# UAv6: Alias Resolution in IPv6 Using Unused Addresses

Ramakrishna Padmanabhan, Zhihao Li*, Dave Levin, and Neil Spring

University of Maryland, College Park, MD
{ramapad,zhihaoli,dml,nspring}@cs.umd.edu

As the IPv6 Internet grows, alias resolution in IPv6 becomes more important. Traditional IPv4 alias resolution techniques such as Ally do not work for IPv6 because of protocol differences. Recent techniques adopted specifically for IPv6 have shown promise, but utilize source routing, which has since been deprecated, or rely upon sequential fragment identifiers supported on only a third of router interfaces. As a result, IPv6 alias resolution remains an open problem.

This paper introduces *UAv6*, a new alias resolution technique for IPv6. UAv6 finds aliases in two phases. The first "harvest" phase gathers potential alias pairs, and is based on our empirical observation that addresses adjacent to router interface addresses are often unused. UAv6 probes these unused addresses, eliciting ICMPv6 Address Unreachable responses. The central assumption of this work is that the source address of such a response belongs to a router directly connected to the prefix containing the unused and router interface addresses. The second "disambiguation" phase determines which interface address is an alias of the Address Unreachable's source address. UAv6 uses both new and established techniques to construct proofs or disproofs that two addresses are aliases.

We confirm the accuracy of UAv6 by running the Too-Big Trick test upon the aliases we find, and by comparing them with limited ground truth. We also show that the classic address-based technique to resolve aliases in IPv4 works for IPv6 as well, and show that the address-based technique, UAv6, and the Too-Big Trick are complementary techniques in resolving IPv6 aliases.

## 1 Introduction

With the impending exhaustion of IPv4 addresses, IPv6 adoption has seen steady growth [8], and particularly robust growth in the last two years [7]. As IPv6 deployment increases, knowledge of its topology becomes valuable to researchers and commercial providers. Traceroutes are the traditional tool for inferring network topology [5, 18], but using traceroutes alone for topology-mapping does not suffice. Traceroutes discover multiple interfaces of a router, but do not reveal which interfaces belong to the same router. *Alias resolution* is the process of grouping interfaces onto their corresponding routers, thereby rendering a more accurate picture of the actual network topology.

Numerous alias resolution techniques exist for IPv4 [2, 17, 18], but protocol differences prevent their straightforward application to IPv6. Researchers have

* The first two authors contributed equally to this work.

come up with several IPv6-specific techniques over the last decade. Early techniques used the source routing feature in IPv6 to resolve aliases [14, 15, 19], but source routing in IPv6 has since been deprecated [1]. Another successful approach to resolve aliases is the shared counter method: Ally [18] and Radargun [2] use this technique in IPv4, and recently, the Too-Big Trick (TBT) applied this approach to find aliases in IPv6 [3, 12]. However, Speedtrap [12] reports that 68% of router interfaces do not respond to the Too-Big Trick. Thus, alias resolution in IPv6 remains an open problem.

In this paper, we describe a new alias resolution technique, *UAv6*, which operates in two phases. The first phase, called the *harvest phase*, collects candidate aliases by probing unused addresses in IPv6 router interface prefixes. The IPv6 address space is large enough that addresses for point-to point links are not typically assigned out of 127-bit prefixes which have only two addresses; rather, point-to-point links typically use only two of the four addresses in a /126 prefix. By sending a packet to an address that is within a prefix but not assigned to an interface, we solicit an ICMPv6 Address Unreachable (AU) error. Only a router directly connected to the prefix is likely to respond with an AU. Therefore, the source address of the AU is an alias for one of the used addresses within the prefix. This results in two possible alias pairs, but the harvest phase does not determine which of them is the true alias.

UAv6's second phase, called the *disambiguation phase* determines which of the harvest's candidate aliases are true aliases. Because one of the two candidate aliases produced by the harvest phase must be a true alias, we can either prove one of them to be true, or we can disprove one and conclude the other must be true by process of elimination. We provide tests of both types and show that they are complementary. The first test uses traceroutes to *disprove* one of the candidate aliases: If one of the addresses in the pair appears on the path to the other, they are unlikely to be aliases of one another. The second test uses shared Path MTU (PMTU) caches in some router implementations to *prove* one of the alias pairs true: If an address pair shares PMTU caches, it is a true alias pair, as only aliases share PMTU caches.

The contributions of this work are :

- We observe the presence of unused addresses in router interface address prefixes. We present UAv6, a two-phase alias resolution technique in IPv6 that uses these partially used prefixes.
- We verify UAv6's accuracy by running the TBT test [3] where possible. TBT could be applied to 23.2% of the alias pairs we found and it confirmed 99.86% of them. We also compare the aliases we find against limited ground truth from the Internet2 dataset and verify all the Internet2 aliases we discover.
- We demonstrate that a classic IPv4 alias resolution technique, the address-based technique [9,13,18], works in IPv6, in spite of recommendations in RFC 4443 [6]. We show, however, that UAv6 finds almost twice as many aliases as the address-based technique within router interface addresses derived from traceroutes sent by the Ark project [4].

## 2 Related Work

Alias resolution schemes can be broadly classified into the following categories:

**Address-based**: In IPv4, some routers are configured to use the outgoing interface's address as the source address for certain ICMP response types. Pansiot and Grad [13] harness this to obtain aliases by checking when the source address in a response is different from the destination probed. Some researchers [12, 19] have been discouraged from applying a similar approach in IPv6, because the ICMPv6 specification [6] states that IPv6 routers must use the address to which the packet was sent as the source address in ICMPv6 responses, if the address belongs to the router. We demonstrate in Section 5 that, contrary to the specification, the address-based approach finds aliases in IPv6.

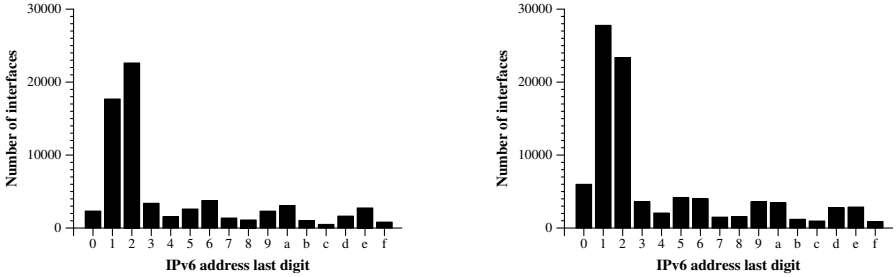
**Source routing-based**: In the early 2000s, only 8% of IPv4 routers supported source routing [9], but the IPv6 Internet supported the feature in most routers [19]. Early IPv6 alias resolution techniques used source routing-based methods to find aliases [14, 15, 19]. However, source routing in IPv6 has been deprecated because of security concerns [1] and support is likely to decline further.

**Shared counter-based**: In IPv4, Rocketfuel [18] introduced Ally, an alias resolution scheme that determines aliases by checking if the "IP-ID" fields on two interfaces are generated from a shared counter. IPv6 dispensed with the IP-ID field because routers do not fragment packets in IPv6 when forwarding. Instead, if an interface obtains a too-large packet, it sends an ICMP Packet Too Big (PTB) message to the source. The source then sends subsequent too-large packets as fragments and inserts a common Fragment ID into fragments for reassembly.

The "Too-Big Trick" (TBT) technique introduced by Beverly et al. [3] found that many IPv6 routers use a counter that is shared among all of its interfaces, from which these fragment IDs are obtained. To solicit fragmented packets, TBT sends a large Echo Request packet (1300 bytes) to both addresses in a candidate alias pair, followed by a PTB message to each of them. Next, it sends large Echo Requests alternately to each address. If the returned fragments have sequential fragment IDs, then TBT declares the pair to be aliases.

Given a set of router interface addresses obtained from traceroutes, TBT requires a number of probes proportional to the number of pairs of addresses, since TBT is a pairwise test. Speedtrap [12] obtains the same aliases that TBT would have obtained, but does so more efficiently. It probes interface addresses in parallel and groups together candidate alias pairs into smaller sets before performing TBT's pairwise test upon members of the set. However, only 32% of router interfaces in the IPv6 Internet provide fragments from a shared sequential counter [12].

**Prefix-based**: UAv6 does not depend upon shared sequential counters, support for source routing, or on ICMPv6 responses from different source addresses. Instead, it relies upon the presence of prefixes that contain unused addresses adjacent to router interface addresses. The next section shows that such partially used prefixes are common in IPv6.




(a) Addresses from CAIDA traceroutes (b) Addresses that responded to probes

Fig. 1: Distribution of the final hex digit of router interfaces' IPv6 addresses.

## 3 Unused Addresses in IPv6 Prefixes

Since the IPv6 address space is immense, we expect that IPv6 router interface addresses on point-to-point links are assigned out of /126 prefixes, or larger, leaving some addresses unused. This is similar to the existing practice of using /30s in IPv4 [17]. However, two conflicting RFCs for IP address assignment in IPv6 create uncertainty. RFC 3627 [16], published in 2003, finds that /127 prefix lengths in IPv6 are harmful and recommends the use of /64 prefixes instead for point-to-point links. RFC 6164 [11], published in 2011, recommends the use of /127s for point-to-point links.

We investigate if IPv6 router interface addresses are allocated from /126 or larger prefixes by studying the distribution of their last digits. We extracted 68,474 router interface addresses from traceroutes sent by the Ark project in July 2014 [4]. Figure 1(a) shows the distribution of router interface addresses across the last hex digits for these addresses. Most (59%) addresses end in hex digits "1" or "2". Further, 82% end in the binary digits "01" or "10".

We believe that this distribution is a result of ISPs assigning addresses out of /126s, or larger, to point-to-point links. In such networks, one end of the point-to-point link is assigned an address ending with the binary suffix "01" and the other end is assigned an address with the binary suffix "10". The other addresses in the /126 prefix, with suffixes "00" and "11", are unused, or assigned as broadcast addresses.

CAIDA's traceroutes may have recovered addresses in only one direction of a path, if the path had not been probed in the reverse direction. To address this potential bias, we send ICMPv6 Echo Request probes to the rest of the addresses in each address' enclosing /126. In total, we sent probes to 227,212 addresses and received ICMPv6 Echo Replies from 89,756 (39.5%) of them. We plot the frequency of the last hex digit for these responsive addresses in Figure 1(b). Unlike Figure 1(a), we find that the peak for addresses ending in "1" is higher than "2" and the peak for "0" is higher than the other last digits. We speculate

that this may be due to some ISPs using hexadecimal"1"s and "0"s on opposite ends of a link.

The peaks for {"5", "6"}, {"9", "a"} and {"d", "e"} are of comparable heights, suggesting that that these addresses are used for end-points of a link. Overall, we find that 80.3% of addresses that responded to our probes with ICMPv6 Echo Replies end in binary suffixes "01" or "10". This supports our belief that IPv6 point-to-point link prefixes are /126s or larger. Only the two addresses assigned to opposite ends of a link are in use and the remaining addresses in the prefix are unused.

## 4 UAv6 Design

In this section, we describe how UAv6 resolves aliases by using unused addresses. UAv6 consists of two phases, the *harvest* phase and the *disambiguation* phase. In the harvest phase, we obtain Address Unreachable responses from unused addresses and obtain potential alias pairs. In the disambiguation phase, we use established and new methods to prove which potential pairs are truly aliases.

### 4.1 The Harvest Phase

In the harvest phase, we probe /126 prefixes and obtain potential aliases from the responses. Given a /126 prefix, the harvest phase first determines if we can collect candidate alias pairs from this prefix by sending ICMPv6 Echo Requests to each of the addresses and inspecting the responses. If all addresses in the prefix are used, then all ICMPv6 Echo Replies we receive are, according to the specification [6], supposed to originate from the address we probed, thereby providing no information about aliases. Likewise, we learn no new aliases if none of the addresses in the prefix are used, as we will receive either ICMPv6 Address Unreachable (AU) responses or no responses at all. However, if some addresses in the prefix are used and some are not, then we receive ICMPv6 Echo Replies from the used addresses and AU responses from potential aliases of the used addresses. The harvest phase uses this combination of responses to obtain candidate alias pairs.

Figure 2 shows an example of how the harvest phase works. In this example, there are two routers connected by a point-to-point link; one of the end-points has address X1[1] and the other has X2. The harvest phase sends probes to each address in the /126 prefix "X" viz. X0, X1, X2 and X3. Because X1 and X2 are in use, they will respond with ICMPv6 Echo Replies. As for the unused addresses X0 and X3, we assume that the AU response is sent by an interface (Y) that belongs to one of the routers that is directly connected to the X prefix. We make this assumption because in general, only the routers directly connected to prefix X know that X0 and X3 are unused. Since X1 and X2 are the addresses from this prefix that responded with ICMPv6 Echo Replies, we infer that Y is an alias of

[1] We use XN as notational shortcut for X::N.

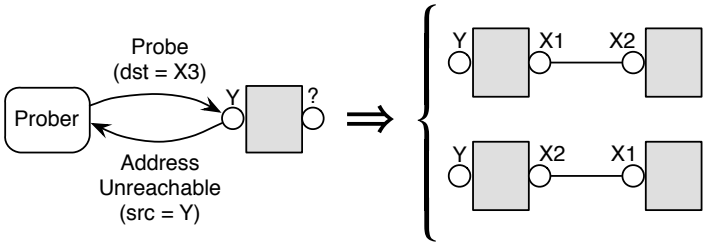



Fig. 2: In its harvest phase, UAv6 sends probes to each address in a given /126 beginning with the prefix "X". A probe for X3, which is likely unused, will probably elicit an ICMPv6 Address Unreachable (AU) message—we assume that this message will be sent from a router that has an interface from the X prefix. In this example, interface Y responded to our probe for X3 with an AU message, so we can deduce that Y is likely an alias for X1 or X2, but not both. The disambiguation phase determines which is the true alias.

X1 or X2. We define (Y, X1) and (Y, X2) to be the two members of a candidate alias pair set, exactly one of which is a true alias pair. For each /126 or larger prefix with used and unused addresses, we obtain one candidate alias pair set at the end of the harvest phase.

### 4.2 The Disambiguation Phase

In the disambiguation phase, we find the correct alias pair in a candidate alias pair set provided by the harvest phase. We apply two tests which either prove that an alias pair is correct, or prove that one is *not* and thus the other must be. While some candidate alias pair sets can be disambiguated by either test, we show in Section 5 that these two tests are complementary, as they rely on different router behaviors.

#### *4.2.1 Traceroute Test*

We use traceroutes to obtain disproofs about candidate alias pairs by checking if one of the addresses lies on the route to the other. We expect that a typical IPv6 router first checks if the destination address in the packet belongs to it before decrementing the Hop Limit. An alias of a traceroute destination should thus never send an ICMPv6 Hop Limit Exceeded message, which implies that it should never appear on the route to the destination. We send ICMPv6 traceroutes to X1 and X2, and if Y appears on the route to one of them, we use that as proof that Y is *not* an alias of that address.

The Traceroute test cannot disambiguate all candidate alias pair sets. For instance, traceroute probes may be blocked by some ISPs. Alternately, traceroutes to X1 and X2 may both not find Y on the route if the traceroutes traverse different paths. Therefore, we introduce a complementary technique, which we call the SPMTU test.

#### *4.2.2 Shared PMTU Cache (SPMTU) Test*

In the SPMTU test, we use the presence of fragmentation to provide proofs about which of (Y, X1) and (Y, X2) is the true alias pair. By default, IPv6 routers do not fragment packets. However, an IPv6 router can be induced to fragment packets it originates if a host sends a Packet Too Big (PTB) message to the router claiming that the response from the router is too big for its link to handle [3]. The PTB sent by the host contains the claimed MTU, $M$, of the host's link. The router then makes an entry in its Path MTU (PMTU) cache, indicating that packets sent to the host need to be fragmented if their size exceeds $M$.

PMTU caches are commonly shared across all interfaces of a router, including routers manufactured by Huawei, Vyatta, HP, and Mikrotik [12]. When a router with a shared PMTU cache receives a PTB message from host $h$ with stated MTU $M$, it inserts an entry $(h, M)$ into its shared cache. As a result, all interfaces on the router will fragment subsequent packets that exceed $M$ to that host. We use evidence of shared PMTU caches as proof that a candidate alias pair is correct.

We determine which address pair in the candidate alias pair set shares PMTU caches by using the following procedure:

1. **Initialize**: The prober sends an ICMPv6 Echo Request of size $S$ to each of Y, X1, and X2, and verifies that all of them respond with an unfragmented Echo Reply. This step is necessary to ensure that none of the addresses has the prober's address in its PMTU cache.
2. **Populate cache**: If all addresses responded with an unfragmented Echo Reply in Step 1, the prober sends a PTB message with MTU $M < S$ to Y alone. If Y shares its PMTU cache with its aliases, all of them will fragment a packet of size $S$ sent to the prober.
3. **Resolve**: The prober sends an ICMPv6 Echo Request of size $S$ to each of Y, X1, and X2 again. If Y and X1 respond with a fragmented Echo Reply, and X2 responds with an unfragmented Echo Reply, we infer that Y and X1 share a PMTU cache, and must therefore be aliases. Conversely, if Y and X2 fragment and X1 does not, we infer that Y and X2 are aliases.

The SPMTU test is generic and can be applied to *any arbitrary pair* of IPv6 addresses to determine if they are aliases. However, it uses state in routers' caches and hence cannot be repeated with the same prober address until the PMTU cache entry for that prober address expires. We repeat tests using different prober addresses and rely on routers utilizing per-destination PMTU caches; thus a response from the router to a different prober address will not be fragmented. We own a /64 prefix, and use different addresses from the prefix for each test.

Although the SPMTU test can in theory be used as an all-pairs test, we are careful to use it only on candidate alias pairs from the harvest phase, as varying prober addresses may fill routers' caches with addresses from our tests. Since we send one PTB message per candidate alias pair set, the number of prober addresses in the PMTU cache will be at most the number of interfaces on the router.

## 5 Evaluation

In this section, we evaluate the accuracy of UAv6 against existing IPv6 alias resolution techniques and against limited ground truth from the Internet2 dataset [10]. We also show that a classic IPv4 alias resolution technique, the address-based technique (Section 2), works in IPv6, in spite of recommendations in RFC 4443 [6]. Finally, we combine the alias pairs found by UAv6 and the address-based technique and resolve 5,555 aliases in the Ark dataset [4].

### 5.1 Data Collection

We extracted 68,474 router interface addresses from traceroutes sent by CAIDA's IPv6 Ark project in July 2014 [4]. We found 56,803 /126 prefixes in total, and fed them into the harvest phase.

Recall that the harvest phase discards prefixes wherein the used addresses do not respond to our probes with ICMPv6 Echo Replies or the unused addresses do not elicit AU responses. Sometimes, AU responses do not arrive for the first ICMPv6 Echo Request; we therefore retransmit requests up to 3 times and each request has a timeout of 3 sec. Of the 56,803 prefixes, we did not receive ICMPv6 Echo Replies from X1 or X2 for 27,014 (47.6%) prefixes. For 7935 (14.0%) prefixes, we did not get AU responses from probes sent to X0 or to X3.

The remaining 21,854 (38.5%) prefixes are UAv6-applicable. We applied the harvest and disambiguation phases to them and found 15,260 alias pairs.

### 5.2 The Address-based Technique in IPv6

We discover that the address-based technique, a classic method of resolving aliases in IPv4 [9, 13, 18], works in IPv6, too. The address-based technique finds aliases in IPv4 by testing if UDP responses to high-numbered ports contain a different source address from the destination probed. The ICMPv6 specification states that if a message is sent to an address that belongs to a router, the source address of the ICMPv6 response must be that address [6]. If the specification is followed, the address-based technique would not work for IPv6.

However, we find that there exist routers that do not follow the specification: while running UAv6's harvest phase, we observed that some of the ICMPv6 Echo Replies to our probes had a different source address from the probed destination. This implies that the address-based technique also works in IPv6, so we investigated how often it applies. We sent UDP probes with high port numbers to all the addresses we probed in the harvest phase. UDP probes to 227,212 addresses provided 72,457 responses with ICMPv6 Port Unreachable responses. Among them, 8729 (12%) of the responses had a different source address from the destination of the UDP probes. Of the 89,756 ICMPv6 Echo Replies we received, 1450 (1.6%) had a different source address. In 1030 cases, both UDP and ICMP probes had a different source address in their response. In total, we discovered 9,143 alias pairs using the address-based technique.

| | Aliases discovered | TBT-applicable | TBT verified |
|---|---|---|---|
| Traceroute | 11,128 | 2810 (25.3%) | 2806 (99.86%) |
| SPMTU | 8422 | 1264 (15.0%) | 1263 (99.92%) |
| Union | 15,260 | 3539 (23.19%) | 3534 (99.86%) |

Table 1: Comparison of UAv6's accuracy against TBT for alias pairs where both addresses draw fragment IDs from sequential counters.

Although it is encouraging that the address-based technique works in IPv6, it has two drawbacks: first, it can only be applied to a small portion of the addresses, and second, it may not work in the future since it does not comply with the ICMPv6 specification. This serves as motivation for complementary techniques like UAv6.

### 5.3 Accuracy of UAv6

Alias resolution demands very high accuracy, as an incorrectly inferred alias may group two independent routers together, significantly altering the inferred topology. We next turn to evaluate UAv6's accuracy. For alias pairs to which the Too-Big Trick (TBT) is applicable, we use it for cross-validation. We also run UAv6 on the addresses from the Internet2 dataset [10] and verify the aliases it finds against ground truth.

#### *5.3.1 Comparison with TBT*

We first evaluate the accuracy of the SPMTU test and the Traceroute test against TBT. We can apply TBT to an address pair if both addresses' routers draw their fragment IDs from sequential counters. For aliases found by the Traceroute and SPMTU tests, we find TBT-applicable pairs and run TBT on them. Table 1 compares the accuracy of our tests against TBT.

**Traceroute test**: Using the Traceroute test, we find 11,128 alias pairs from 21,854 UAv6-applicable prefixes. Of them, 2810 pairs (25.3%) are TBT-applicable. All but 4 of these pairs (0.14%) are verified by TBT. We manually inspected these pairs and found that, although TBT indicates they have non-sequential fragment IDs, all 4 pairs are verified by the address-based technique. In future work, we plan to examine in greater depth why these established techniques contradict each other in some cases.

Recall that our central assumption is that if Y is the source of an AU response to a packet for X0 or X3, then Y is directly connected to the prefix containing X1 and X2. The Traceroute test provides us with some instances where this assumption is violated. For example, in 527 cases (2.41%), Y appears on the paths to both X1 and X2. In 55 other cases (0.25%), Y is more than one hop away from X1 or X2, which indicates that Y is not directly connected to the prefix. We detect these cases and discard them.

**SPMTU test**: The SPMTU test finds 8422 alias pairs. For the 1263 (15.0%) alias pairs where TBT could be applied, TBT verified all the alias pairs found by

SPMTU except one. We manually inspected this case and found that SPMTU no longer identified the pair as aliases. We recovered the fragment IDs that we had obtained when we first ran SPMTU upon them, and found that the fragment IDs for both addresses in that run had been sequential . We believe that one of the addresses from the pair was reassigned to another router in the sub-24 hour gap between our SPMTU run and our TBT run, causing the results to conflict.

**Comparison between disambiguation tests**: We now compare the aliases found by our disambiguation tests against each other. The union of alias pairs found by the SPMTU and Traceroute tests contains 15,260 pairs, and the intersection has 4289 pairs. There is one alias pair where the two tests conflict. The alias pair chosen by the traceroute test was confirmed by the address-based method, whereas the pair chosen by SPMTU was confirmed by TBT. We believe that this behavior is caused by a misconfigured router responding to probes not addressed to it.

**UAv6 is complementary to TBT**: We observe that 11,721 (76.8%) alias pairs found by UAv6 are not TBT-applicable, demonstrating that UAv6 is a complementary technique to TBT. For aliases found by the Traceroute test, we find that 74.7% are not TBT-applicable. 54% of these alias pairs do not respond with fragments after a PTB message and 46% respond with random fragments.

Like TBT, the SPMTU test also relies upon fragments received from the addresses. Yet SPMTU differs from TBT in that it relies upon shared PMTU caches in routers while TBT relies upon shared sequential counters from which the fragment ID is drawn. The majority of aliases found by the SPMTU test (85.0%) are not TBT-applicable. This implies that at least one of the addresses in the pair returned fragments not derived from a sequential counter. However, Speedtrap [12] had found in their tests that all routers which implemented shared PMTU caches also used sequential counters. We believe that at least one main router manufacturer is now implementing shared PMTU caches and non-sequential counters on its routers.

#### *5.3.2 Comparison with Ground Truth*

We next study UAv6's accuracy using ground truth data from the Internet2 network [10]. We obtained ground truth aliases from Internet2 routers' configuration files. We believe these aliases to be correct, although we omitted some interfaces that are not physical interfaces. The Internet2 topology consists of 579 interface addresses on 11 routers. We obtain the /126 prefix of each interface address and run the harvest phase upon the prefix. Of the 500 /126 prefixes from the Internet2 dataset, we find 62 (12.4%) candidate alias pair sets. The number is small since many prefixes in Internet2 did not respond in the harvest phase.

For each candidate alias pair set, we apply the disambiguation phase and show the results in Table 2. Not all aliases found by the tests could be verified: some aliases are aliases of routers *connected* to Internet2 routers, but not of the Internet2 routers themselves. For these aliases, we do not have ground truth, and thus cannot verify them. The Traceroute test found 31 such aliases and the SPMTU test found 22 of them. The Traceroute test found 6 alias pairs that

| | Aliases discovered | Aliases verifiable | Alias verified | Accuracy |
|---|---|---|---|---|
| SPMTU | 37 | 15 | 15 | 100.00% |
| Traceroute | 37 | 6 | 6 | 100.00% |

Table 2: Comparison of UAv6's accuracy against Internet2.

| | Discovered alias pairs | Routers with aliases | Resolved Ark aliases | Discovered aliases |
|---|---|---|---|---|
| UAv6 | 15,259 | 5711 | 4148 | 14,760 |
| Address-Based | 9143 | 5477 | 2091 | 9118 |
| Combined | 22,080 | 9307 | 5555 | 21,415 |

Table 3: Number of aliases found by the UAv6 and Address-Based techniques.

belonged to Internet2 routers, and the SPMTU test found 15 such pairs. All of these aliases were verified by ground truth, demonstrating UAv6's accuracy.

### 5.4 Alias Resolution with UAv6 and the Address-based Method

We close this section by investigating how many aliases each technique finds within the 68,474 router interface addresses extracted from the Ark project in July 2014 [4]. For this comparison, we use the *number of aliases* that each technique finds instead of comparing the number of alias pairs, because a router with $n$ interfaces has $\binom{n}{2}$ alias pairs, but only $n-1$ aliases. We believe this is an unbiased way of measuring the completeness of an alias resolution technique.

We combine the alias pairs we found using UAv6 and the address-based technique and show the results in Table 3. Though UAv6 found only 67% more alias pairs than the address-based technique, it found nearly double the aliases within the addresses already discovered by Ark. Of course, both UAv6 and address-based methods may discover new addresses that were not present in a traceroute measurement. Resolving aliases of interfaces already discovered by traceroute contributes accuracy to an inferred router-level map, while discovering new addresses yields additional detail. However, there were 1407 aliases that the address-based technique alone resolved. Combining the approaches yielded 34% more aliases than the use of UAv6 alone.

## 6 Conclusions

IPv6 deployment is on the rise and alias resolution techniques are vital in mapping its topology. In this work, we augmented existing alias resolution methods with UAv6: a new technique that uses partially used IPv6 prefixes to find aliases. We found potential alias pairs by probing /126 prefixes and introduced two tests to disambiguate potential alias pairs. Existing alias resolution techniques and ground truth from the Internet2 topology confirmed UAv6's accuracy. UAv6 is complementary to the address-based technique and to TBT, finding alias pairs that other techniques do not.

The disambiguation tests we employ in this work helped UAv6 recover aliases from 70% of applicable prefixes, and we believe this can be increased further. For instance, one area of future work is to employ other disambiguation tests, such as the Hop Limit on received packets, to find more aliases. Additionally, we believe that, through the use of multiple vantage points, UAv6 can harvest more applicable prefixes.


### Acknowledgments

We thank Matt Lentz and our anonymous reviewers for their comments and suggestions. This work was partially supported by NRL Grant N00173131G001.


## References


1. Abley, J., Savola, P., and Neville-Neil, G. Deprecation of type 0 routing headers in IPv6. *RFC 5095* (2007).
2. Bender, A., Sherwood, R., and Spring, N. Fixing Ally's growing pains with velocity modeling. In *ACM IMC* (2008).
3. Beverly, R., Brinkmeyer, W., Luckie, M., and Rohrer, J. P. IPv6 alias resolution via induced fragmentation. In *PAM* (2013).
4. CAIDA's IPv6 Ark Topology Data. http://www.caida.org/data/active/ipv6_allpref_topology_dataset.xml.
5. Claffy, K., Monk, T. E., and McRobb, D. Internet tomography. *Nature 7*, 11 (1999).
6. Conta, A., and Gupta, M. Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification. *RFC 4443* (2006).
7. Czyz, J., Allman, M., Zhang, J., Iekel-Johnson, S., Osterweil, E., and Bailey, M. Measuring IPv6 Adoption. In *ACM SIGCOMM* (2014).
8. Dhamdhere, A., Luckie, M., Huffaker, B., Elmokashfi, A., Aben, E., et al. Measuring the deployment of IPv6 Topology, routing and performance. In *ACM IMC* (2012).
9. Govindan, R., and Tangmunarunkit, H. Heuristics for Internet map discovery. In *INFOCOM* (2000).
10. Internet2 Topology. http://noc.net.internet2.edu/i2network/live-network-status/visible-network.html.
11. Kohno, M., Nitzan, B., Bush, R., Matsuzaki, Y., Colitti, L., and Narten, T. Using 127-Bit IPv6 Prefixes on Inter-Router Links. *RFC 6164* (2011).
12. Luckie, M., Beverly, R., Brinkmeyer, W., et al. Speedtrap: Internet-Scale IPv6 Alias Resolution. In *ACM IMC* (2013).
13. Pansiot, J.-J., and Grad, D. On routes and multicast trees in the Internet. *ACM SIGCOMM CCR 28*, 1 (1998), 41–50.
14. Qian, S., Wang, Y., and Xu, K. Utilizing Destination Options Header to Resolve IPv6 Alias Resolution. In *GLOBECOM* (2010).
15. Qian, S., Xu, M., Qiao, Z., and Xu, K. Route Positional Method for IPv6 Alias Resolution. In *ICCCN* (2010).
16. Savola, P. Use of/127 Prefix Length Between Routers Considered Harmful. *RFC 3627* (2003).
17. Sherwood, R., Bender, A., and Spring, N. Discarte: A Disjunctive Internet Cartographer. In *ACM SIGCOMM* (2008).
18. Spring, N., Mahajan, R., and Wetherall, D. Measuring ISP topologies with Rocketfuel. In *ACM SIGCOMM* (2002).
19. Waddington, D. G., Chang, F., Viswanathan, R., and Yao, B. Topology discovery for public IPv6 networks. *ACM SIGCOMM CCR 33*, 3 (2003), 59–68.