IPv4暴力扫描技术不再可行, 当前主流的解决方案难以应对复杂多变的IPv6地址空间。针对IPv6网络的动态性, 提出一种基于特征融合的IPv6地址空间动态探测(feature-fused dynamic IPv6 probing, FFDIP)算法。FFDIP提取单个地址的编址特征对地址集进行分类, 利用分段平衡空间模式表示对分类后的地址集建模并生成目标地址, 并在探测过程中采取反馈机制动态调整探测资源的分配; 同时将存量地址的特征持久化, 通过特征融合算法将增量地址特征与存量地址特征相结合, 生成目标地址集合, 提高探测效率。为解决IPv6地址分配中的别名前缀问题, 设计一种前缀自适应IPv6探测适配器(prefix-adaptive IPv6 probing adapter, PAIPA)。将网络前缀格式化, 并采用别名前缀验证算法替换活跃地址探测算法, 改变了传统的基于“检测”和“验证”的被动探测机制, 实现了IPv6别名前缀的“主动探测”。实验结果表明: FFDIP在无存量地址场景和有存量地址场景均比DET、6Tree、Entropy/IP等算法有更高的命中率; 将PAIPA应用于DET和6Tree算法, 其在别名前缀探测中的命中率比在地址探测中有显著提升。

关键词: 地址探测; 种子地址; 别名前缀; 动态探测

中图分类号: TP393

文献标志码: A

DOI: 10.11959/j.issn.2096-109x.2025072

Feature-fusion-based IPv6 address-space probing algorithm

WANG Zehao¹, LI Dong², YU Junqing^{1,2}, CHEN Junxin¹, LIU Taoran¹, SHUI Shuaice¹

1. Network and Computation Center, Huazhong University of Science and Technology, Wuhan 430074, China

2. School of Cyber Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China

Abstract: The vast address space of IPv6 makes brute-force scanning techniques of traditional IPv4 networks no longer feasible. The current mainstream solution can not adapt to the complex and changeable IPv6 address space. In response to the dynamic nature of IPv6 networks, a feature-fused dynamic IPv6 probing (FFDIP) technology is proposed. FFDIP classifies addresses using individual address encoding features, models them via balanced spatial pattern representation to generate target addresses and dynamically adjusts resource allocation through feedback mechanisms during probing. By persisting stock address features, FFDIP integrates incremental addresses with stock data through feature fusion, to improve efficiency and reduce redundancy. Additionally, aiming at the problem of alias prefixes in IPv6, a prefix-adaptive IPv6 probing adapter (PAIPA) is designed. By formatting address prefixes and replacing the active address verification algorithm with an alias prefix verification algorithm, it

收稿日期: 2025-06-11; 修回日期: 2025-07-20

通信作者: 于俊清, yjqing@hust.edu.cn

基金项目: 国家重点研发计划(2022YFB2901202)

Foundation Item: The National Key Research and Development Program of China (2022YFB2901202).

changes the traditional passive probing mechanism based on "detection" and "verification", realizes the "active probing" of IPv6 alias prefixes. Experiments show that FFDIP has a higher hit rate than DET, 6Tree and Entropy/IP in both non-stock address scenarios and stock address scenarios. When PAIPA is applied to the DET and 6Tree algorithms, the hit rates in alias prefix detection are higher than those in address detection. FFDIP and PAIPA can significantly improve the efficiency of IPv6 address space detection.

Keywords: address probing, seed address, alias prefix, dynamic probing

0 引言

网络地址探测旨在获取活跃设备的 IP 地址, 用于拓扑发现、入侵检测或网络攻击等, 在网络安全管理中发挥关键作用。随着 IPv4 地址资源枯竭, IPv6 网络大规模部署已成必然趋势。然而, IPv6 协议巨大的地址空间使传统 IPv4 暴力扫描技术完全失效: 研究表明, 即使采用当前最先进的扫描技术对整个 IPv6 地址空间穷举扫描, 也至少需要上亿年时间才能完成^[1]。

在现有技术框架下, 主流的方案是基于种子地址的 IPv6 探测^[1], 通过分析已知活跃的“种子地址”的特征进行建模, 生成高概率处于活跃状态的候选地址, 从而有效缩小探测范围, 显著提高探测效率。尽管该技术已取得显著进展, 但其仍存在诸多不足。

1) 难以适应动态变化的复杂 IPv6 网络环境。受设备上下线、防火墙策略等多种因素影响, IPv6 地址的活跃性是动态变化的。当前大部分算法的设计仅针对一次性地址输入, 无法利用动态的活跃地址特征进行高效探测。这种局限性导致系统在长期运行中需要频繁重新建模, 造成严重的计算资源浪费。

2) 缺乏主动的别名前缀探测方案。目前对别名前缀的检测算法都停留在被动检测和验证, 没有对别名前缀进行主动探测和处理。同时别名前缀数据集覆盖范围有限, 且存在严重的冗余问题。这种被动性导致大量潜在别名地址因未被纳入候选集而漏报, 直接影响生成地址空间的质量, 增加不必要的资源开销。

针对目前基于种子地址的 IPv6 地址探测方法在动态场景的适应性与探测效率方面的局限性, 提出一种动态高效的 IPv6 地址探测算法和一种主动的别名前缀探测算法。本文的核心贡献如下。

1) 基于单个 IPv6 地址的编址特征设计地址聚类 and 特征融合算法, 结合存量地址特征持久化技术, 提出了基于特征融合的 IPv6 地址空间动态探测 (feature-fused dynamic IPv6 probing, FFDIP) 算法, 显著提升动态地址空间中增量地址的处理效率, 减少重复建模的开销, 能够有效应对真实复杂网络环境中 IPv6 地址的动态变化。

2) 设计一种前缀自适应 IPv6 探测适配器 (prefix-adaptive IPv6 probing adapter, PAIPA), 通过网络前缀输入格式化, 以及依靠别名前缀分层随机验证算法, 实现别名前缀生成并主动探测, 摆脱传统被动别名发现的局限性。

3) 基于公开数据集在多种场景中测试发现, FFDIP 能有效提升增量地址探测效率, 相较于传统方法不论在命中率上还是在耗时上, 均有出色表现; 将 PAIPA 应用于现有算法进行别名前缀主动探测, 其在别名探测中的命中率明显高于在活跃地址中的命中率。

1 相关工作

本文工作的主要目的是提高 IPv6 地址空间的探测效率。国内外学术界对此已开展大量研究, 重点聚焦 IPv6 活跃地址探测与 IPv6 别名前缀探测两大核心方向。

(1) IPv6 活跃地址探测

基于种子地址的 IPv6 地址探测研究经历了 3 个发展阶段: 规则挖掘阶段、统计建模阶段和智能优化阶段。

规则挖掘阶段以种子地址间的潜在依赖关系为建模核心。在种子数据集中, 某个特定地址的存在会影响挖掘到的地址模式。目标生成算法 (target generation algorithm, TGA) 可利用这些依赖关系构建更有效的扫描模式。Ullrich 等^[2]提出了一种基于模式的 IPv6 地址探测技术, 可自动从种子地址集中挖掘每个比特的取值规律, 既避

免人工分析的局限，又能直接迁移到新的种子地址集。然而，该方法对种子地址集的质量要求较高：一旦种子地址集随机化程度增大，规则挖掘的误差率便急剧上升，导致建模空间碎片化。尽管其探测能力远逊于后续技术，他们的实验仍首次证明，自动化的启发式探测在效率上显著优于暴力扫描和人工分析。

在统计建模方案中，种子地址被视为真实活动地址的独立同分布随机样本。该方案遵循“密度一致性假设”，即地址空间中活跃地址密度较高的区域，其种子地址密度亦呈现同态分布特性。Foremski等^[3]提出Entropy/IP算法，将香农熵应用到IPv6地址模型构建，有助于读者理解地址结构，但生成候选地址命中率低。Murdock等^[4]提出了6Gen算法，该算法利用种子地址半字节取值密集区域生成潜在活跃地址，从而缓解地址分配随机化带来的负面影响，但其时间开销与内存消耗均较大。Liu等^[5]随后提出6Tree算法，通过分裂式层次聚类构建空间树，解决了6Gen算法在时间与内存上的性能瓶颈；然而，扫描与地址生成需相互反馈，导致整体耗时增加，且难以应用并发扫描技术。Song等^[6]在6Tree基础上改进，提出DET算法，利用香农熵选取分裂点，进一步提高了活跃地址探测的命中率。李果等^[7]提出的Multi-BSPR-Gen采用多层次分类与平衡空间建模，同样获得了较高的命中率，但预处理阶段耗时较长。在实际网络环境中，收集到的种子地址往往仅占实际活跃地址的极小部分，规则挖掘模型难以准确推断寻址模式^[8]，因此，相比之下，基于统计建模的方案可能更为合适。

6Tree算法的动态反馈调整机制在很大程度上启发了机器学习等技术在IPv6地址探测中的应用。Cui等^[9-10]提出6GCVAE与6GAN算法，分别利用变分自编码器和结合强化学习的生成对抗网络来生成地址，可直接从原始地址数据中习得其潜在分布，但存在mode collapse（模式崩溃）风险。Hou等^[11]提出6Hit算法，将强化学习用于目标地址生成；Yang等^[12-13]进一步提出基于集成学习的6Forest算法与基于密度聚类图切割的6Graph算法，可自动挖掘地址模式并适应

网络拓扑或地址规则的变化。Liu等^[14]提出6Former算法，基于Transformer的自注意力机制捕捉地址长距离依赖，实现地址生成。Song等^[15-16]提出AddrMiner与6EDL算法，分别针对有种子和无种子区域设计探测策略，拓展了地址探测边界。

然而，上述机器学习技术计算复杂度高且需要大量训练时间，难以扩展至大规模IPv6种子地址数据集；当训练集较小时，模型难以学到有效的活跃地址模式，导致命中率下降；当训练集增大时，高昂的计算开销又会显著增加训练时间，降低整体扫描效率。

(2) IPv6别名前缀探测

IPv6无状态地址自动配置允许设备根据路由器公告自动生成多个地址^[17]。IPv6别名地址指在同一网络接口上配置的多个IPv6地址，它们服务于不同的网络功能或通信需求。当某个网络前缀下的所有地址均被绑定到同一接口时，该前缀被称为别名前缀^[18]。大量别名前缀的存在会使研究者对IPv6网络空间结构产生误判，进而导致拓扑快照失真。此外，攻击者可利用别名地址为同一接口配置多个IPv6地址，从而绕过防火墙并发起攻击。对于依赖种子地址的IPv6探测算法而言，若别名地址大量进入活跃地址列表，将造成目标探测资源的极大浪费，并显著降低命中率。

Murdock等^[4]首先发现别名前缀问题，并提出一种简单的推测方法：对每个/96前缀随机生成3个地址进行探测，若3个地址均返回响应，则判定该前缀为别名前缀。该方法基于如下直觉：IPv6地址空间稀疏，给定一个/96前缀（ 2^{32} 个地址），在一个非别名前缀中随机选择3个地址均为活跃地址的概率是可以忽略不计的。Gasser等^[19]对此加以改进，提出多层次别名前缀检测算法：对待测前缀的16个子前缀各生成1个伪随机地址进行探测，确保探测均匀分布，从而提高了判定可靠性。6Tree与DET算法则采用活跃地址数阈值法，通过一小片地址空间内的请求-响应比例推测别名前缀的存在。此外，部分算法在生成目标地址后，利用已知的别名前缀列表进行过滤，以排除别名地址。

然而, 当前 IPv6 地址空间探测仍缺乏面向活跃地址动态变化的探测机制, 亦缺少别名前缀的主动探测手段。本文从以下两方面进行改进。

1) 针对 IPv6 活跃地址探测, 考虑网络动态性, 利用单个地址的编址特征设计二级分类方案; 通过特征融合算法将新增地址与存量地址相结合, 实现动态更新。同时, 引入平衡空间建模与多轮探测反馈机制, 合理分配探测资源, 提升探测效率。

2) 针对 IPv6 别名前缀现象, 对网络前缀进行格式化, 并设计别名前缀验证算法, 使大多数基于种子地址的探测方法可直接应用于别名前缀探测, 从而实现别名前缀的主动探测。

2 IPv6 活跃地址探测

基于种子地址的 IPv6 地址探测流程如图 1 所示, 首先利用聚类算法将地址划分为不同空间节点; 随后统计各节点地址在不同维度的取值, 并采用特定策略扩展可变维度, 完成空间建模; 最后依据建模结果生成目标地址, 进行活跃性验证, 部分算法还会进一步过滤别名地址, 得到最终活跃地址。

在单向探测框架下, 一旦引入高并发机制, 即可通过并行化方式同时对不同地址空间进行探测, 从而高效完成单轮任务。然而, 该框架的显著弊端在于: 空间建模严重依赖种子地址的初始分布, 极易造成探测资源分配失衡, 进而降低探测效率与命中率。为此, 6Tree 算法引入多轮反馈策略, 根据上一轮探测结果动态调整下一轮资源分配, 有效缓解了种子分布不均

带来的误差。

然而, 上述两种框架面临的共同问题是: 在动态网络环境下, 其探测流程仅针对一次性输入的活跃地址集。当网络环境发生变化、新的活跃地址出现时, 系统既难以高效复用存量地址已提取的特征, 又不得不将新增地址与存量地址全量合并后重新聚类建模, 导致每次都要对全部存量地址重新计算, 造成严重计算资源浪费。

为此, 本文提出 FFDIP 算法。初始阶段, 利用单个地址的编址特征聚类构建空间树, 并将节点特征持久化存储。当新增活跃地址到达时, 系统提取其一级/二级独立特征, 并将其映射至已有的对应节点; 后续建模与探测仅在该局部进行。若某一级节点的新增地址数达到阈值, 则仅对该节点进行子树重构, 避免大范围特征提取与聚类。通过特征融合算法, 算法以较低代价复用存量特征, 适应动态变化的 IPv6 地址空间; 同时在增量探测中利用已有特征定位活跃地址高密度区域, 从而提高命中率并减少资源开销。

本文提出的 FFDIP 算法将 IPv6 地址视为 32 维向量, 采用特征融合的动态地址空间划分、分段平衡空间建模与动态反馈机制。算法核心思想是: 在同一地址空间内, 前一轮输入的活跃地址特征可被新一轮直接复用; 对新增地址仅提取其编址特征并融合至已有空间树, 即可生成更多目标地址, 且仅处理增量数据, 保证高效性。同时, 已探测的存量地址空间已持久化存储, 可短期内避免重复探测, 减少资源浪费。IPv6 活跃地址动态探测框架如图 2 所示。

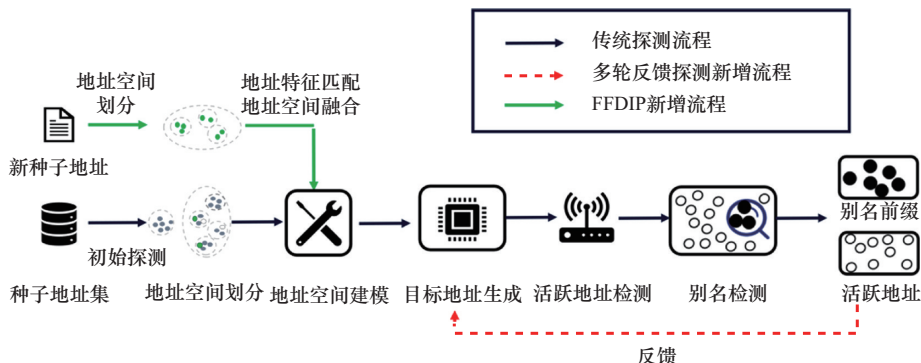


图1 基于种子地址的IPv6地址探测流程

隔。按 LOG_Q 从高到低遍历, 若相邻地址的 LOG_Q 差值大于间隔, 则在此切分, 形成新的二级节点。具体过程如算法 1 所示。

表 1 IPv6 地址后缀取值量化指标示例

后缀示例	后缀长度	r	Q	LOG _Q
1231:0000:0000:0000:0000	20	4	4	1.5
1234:1234:5678:9abc:def0	20	16	0.125	0.25
1111:2222:3333:4444:5555	20	5	0.8	0.92
0000:0000:0000:0000	16	1	16	2.0
1234:5678:9abd:def0	16	16	0.0625	0.0
1111:2222:3333:4444	16	4	1	1.0

算法 1 地址空间划分算法

输入 original_addresses[] //已归类到一级节点的地址集合

输出 first_level_nodes[] //地址划分后构建的一级节点类

```

1) function BUILD_NODE(addresses, feature1):
2)   node = new FirstLevelNode(feature1, len(addresses))
3)   splitpoints = []
4)   interval = h_rate * (max_LOG_Q - min_LOG_Q)
5)   sorted(addresses, reverse = True)
6)   for i in range(len(addresses) - 1):
7)     if (addresses[i].LOG_Q - addresses[i+1].LOG_Q) > interval:
8)       splitpoints.expand(i)
9)   node.second_level_node = SPLIT(addresses, splitpoints)
10)  return node
11) function TWO_LEVEL_CLASSIFICATION(original_addresses):
12)  //一级分类
13)  iidtype_to_addresses = CLASSIFY_BY_IIDTYPE(original_addresses)
14)  for iidtype, addresses in iidtype_to_addresses.items():

```

```

15)    feature1_to_addresses = CLASSIFY_BY_BGP(iidtype, addresses)
16)    //二级分类
17)    ex_nodes = GET_FIRSTLEVELNODE_FROM_DB
18)    ex_feature1 = ex_nodes.feature1
19)    inex_feature1 = feature1_to_addresses.keys().difference(ex_feature1)
20)    for first_level_node in ex_nodes:
21)      //一级节点已存在, 调用动态节点管理函数
22)      addresses = feature1_to_addresses[first_level_node.feature1]
23)      NODE_MANAGE(addresses, first_level_node)
24)      first_level_nodes.expand(ex_nodes)
25)      for feature1 in inex_feature1:
26)        //一级节点不存在, 调用构建一级节点函数
27)        addresses = feature1_to_addresses[feature1]
28)        for address in addresses:
29)          node = BUILD_NODE(addresses, feature1)
30)          first_level_nodes.expand(node)
31)      return first_level_nodes

```

(3) 动态节点管理

随着新增地址数量的增加, 原有二级节点的分裂点将不再适用, 需对这些节点适时重构。新增地址完成一级分类后, 算法检查对应的一级空间节点是否存在。若节点已存在, 且新增地址数超过该节点原有活跃地址数的 40%, 则依据节点内所有地址的后缀取值量化指标重新计算分裂点, 并重新生成二级节点。以 40% 作为阈值, 可在及时捕捉地址空间变化的同时, 避免因频繁重构而造成资源浪费。

2.2 分段平衡空间建模

空间建模旨在识别并适度扩展地址空间的可变维度, 从而生成目标地址。Multi-BSPR-Gen 提出的平衡空间模式表示 (balanced space pattern representation, BSPR) 通过分析 IPv6 地址集

合在每个十六进制位上的取值分布，生成紧凑的模式表示。BSPR虽在一定程度上缩小了建模空间，却忽略了IPv6各语义区域的不同建模需求。相较IID区域，网络前缀变化更为单一，宜采用保守策略，以避免建模空间膨胀。为此，本文提出分段平衡空间模式表示（segmented balanced space pattern representation, S-BSPR），由多策略模式生成与分段策略选择两部分组成。

（1）多策略模式生成

S-BSPR沿用BSPR的4种表示策略，具体含义与取值范围如下。

- 1) Single: 固定值，该半字节无变化。
- 2) List: 变化值，取值仅限于地址集合在该位上实际出现的所有不同值。
- 3) Range: 变化值，取值范围为闭区间 $[\min, \max]$ ，部分中间值可能未出现在原始集合。
- 4) Wildcard: 变化值，取值范围为十六进制全集 $\{0 \sim F\}$ ，部分中间值可能未出现。

为确定每个十六进制位的最优表示方式，算法计算3位统计量：极差、香农熵和取值种类数。第 i 个半字节的香农熵定义为：

$$\text{Shannon}(x_i) = -\sum_{v=0}^f p(x_i = v) \log_{16} p(x_i = v) \quad (4)$$

其中，底数16确保熵值归一化至 $[0, 1]$ ， $p(x_i = v)$ 为地址集合在第 i 位取值为 v 的相对频率。算法将每位极差、熵值和取值种类数分别与预设阈值 range_t 、 entropy_t 、 type_t 比较，综合决策最终表示策略。单个维度策略与极差、熵值、取值种类数的关系见表2。

表2 单个维度策略与极差、熵值、取值种类数的关系

极差	熵值	取值种类数	表示策略	表示示例	建模数
0	0	1	Single	{a}	1
较大	较大	较多	Wildcard	{1, 2, 3, ..., d, e, f}	16
较大	较大	较少	List	{3, 8, a, d}	4
较大	较小	较多	Range	{3, 4, 5, 6, 7, 8, 9, a, b}	9
较大	较小	较少	List	{2, 3, e}	3
较小	较大	较多	Range	{5, 6, 7, 8, 9}	5
较大	较大	较少	List	{6, 8, 9}	3
较小	较小	较多/较少	List	{6, 7}	2

（2）分段策略选择

令 l 、 r 、 w 分别表示模式表示中采用List、Range、Wildcard策略的十六进制位个数；用 L_i 代表在第 i 个List位的取值个数（ $i=1, \dots, l$ ）；用 R_j 代表在第 j 个Range位的取值个数（ $j=1, \dots, r$ ），则模式表示的空间范围（space range, SR）由式(5)给出：

$$\text{SR} = 16^w \cdot \prod_{i=1}^l L_i \prod_{j=1}^r R_j \quad (5)$$

与BSPR类似，S-BSPR也需确定极差阈值 range_t 、熵阈值 entropy_t 和取值种类数阈值 type_t 。若阈值过小，建模策略会退化为全Wildcard（极端情况下与6Tree、DET算法的饱和建模一致），导致空间急剧膨胀并降低命中率；若阈值过大，则退化为纯List，使空间过度收缩，放大样本偏差。

式(5)中的 l 、 r 、 w 由 type_t 、 range_t 、 entropy_t 3个超参共同决定，因此可将SR视为关于这三者的三元函数： $\text{type_t} \in \{1, 2, \dots, 16\}$ ， $\text{range_t} \in \{0, 1, \dots, 15\}$ ， $\text{entropy_t} \in [0, 1]$ 。本文以0.05为步长将 $[0, 1]$ 离散化，得到21个离散点，作为 entropy_t 的有效定义域。

设SR在式(5)定义域的值域为集合 Y ，则其定义域由有限个离散值构成，所以集合 Y 同样为有限集。设SR的 Y 值域有 N 个取值，选择 Y 值域的几何平均值作为S-BSPR最终的BalancedSR。几何平均值对极端值不敏感，适合处理近似等比关系的数据。最终，选择使SR最接近BalancedSR的超参组合 $(\text{range}_t, \text{entropy}_t, \text{type}_t)$ 作为最优阈值，并按表2生成对应的最优模式表示。计算过程见式(6)和式(7)。

$$\text{BalancedSR} = \sqrt[N]{\prod_{\text{SR}_i \in Y} \text{SR}_i} \quad (6)$$

$$\text{argmin}_{\text{range}_t, \text{entropy}_t, \text{type}_t} |\text{SR} - \text{BalancedSR}| \quad (7)$$

本文在计算超参阈值时，将IPv6地址的前64 bit（前缀）与后64 bit（后缀）分开处理：分别计算不同超参组合下前缀和后缀的空间范围，再利用式(7)各自寻找最接近BalancedSR-P与BalancedSR-S的最优超参。后续建模过程中，前缀与后缀各自采用对应的最优阈值确定表示策略，从而区分IPv6地址的不同语义段，减少彼此干扰。

2.3 探测资源分配和目标地址生成

本方案的资源分配方法具有 4 个核心特点。

1) 以一级空间节点为最小分配单元, 以二级空间节点数量为实际资源计量单位。每轮探测按优先级为一级节点分配 n 个二级节点, 实施全量探测。由于二级节点队列在生成时已按后缀取值量化指标 LOG_Q 降序排列, LOG_Q 越高表明后缀取值越均匀, 越易被模型准确刻画。完成全量探测后, 该二级节点的模式表示即可视为已覆盖的地址空间模板; 在后续轮次中, 通过与新节点模式表示的笛卡儿积差集运算, 可直接推导出尚未探测的地址集合, 从而避免重复扫描, 降低资源消耗。

2) 多因素优先级计算。对于已探测节点, 可直接用命中率作为优先级; 对于未探测节点, 种子地址数量受初始分布不均影响, 难以单独反映发现活跃地址的潜力。为此, 本文引入地址膨胀比, 定义为种子地址数 (SC) 与建模空间大小 (SR) 的比值, 膨胀比公式如下:

$$\text{ExpandRate} = \frac{\text{SC}}{\text{SR}} \quad (8)$$

比值越高, 表明该空间种子分布越密集、特征越相似, 越容易发现新的活跃地址。

实验结果表明, IID 类型与发现活跃地址的概率密切相关。因此, 对未探测节点计算优先级时, 本文为不同 IID 类型分配系数 γ_I , 最终优先级如式(9)所示:

$$\text{priority} = \gamma_I \cdot \text{ExpandRate} \quad (9)$$

3) 梯度资源分配。每轮按优先级选取若干一

级节点, 随后依据其在队列中的次序梯度分配探测预算, 重点投入活跃地址密集区域。

4) 并行处理待测一级节点。完成节点选取与预算分配后, 各一级节点的探测相互独立, 采用高并发技术并行执行, 提升整体探测效率。

3 IPv6 别名前缀主动探测

IPv6 网络前缀和 IPv6 地址在结构上十分相似, 受基于种子地址的 IPv6 地址探测方案启发, 本文提出基于种子前缀的别名前缀主动探测方案。通过提取已知别名前缀的特征进行分析和建模, 生成高概率的候选前缀, 从而缩小探测范围, 高效主动探测未知别名前缀。为此, 本文设计了前缀自适应 IPv6 探测适配器 PAIPA, 利用别名前缀格式化和目标前缀别名验证机制, 使现有基于种子地址的 IPv6 地址探测方案可直接应用于别名前缀的探测。基于种子的 IPv6 别名前缀主动探测与基于种子的 IPv6 活跃地址探测对比如图 3 所示。

3.1 种子前缀格式化方案

种子前缀格式化用于适配 IPv6 探测算法所需的 32 位十六进制格式, 并确保格式化前后字符串一一对应。具体方案如下。

1) 非 4 倍数长度前缀: 将其替换为若干个长度是 4 的倍数的最大子网, 以便通过字符串长度直接解析前缀长度。

2) 一对一映射: 低位填 0 无法区分“填充 0”与“原有 0”, 本文采用“前缀展开+高位填 0”方案; 程序可去高位 0 后唯一还原目标前缀。

3) 网络前缀分类: 不同长度前缀混用会互相

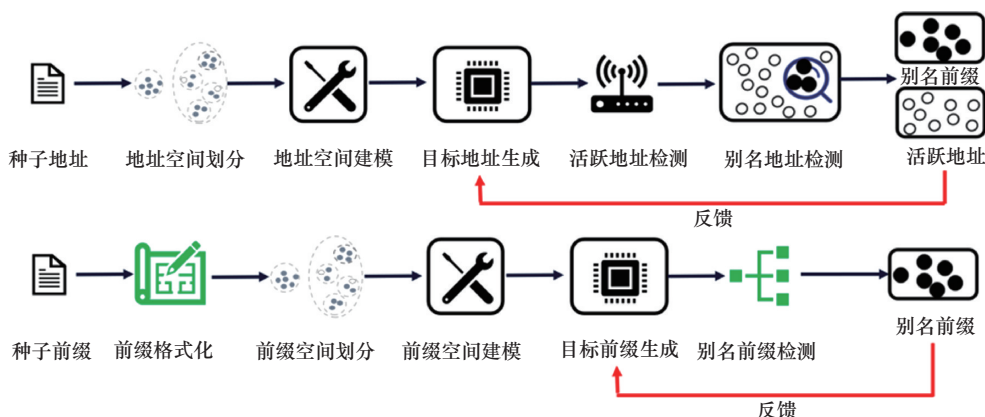


图 3 基于种子的 IPv6 别名前缀主动探测与基于种子的 IPv6 活跃地址探测对比

干扰，处理前须按前缀长度分类，避免语义混淆。

3.2 目标前缀验证方法

本文采用Gasser等^[19]提出的分层随机地址验证法：在目标前缀的多个子网下各随机生成1个IPv6地址，发送ICMPv6报文，若响应包数量与发送包数量相等，则判定该前缀为别名前缀。该方案基于经验事实：IPv6地址空间极度稀疏，同一子网内16个随机地址全部活跃的概率可忽略不计。

4 实验分析

4.1 数据集介绍

为验证方法的泛化能力，本节选用多个具有代表性的数据集进行测试。

1) Gasser IPv6 Hitlist：由Gasser等^[19]通过全球分布式扫描获得，含9 203 800个活跃IPv6地址，覆盖58 200个BGP前缀，在网络拓扑表征上具有广泛性与代表性。

2) CAIDA DNS-names & ZDNS：基于DNS流量分析生成，分别含248.74万个与68.14万个活跃地址，侧重基础设施关联地址，可与Gasser数据集在地域与功能上形成互补。

3) 别名前缀数据集：本文对Gasser IPv6 Hitlist深度挖掘，提取18 001 000个候选别名前缀，经前缀树去重后构建而成。

4) BGP前缀数据集：整合Route Views项目数据，含227 700个BGP前缀及全球AS级前缀分布信息。

本节实验基于多源异构的公开数据集，构建涵盖IPv6活跃地址、IPv6别名前缀及IPv6 BGP网络前缀的综合性测试数据集，确保实验场景的真实性以及实验结果的普适性，实验数据类型及数据源见表3。

表3 实验数据类型及数据源

数据类型	数据源	数量/个
IPv6 活跃地址	Gasser IPv6 Hitlist	920.38 万
IPv6 活跃地址	CAIDA DNS-names	248.74 万
IPv6 活跃地址	CAIDA ZDNS	68.14 万
IPv6 别名前缀	Gasser IPv6 Hitlist	180.01 万
IPv6 BGP 前缀	Route Views	22.77 万

4.2 IPv6 活跃地址动态探测实验分析

4.2.1 实验设计

为系统评估FFDIP的实际性能，本节在无存量与有存量两种场景下分别进行实验，并与DET、6Tree、Entropy/IP 3种主流算法对比。为保证结果准确可靠，共设置4组严格控制变量的子实验，全面考察不同条件下的真实表现。实验以命中率为核心指标（探测到的活跃地址数/总生成地址数），并同步记录运行耗时，具体实验内容如下。

(1) 数据分组

先对多源数据集进行活跃地址的筛选与合并，再通过随机抽样构建含100万个IPv6活跃地址的地址集 C ，随后将 C 分为数据集 s 和 Δs ，实验数据集说明见表4。

表4 实验数据集说明

数据集	数量/个	说明
s	80 万	作为第一次地址探测的种子地址集，并在第二次探测中作为FFDIP的存量地址
Δs	20 万	作为第二次地址探测的增量地址

(2) 实验1：无存量地址场景

对数据集 s 使用4种算法分别生成8万、16万、24万、32万、40万个新地址（待测/种子=1/10、2/10、3/10、4/10、5/10），模拟无存量场景下的首次全量探测。

(3) 实验2：有存量地址增量探测

对数据集 Δs 分别生成2万、4万、6万、8万、10万个新地址（待测/种子=1/10、2/10、3/10、4/10、5/10）。DET、6Tree、Entropy/IP算法仅针对增量地址进行探测，模拟有存量时的增量发现。

(4) 实验3：有存量地址全量探测

DET、6Tree、Entropy/IP算法对完整数据集 C （100万个）分别生成10万、20万、30万、40万、50万个新地址（待测/种子=1/10、2/10、3/10、4/10、5/10），每次重新聚类并全量探测，模拟存量与增量混合场景。

(5) 实验4：大规模地址生成场景

用4种算法对完整数据集 C 进行深度全量探

测, 分别生成 100 万、300 万、500 万、1 000 万个新地址 (待测/种子=1/1、3/1、5/1、10/1), 待测/种子显著高于前 3 组实验, 用于全面评估算法在大规模地址生成条件下的性能。

(6) 算法命中率与耗时分析

系统对比 4 种算法在 4 类实验中的命中率, 评估其发现活跃地址的能力, 并在时间复杂度分析的基础上, 从运行耗时维度对 4 种算法进行全场景性能评价。

4.2.2 无存量地址场景探测

实验 1 针对无存量地址场景, 4 种算法对 80 万个地址初始全量探测的命中率对比如图 4 所示。即便缺乏历史数据, FFDIP 算法命中率最高仍达 44.98%, 显著优于 DET、6Tree 与 Entropy/IP 算法, 验证了其在初始部署条件下的有效性。随着目标地址生成量增大, 4 种算法命中率均下降: 高价值区域被优先耗尽后, 资源被迫转向低价值区域, 探测效率自然降低。

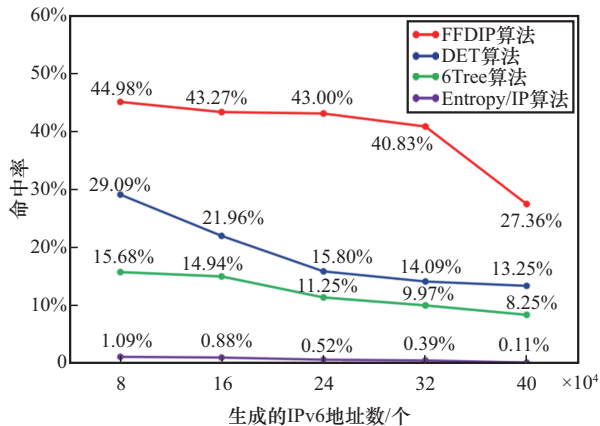


图4 4种算法对80万个地址初始全量探测的命中率对比

4.2.3 有存量地址场景增量探测

实验 2 针对存量地址场景, 4 种算法对 20 万个增量地址的探测命中率对比如图 5 所示。面对规模有限且分布稀疏的新增活跃地址, 传统算法可用的种子地址更少, 特征提取困难, 建模精度下降, 命中率随之降低。相比之下, FFDIP 算法凭借特征融合机制同时利用增量与存量地址特征进行预算分配和空间建模, 仍维持 40% 以上的高命中率。

值得注意的是, 当待测地址与种子地址比

例为 2/10~5/10 时, FFDIP 算法命中率较实验 1 最高可提升 50.04%。存量地址的先验信息引导算法将资源集中于高价值区域, 显著优化了空间配置与探测效率。该实验充分验证了 FFDIP 算法在有存量地址场景下处理增量探测的高效性。

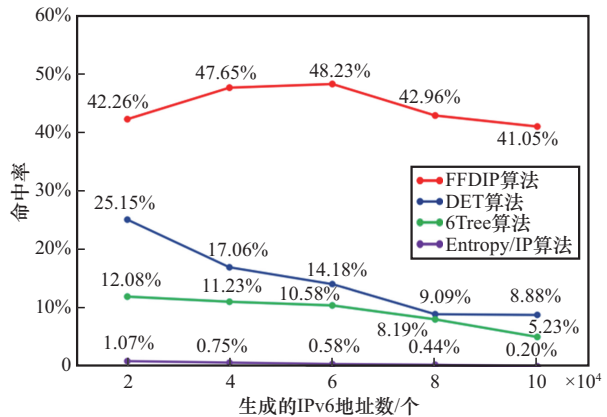


图5 4种算法对20万个增量地址的探测命中率对比

4.2.4 有存量地址场景全量探测

实验 3 针对存量和增量地址场景, 4 种算法对 100 万个地址进行全量探测命中率对比如图 6 所示。为全面评估算法性能, 需考虑以下两点。

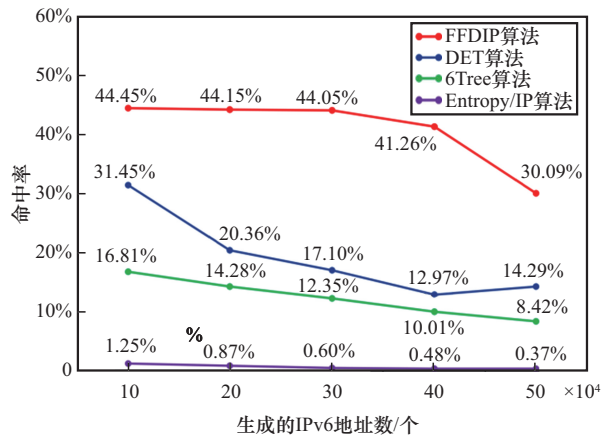


图6 4种算法对100万个地址进行全量探测命中率对比

1) 训练集规模公平性: FFDIP 算法在特征提取阶段使用 100 万个种子地址, 而实验 2 中对比算法仅基于 20 万个地址运行; FFDIP 算法通过持续并入新增活跃地址实现模型迭代, 相当于动态扩充了训练集。若对比算法仍沿用原有小规模种子, 将因训练集规模与特征覆盖度不足而处于劣

势。为此，设计实验3以消除种子数量差异，确保对比公平。

2) 输入模式真实性：实验2采用纯增量输入，而实际研究多采用“增量+存量”合并的全量探测策略。实验3增设全量场景，使评估更贴近真实应用。

结果显示，即使所有算法均以相同的100万个地址全集作为输入，FFDIP算法仍凭借特征融合技术显著优于其他3种传统算法，命中率最高达44.45%，再次验证其优越性。

4.2.5 大规模地址生成全量探测

实验1~3的新地址生成量相对有限，最大仅为种子地址数的50%。为全面检验算法在大规模生成场景下的性能，实验4显著扩大地址生成规模并实施全量探测，结果如图7所示。即便在千万级规模下，FFDIP依旧表现优异：生成1 000万个新地址时，其命中率较次优算法提升21.86%，展现出良好的可扩展性与高效性，充分验证了FFDIP算法在大规模地址生成场景中的稳定性能与技术优势。

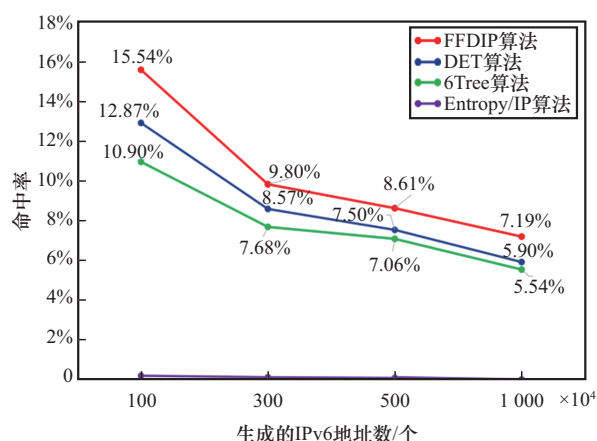


图7 4种算法对100万个地址进行探测生成大规模地址命中率对比

4.2.6 算法时间复杂度分析

从算法复杂度理论视角分析，Entropy/IP算法通过对IPv6地址32个半字节位置进行统计分析并熵计算，其时间复杂度呈典型的线性阶 $O(n)$ 。6Tree算法虽在聚类阶段具备线性时间复杂度 $O(n)$ ，但因空间树生成过程中引入对全体地址向量排序的操作，该步骤时间复杂度为 $O(n \log n)$ ，导致6Tree整体时间复杂度为 $O(n \log n + n)$ ，可

简化为 $O(n \log n)$ 。相比之下，DET算法避免了地址向量排序开销，与Entropy/IP算法类似，可直接对每个半字节进行熵计算，其时间复杂度同样维持线性阶 $O(n)$ 。

本文提出的FFDIP算法，其核心操作模块在时间复杂度层面均呈线性特性：动态地址空间划分算法遍历所有地址并提取特征，时间复杂度为 $O(n)$ ；分段平衡空间建模针对32个维度进行特征计算（包括半字节香农熵、极差及取值种类数量化分析），虽涉及维度遍历，但经简化后仍可归约为 $O(n)$ 量级；在探测资源分配与目标优先级计算过程中，地址膨胀比与不同IID类型优先级系数的计算同样维持在 $O(n)$ 复杂度水平。FFDIP算法通过避免全局排序，并利用二级分类和并行处理，相较于DET算法，在动态环境下能呈现出更优的时间复杂度性能表现。

4.2.7 算法耗时对比

为更全面地评估上述4种算法的性能，本文对其运行耗时进行统计分析，实验结果见表5。从中可以看出，FFDIP算法在运行耗时方面展现出显著优势：相较DET算法，其运行耗时降低幅度达63.9%~82.5%；与6Tree和Entropy/IP算法相比，耗时减少量更是超过90%。这主要归因于FFDIP算法设计方面的优化：其一，算法除了对一级空间节点进行排序和对待测地址探测外，其余处理步骤均以一级空间节点为单位进行，并通过多线程技术实现不同一级空间节点的并行处理，效率更高；其二，FFDIP采用基于单个地址编址特征的分类策略，避免对整个数据集的大规模统计与计算，从而进一步有效降低算法执行开销。此外，对于同样基于单个地址编址特征的Multi-BSPR-Gen算法，相较于DET和6Tree算法，运行耗时也分别减少37.68%和74.01%。然而，由于该算法代码未公开，本文在此未进行具体的对比实验。

综合上述3组实验的命中率与运行耗时统计及对比分析，可充分验证：FFDIP算法在各类探测场景下均展现出显著的性能优势与高效特征。

表5 4种算法处理不同数量种子地址运行耗时

算法	时间复杂度	种子地址数/个	运行耗时/s
FFDIP	$O(n)$	20 万	58.56
		100 万	148.77
		500 万	484.18
		1 000 万	873.01
DET	$O(n)$	20 万	162.66
		100 万	961.59
		500 万	3 070.38
		1 000 万	4 989.67
6Tree	$O(n \log n)$	20 万	351.73
		100 万	2 058.94
		500 万	13 194.18
		1 000 万	28 480.74
Entropy/IP	$O(n)$	20 万	287.33
		100 万	1 566.24
		500 万	8 125.29
		1 000 万	18 266.17

4.3 IPv6别名前缀主动探测实验分析

4.3.1 实验设计

本文将PAIPA嵌入DET与6Tree算法,分别构建别名前缀主动探测算法DET-P和6Tree-P算法;在去重后的别名前缀数据集上进行别名前缀探测实验,并与原DET、6Tree算法的活跃地址探测性能进行对比。

4.3.2 数据集去重处理

本文通过数据采集获取了规模达5 264万个的别名前缀数据集,随后提出一种基于前缀树的IPv6网络前缀去重算法进行数据处理。经严格去重后,最终筛选出1 052万个有效前缀,并对网络前缀进行格式化操作便于后续处理。统计分析显示,原始数据集冗余数据占比高达80.02%,这一结果不仅验证了去重算法的有效性,更充分揭示了在IPv6网络前缀管理中实施去重操作的必要性与重要意义。

4.3.3 别名前缀主动探测实验

将PAIPA应用于DET和6Tree算法进行别名前缀主动探测实验,与原DET、6Tree算法的活跃地址探测结果进行对比,其结果如图8所示。可以看出,相较于活跃地址探测场景,DET算法和6Tree算法在别名前缀探测场景下均展现出更高的命中率。这主要归因于别名前缀的字符长度较短,更容易发现特征;此外在地址分配机制层面,别名前缀分配策略相对简洁,避免了Random、IEEE等复杂地址分配规则带来的干扰。

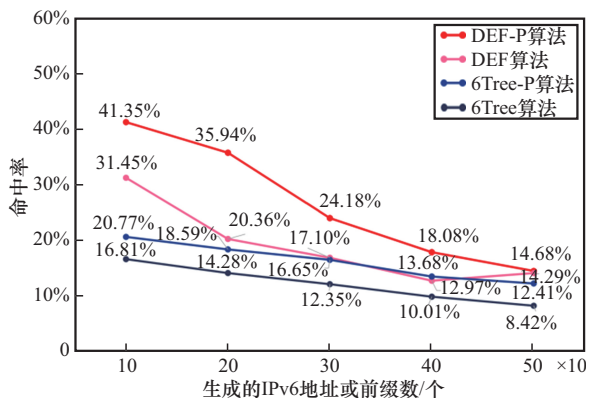


图8 DET、6Tree算法在别名前缀探测和活跃地址探测中的效果对比

以上实验结果不仅验证了PAIPA在IPv6地址探测算法中的技术可行性,同时也证实通过主动探测手段能够有效获取别名前缀信息,且该探测对象在IPv6地址空间中具有更高的可探测性。

5 结束语

本文深入分析了IPv6地址空间探测的问题及传统方法在动态网络环境中的不足,针对IPv6网络的动态性,提出FFDIP算法,通过特征融合算法将新增地址与存量地址结合实现动态探测,并利用平衡空间建模、多轮探测反馈等合理分配探测资源,提高探测效率。针对IPv6中的别名前缀现象,设计了PAIPA,通过对网络前缀格式化和别名前缀验证算法,使当前大多数基于种子地址的IPv6地址探测算法能够应用于IPv6别名前缀的探测,实现了IPv6别名前缀的主动探测,缓解了传统被动别名发现方案在实际应用

中的局限性。

实验结果表明,相较于DET、6Tree和Entropy/IP算法,在无存量地址场景和有存量地址场景中探测效率均有所提升,实验证明了FFDIP算法的高效性和动态探测的可行性。将PAIPA应用于DET和6Tree算法,其在别名前缀探测中的效果优于在活跃地址探测中的效果,验证了PAIPA的可行性和别名前缀的高可探测性。

参考文献:

- [1] 李果,何林,宋光磊,等.基于种子地址的IPv6地址探测技术综述[J].电信科学,2019,35(12):24-37.
LI G, HE L, SONG G L, et al. Survey on IPv6 address scanning technology based on seed sources[J]. Telecommunications Science, 2019, 35(12): 24-37.
- [2] ULLRICH J, KIESEBERG P, KROMBOLZ K, et al. On reconnaissance with IPv6: a pattern-based scanning approach[C]//Proceedings of the 2015 10th International Conference on Availability, Reliability and Security. Piscataway: IEEE Press, 2015: 186-192.
- [3] FOREMSKI P, PLONKA D, BERGER A. Entropy/IP: uncovering structure in IPv6 addresses[C]//Proceedings of the 2016 Internet Measurement Conference. New York: ACM, 2016: 167-181.
- [4] MURDOCK A, LI F, BRAMSEN P, et al. Target generation for Internet-wide IPv6 scanning[C]//Proceedings of the 2017 Internet Measurement Conference. New York: ACM, 2017: 242-253.
- [5] LIU Z Z, XIONG Y Q, LIU X, et al. 6Tree: Efficient dynamic discovery of active addresses in the IPv6 address space[J]. Computer Networks, 2019, 155: 31-46.
- [6] SONG G L, YANG J H, WANG Z L, et al. DET: enabling efficient probing of IPv6 active addresses[J]. IEEE/ACM Transactions on Networking, 2022, 30(4): 1629-1643.
- [7] 李果,何林,宋光磊,等.基于多层级分类和空间建模的IPv6活跃地址发现算法[J].清华大学学报(自然科学版),2021,61(10):1177-1185.
LI G, HE L, SONG G L, et al. IPv6 active address discovery algorithm based on multi-level classification and space modeling[J]. Journal of Tsinghua University (Science and Technology), 2021, 61(10): 1177-1185.
- [8] CARPENTER B, JIANG S. Significance of IPv6 Interface Identifiers: RFC 7136[S]. Fremont, CA: RFC Editor, 2014.
- [9] CUI T, GOU G, XIONG G. 6GCVAE: gated convolutional variational autoencoder for IPv6 target generation[C]//Proceedings of the 24th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2020), Singapore. Cham: Springer International Publishing, 2020: 609 - 622.
- [10] CUI T Y, GOU G P, XIONG G, et al. 6GAN: IPv6 multi-pattern target generation via generative adversarial nets with reinforcement learning[C]//Proceedings of the IEEE INFOCOM 2021 - IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2021: 1-10.
- [11] HOU B N, CAI Z P, WU K, et al. 6Hit: a reinforcement learning-based approach to target generation for Internet-wide IPv6 scanning[C]//Proceedings of the IEEE INFOCOM 2021 - IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2021: 1-10.
- [12] YANG T, CAI Z P, HOU B N, et al. 6Forest: an ensemble learning-based approach to target generation for Internet-wide IPv6 scanning[C]//Proceedings of the IEEE INFOCOM 2022 - IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2022: 1679-1688.
- [13] YANG T, HOU B N, CAI Z P, et al. 6Graph: a graph-theoretic approach to address pattern mining for Internet-wide IPv6 scanning[J]. Computer Networks, 2022, 203: 108666.
- [14] LIU Q K, LI X. 6Former: transformer-based IPv6 address generation[C]//Proceedings of the 2023 IEEE Symposium on Computers and Communications (ISCC). Piscataway: IEEE Press, 2023: 1142-1148.
- [15] SONG G L, HE L, ZHU F Y, et al. AddrMiner: a fast, efficient, and comprehensive global active IPv6 address detection system[J]. IEEE/ACM Transactions on Networking, 2024, 32(5): 3870-3887.
- [16] 宋光磊,张文健,林金磊,等.6EDL: 高效的大规模活跃IPv6地址探测系统[J].计算机学报,2024,47(8):1949-1969.
SONG G L, ZHANG W J, LIN J L, et al. 6EDL: efficient large-scale active IPv6 address probing system[J]. Chinese Journal of Computers, 2024, 47(8): 1949-1969.
- [17] THOMSON S, NARTEN T, JINMEI T. IPv6 stateless address auto-configuration: RFC 4862[S]. Fremont, CA: RFC Editor, 2007.
- [18] ALT L, BEVERLY R, DAINOTTI A. Uncovering network tar pits with degreaser[C]//Proceedings of the 30th Annual Computer Security Applications Conference. New York: ACM, 2014: 156-165.
- [19] GASSER O, SCHEITL Q, FOREMSKI P, et al. Clusters in the expanse: understanding and unbiasing IPv6 hitlists[C]//Proceedings of the Internet Measurement Conference 2018. New York: ACM, 2018: 364-378.

[作者简介]



王泽昊(2000-),男,河南郑州人,华中科技大学硕士生,主要研究方向为网络安全、网络探测。



李冬(1979-),男,博士,湖北枣阳人,华中科技大学高级工程师,主要研究方向为计算机网络、软件定义网络、网络安全。



于俊清 (1975-), 男, 博士, 内蒙古赤峰人, 华中科技大学教授、博士生导师, 主要研究方向为数字媒体处理与检索、网络安全、多核计算与流编译。



刘陶然 (2000-), 男, 湖北荆门人, 华中科技大学硕士生, 主要研究方向为网络恶意流量分类、IPv6 地址生成。



陈俊鑫 (2002-), 男, 浙江嘉兴人, 华中科技大学硕士生, 主要研究方向为网络安全、软件定义网络、网络探测。



税帅策 (2001-), 男, 湖北恩施人, 华中科技大学硕士生, 主要研究方向为网络安全、可编程网络。