Resolving IP Aliases with Prespecified Timestamps

Justine Sherry* Ethan Katz-Bassett* Mary Pimenova*
Harsha V. Madhyastha† Thomas Anderson* Arvind Krishnamurthy*
*University of Washington †University of California, San Diego

ABSTRACT

Operators and researchers want accurate router-level views of the Internet for purposes including troubleshooting and modeling. However, tools such as traceroute return IP addresses. Because routers may have dozens of IP addresses, or *aliases*, multiple measurements may return different addresses, obscuring whether they represent the same machine. While many techniques exist to address this issue by identifying some IP aliases, these techniques, even in combination, find only a subset of alias pairs.

To improve this state, we design and evaluate a new alias resolution technique using the IP prespecified timestamp option. This option allows a sender to request timestamp values from multiple IP addresses in the same probe. By careful arrangement of these IP addresses, we show that we can infer aliases in many cases.

In this paper, we conduct a measurement study of how many routers support IP timestamps, demonstrating that enough honor the option to base our technique on it. Using our technique, and compared to the most accurate alias information available, we find that 94.7% of the aliases identified by our technique are true positives. Further, we show that our IP timestamp-based technique complements existing alias resolution techniques, providing significant gains by discovering previously unidentifiable aliases.

Categories and Subject Descriptors

C.2.3 [**Computer Communication Networks**]: Network Operations—*Network Monitoring*; C.2.1 [**Computer Communication Networks**]: Network Architecture and Design—*Network topology*

General Terms

Measurement, Experimentation

Keywords

Alias resolution, IP timestamp, IP options

1. INTRODUCTION

Internet topology measurements from tools like traceroute serve numerous purposes. Network operators and researchers use them to pinpoint outages or failures [22, 12, 26]. Measurements can reveal the underlying structure and evolution of the Internet [7, 21, 23, 24]. Researchers use them as the basis for performance comparisons between ISPs [18], Internet latency estimates [6, 4, 16], and IP geolocation [25, 12].

Because each router has more than one IP address, multiple measurements of the same router can appear disjoint to IP-level tools, making it more difficult to draw conclusions from those measurements. To correct this, one must accurately map from IP addresses to routers. We refer to the multiple IP addresses of a router as its aliases, and the process of mapping from IP addresses to routers as alias resolution.

While considerable progress has been made in building better tools [3, 8, 9, 20, 21] for alias resolution, existing techniques are still limited in scope. For example, RadarGun leverages the fact that, on some routers, the multiple interfaces share a single linearly increasing IPID counter. However, the authors of RadarGun observed only 31% of addresses exhibiting such a counter [3]. Mercator, another technique, relies on the fact that some routers respond to certain probes with a response sourced from an address different from the original destination of the probe, implying the two addresses both belong to the router [8]. However, most routers do not respond in this way: only 66% of addresses respond to such probes at all, and only 23% of those respond from an address other than the original destination [10].

In this paper, we present a new technique for IP alias resolution that uses the IP prespecified timestamp option. This option allows a sender to request timestamp values from up to four IP addresses along the path, all in a single packet. Our technique identifies alias pairs by constructing a single probe with timestamp requests for two suspected aliases. By careful interleaving of the alias addresses, and a small amount of repeated probing, we show that we can very reliably identify aliases. Our technique potentially applies to any router capable of providing multiple timestamps in the same probe; we find 32.4% of IP addresses in our measurements respond in this way. Compared to a set of the most accurate alias information, 94.7% of aliases declared by our technique are true positives, and of those we declare non-aliases only 2.4% are misclassified. Further, we find that our technique complements existing techniques: 76.7% aliases we discover were unidentifiable using previous techniques.

The rest of the paper is organized as follows. In Section 2, we describe the timestamp option and quantify the degree of router support for the option. In Section 3 we describe our alias resolution technique. Section 4 includes evaluations of our alias resolution technique for accuracy and coverage. We discuss related work in Section 5 before concluding in Section 6.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
IMC'10, November 1–3, 2010, Melbourne, Australia.
Copyright 2010 ACM 978-1-4503-0057-5/10/11 ...$10.00.

2. PRESPECIFIED TIMESTAMP OPTION

We next describe the timestamp option, and we provide a measurement study to discover how often routers respond to the timestamp option in ways that we can exploit for alias resolution.

2.1 Background

Timestamp requests are an optional extension to any packet that traverses the Internet [1]. Three 'flags' specify different behaviors of the IP timestamp option; we make use of only one, 'prespecified timestamps.' For a packet with the prespecified timestamp option, the sender lists up to four IP addresses in the options header. As a shorthand, we describe timestamp requests in the following format: $(A|BCDE)$, where A is the destination of the probe, and B, C, D, and E are the ordered list of prespecified addresses. Each machine which forwards the packet will check to see if the first unstamped IP address is one of its own, and, if so, it will record a timestamp in the packet header before forwarding the packet. The order is important: if C receives the packet, it will only record a timestamp if B has already done so.

2.2 Measurement Study of Support

Although timestamps are a standard option in the IP specification, many routers do not honor them and some ISPs filter them. Further, the protocol specification does not dictate a behavior for routers to follow upon encountering their own IP address multiple times in the same probe. Our technique, described in detail in Section 3, relies on a single router providing multiple timestamps for multiple addresses in the same probe. Hence, our technique is only applicable to the subset of routers whose implementation provides multiple stamps in this case.

To assess how many routers honor such requests, we performed a measurement study of all IP addresses discovered by iPlane on May 10, 2010 [15]. iPlane issues daily traceroutes from all PlanetLab sites [2] to approximately 140,000 prefixes. To generate our data set, we gathered all 351,214 IP addresses observed in iPlane traceroutes from May 10, then removed all private addresses and addresses in prefixes whose operators opted out of our experiments. We then issued ICMP echo requests (pings) to each remaining address and report on the 267,736 addresses that responded.

For these 267,736 addresses, we first sent each address D an ICMP echo request with the timestamp option enabled requesting $(D|DXXX)$. X is an address at the University of Washington known not to be on the path, to ensure that extra, invalid stamps were not being introduced into the responses we received. In our previous work, we found that some PlanetLab sites receive few or no timestamp responses, presumably due to filters [11]. To attempt to avoid such filters, we sent the probes from multiple PlanetLab vantage points. Further, to account for packet loss, we sent each measurement five times redundantly from each vantage point. As seen in Table 1, we did not receive any responses from 31% of addresses. An additional 15.5% of addresses responded to our probes without recording a timestamp value. We also identified a common faulty implementation, exhibited by 5.8% of addresses, in which the router recorded two stamps on encountering its address, even though the second prespecified address did not belong to it. The table lists this as *Extra Stamp* behavior.

Classification	IP Addresses	%
Unresponsive	83002	31.0%
Extra Stamp	15606	5.8%
Zero Stamps	41422	15.5%
One Stamp	40886	15.3%
Two Stamps	59450	22.2%
Three Stamps	40	0%
Four Stamps	27330	10.2%
Total	267736	100%

Table 1: Responsiveness to timestamp probes for the set of 267,736 public, ICMP-responsive addresses discovered by iPlane on May 10, 2010. Each address D was sent probes requesting $(D|DXXX)$ and $(D|DDDD)$. The classifications reflect the number of timestamps provided in D's responses. Our alias resolution technique potentially works for targets responding with 2-4 stamps.

The remaining 47.7% of addresses correctly responded to these probes with a single timestamp. To characterize their behavior in response to multiple requests for their own address, we next sent each of them a probe request $(D|DDDD)$; the table indicates how many stamps the addresses responded with. The most common stamping behavior was to provide two stamps; 22.2% of addresses demonstrated this behavior. A further 10.2% of addresses provided four stamps. Since our alias technique will require a router to record multiple stamps in a single probe, it potentially applies to the 32.4% of ping-responsive addresses in our dataset that responded with two or more timestamps. This is an upper bound on the applicability of our technique; we show later that some routers that stamp twice do not always stamp for queries involving different interfaces.

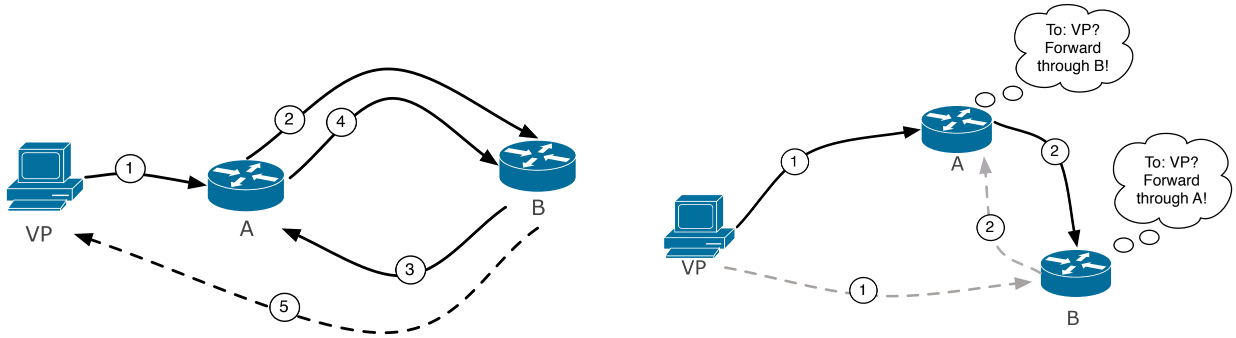
3. ALIAS CONFIRMATION

The prespecified timestamp option allows a single packet to request timestamps from multiple IP addresses. Our technique nests requests for pairs of IP addresses. To check if A and B are aliases, we send two ICMP echo requests, with the timestamp options set to $(A|ABAB)$ and $(B|BABA)$. If they are aliases of the same router and the router stamps multiple times, the router should record timestamps for both addresses at a single point along the path, with consistent timestamp values. To identify this, we look for timestamp replies whose ordering constrains A and B to reside on the same router, and whose timestamp values are identical.

3.1 Timestamp Values as a Fingerprint

Timestamp values allow us to identify whether two IP addresses are likely to share a common clock. A single host recording multiple timestamps at once into a single packet should provide the same value for each stamp.

Making use of this observation, when we receive a response containing timestamps for candidates A and B, we check the difference between the timestamp values. However, in our study in Section 2.2, we observed rare cases when timestamp values incremented between stamps, although the timestamps are provided by the same host (in response to $(D|DDDD)$). To account for this, we allow some leeway. We send duplicates of our probes to generate multiple timestamped responses. If fewer than 90% of the responses received for a pair have all timestamps equal or



(a) Configuration of two distinct routers providing four stamps to the request $(A|ABAB)$.

(b) Configuration of two distinct routers providing two stamps to both the request $(A|ABAB)$ (solid line) and the request $(B|BABA)$ (dashed).

Figure 1: Two configurations of looped routing implied by certain timestamp responses. These cases can only be transient. Thus, repeatedly receiving these responses implies that the IP pairs are likely to be aliases.

if we observe decrements between stamps, we eliminate the pair and declare it to be a non-alias. If the timestamps are consistent, however, we require further evidence.

3.2 Topological Constraints

Because a prespecified address will only be stamped after all previous addresses have been stamped, the ordering of the timestamps recorded must be consistent with the path the packet traverses. By carefully arranging the IP addresses in the timestamp request, we can make it highly likely that only true aliases will respond with a given pattern. The absence of the pattern does not necessarily mean that the addresses are not aliases, just that we cannot determine their relationship.

Loops. While it is possible for a packet to encounter a transient loop, such cases should be rare and can be accounted for via redundant measurements over time. If a packet encounters a persistent loop, we should not receive an echo reply for our probe. Hence, if we receive multiple replies each of which implies a loop, we can conclude that the packets (in all likelihood) did not travel back and forth between two routers, but rather that a single router stamped multiple times for its multiple interfaces.

There are two separate scenarios in which we can infer aliases because the alternative would be a persistent loop. The first occurs when we receive four stamps in response to any of our queries. Figure 1(a) illustrates the route taken by a timestamp request $(A|ABAB)$, assuming A and B are not aliases. Timestamp values for all four prespecified addresses suggest that the packet reached the destination A, then went to B, to A, and to B again. These stamps mean that B must be configured to forward the return traffic en route to S through A, and A must be configured to forward the return traffic en route to S through B. This is a loop, or, if we see this pattern repeated for multiple probes over time, it is more likely that A and B are aliases.

The second scenario, demonstrated in Figure 1(b), occurs when a single vantage point S receives responses with two stamps for both the request $(A|ABAB)$ and the request $(B|BABA)$. If A and B are not aliases, these responses imply that B routes traffic to S through A, and A routes traffic to S through B. Because this explanation is once again a loop, if we see this pattern repeated over multiple probes, we can conclude that A and B are likely to be aliases on the same machine.

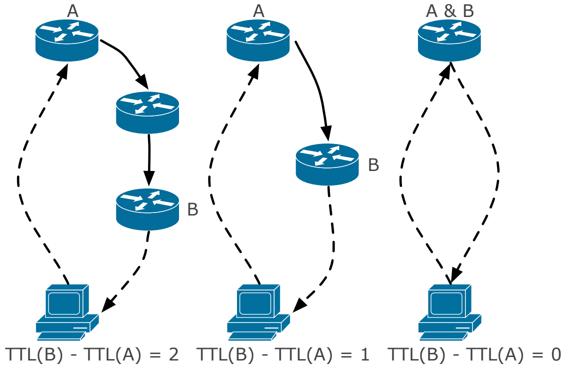


Figure 2: Because TTL values are initialized to a small number of standard values, we can estimate the reverse path length to S from routers A and B. If we know that the reverse path from A traverses B and we assume that the routers are distinct, B's reverse path length to S must be shorter than A's. If their reverse path lengths are the same, A and B likely reside on the same router.

Distance. In our measurements, a single vantage point will often receive two timestamps for $(A|ABAB)$, but not receive any for $(B|BABA)$. In this situation, we are able to infer aliases by coupling the topological relationship implied by the timestamps with reasoning about the relative distance of A and B from the vantage point.

Given that the vantage point S received the first two timestamps in the request $(A|ABAB)$, we know that either A and B are aliases, or that B is on the reverse path from A back to S. A's reverse path length should be equal to B's reverse path length, plus some difference δ representing the number of hops in between A and B along the reverse path. If this δ value is 0, it suggests that there are no hops between A and B, and A and B are likely to reside on the same router. By sending extra pings to A and B, we can acquire TTL values from the response packets that allow us

	Total	True aliases	Non-aliases
Classified as aliases			
Four Stamps	14810	14025 (94.7%)	785 (5.3%)
Two Stamps (Loop)	354	349 (98.6%)	5 (1.4%)
Two Stamps (Distance)	2580	2422 (93.9%)	158 (6.1%)
→ Total classified as aliases	17744	**16796 (94.7%)**	**948 (5.3%)**
Classified as non-aliases			
Bad Clock	17046	**412 (2.4%)**	16634 (97.6%)
Unclassified			
Two Stamps (Unclassified)	8440	729 (8.6%)	7711 (91.4%)
Overall			
All pairs with multiple stamps	43230	17937 (41.5%)	25293 (58.5%)

Table 2: Results and accuracy of timestamp confirmed alias pairs from the `mrinfo` dataset. The candidate pairs included 43,230 two stamping pairs, 17,937 of which were true aliases. We report the accuracy of our categorizations, showing the true positives (16,796), false positives (948), and false negatives (412) in bold.

to estimate the length of the paths taken from A and B back to the source. This inference can be confounded if A and B use different initial TTL values; however, most routers choose their initial value from a small set of standard values. One other source of inaccuracy may arise from 'hidden routers' [20], which do not decrement the TTL value when forwarding the packet, thus violating our TTL argument described in Section 3.2.

Figure 2 shows the relationship between A and B given a δ value of 2, 1, or 0.

3.3 Algorithm

To take advantage of the previously described aliasing characteristics, we perform the following steps to confirm alias pairs.

First, we send a preliminary $(A|AXXX)$ probe to all IP addresses A involved in any candidate pair, and we remove any addresses which display the *Extra Stamp* behavior described in Section 2.

Then, for each candidate pair we send probes $(A|ABAB)$ and $(B|BABA)$ redundantly five times each from several vantage points. We require the redundancy in order to generate enough responses for our 90% shared-clock threshold, as well as to compensate in case some packets are dropped or filtered. Next, we trim our candidates to the set of pairs which responded to our measurements at least five times with two or more timestamps.

Then, we split the candidates by stamp count. We classify those which stamp four times as aliases. For those which stamp twice, we trim the set to those which pass our shared clock test: those which never display a decrement between stamps, and for which at least 90% of the timestamp values are the same. We declare non-aliases those which do not pass this test.

Finally, for those which pass the clock test we identify pairs which exhibit either two stamp (loop) or two stamp (distance) configurations to any single vantage point. To evaluate the 'distance' configuration, we require TTL values from addresses which may not have responded to our timestamp probes, hence we send extra pings to the candidate addresses. Those which display these configurations, we declare aliases. Those which do not, we characterize as unknown.

4. EVALUATION

To assess the accuracy of our technique, we use a dataset of known aliases generated by `mrinfo` measurements on December 30th, 2009 [19]. `mrinfo` identifies aliases by issuing IGMP `ASK_NEIGHBORS` requests to multicast-enabled routers. Multicast-enabled routers respond to such requests with a list of their interfaces, associated IP addresses, and neighboring routers. Assuming the configuration on the router is correct and up-to-date, the alias information should be correct. However, the resulting datasets are limited in several ways: they only include multicast-enabled routers, they are limited to routers which are connected across Mbone to the vantage point used to initiate the requests, and they are restricted to networks which do not filter IGMP traffic. After removing private and blacklisted addresses, the December 30th set contains 9,130 addresses over 1,635 routers. Because of the dataset's limited size and biases introduced by its limitations, it has a different distribution of aliasing behaviors than the iPlane dataset in Section 2.2. More specifically, 21% of addresses in it provide four stamps, while only 14.4% provide two.

4.1 Accuracy

Timestamp-based alias resolution requires candidate alias pairs as inputs. We generated candidate pairs by clustering together IP addresses with a similar vector of TTL values observed in pings from several hundred PlanetLab vantage points, a technique borrowed from iPlane [15]. The intuition for this heuristic is that we only need to consider addresses as potential aliases if they are at similar distances from most locations. By only considering pairs that clustered together, we eliminated roughly 99% of the potential pairs from consideration.

After clustering, we were left with 874,699 candidate pairs. Within those candidate pairs were 32,853 alias pairs. Of these actual aliases, 18,939 included at least one IP address which had previously responded to timestamp requests with multiple stamps, as we described in section 2.2. These 18,939 pairs are an upper bound on the number of aliases our technique could potentially discover.

To each of the 874,699 candidate pairs we sent two probes: $(A|ABAB)$ and $(B|ABAB)$. Of the candidates pairs that we probed, 576,068 pairs responded to at least one probe, 43,230 with two or more stamps. Note that many of the pairs respond with fewer than two stamps because they are not aliases: a response to $(A|ABAB)$ might include only a stamp for A because B is not an alias and is not on the reverse path. Within those 43,230 responsive pairs were 17,937 true alias pairs, 94.7% of the 18,939 pairs we could potentially address. Table 2 displays the categorization and accuracy of pairs which provided two or more stamps.

We classified 17,744 of the pairs as aliases. Overall 16,796 of the 17,744 aliases (94.7%) we found are true positives, according to the `mrinfo` data. It is possible that some of the 5.3% false positives are due to out-of-date `mrinfo` configurations or changes in configuration between the generation of the `mrinfo` data in December 2009 and our experiments in May 2010.

The 16,796 true alias pairs we identified represent 88.7% of the original 18,939 alias pairs our technique could potentially find. We note that we had much more success in identifying alias pairs for addresses which stamped four times than for those that stamped twice. Of those pairs for which one of the addresses stamped four times (when queried with its own address four times), we correctly identified 94.3% of them as aliases. However, for those pairs for which neither address stamped four times, but at least one of the addresses stamped twice, we only identified 63.4% of the pairs as aliases.

Of the 17,046 pairs our technique declares to not be aliases, 412 are classified as aliases by `mrinfo`, only 2.4% false negatives. Over half of these false negatives included a clock decrement between stamps.

4.2 Coverage

To further evaluate our technique, we assess whether it identifies aliases not found by RadarGun [3] and Mercator [8], two existing, widely-used techniques. Similar to how our technique relies on routers stamping multiple times, these techniques also rely on specific router behaviors, limiting their applicability. We look to the overlap of the aliases discovered by each technique, to assess whether by combining techniques we may be able to do better than any single technique on its own. We discuss combined techniques further in related work.

For our evaluation, we again use `mrinfo` as a basis for comparison. It is possible that the selection bias of this data influences the relative effectiveness of the different techniques. For example, RadarGun relies on routers using linear IPID counters, and we found that a higher proportion of addresses in the `mrinfo` dataset returned random IPID values compared to the RadarGun study (also of just over 9000 addresses) [3]. Similarly, as we noted above, a higher proportion of addresses in this dataset will stamp four times, compared to the broader dataset used in Section 2.2, increasing our relative effectiveness.

We evaluate each technique under the condition where all true pairs are included as candidates. Thus, we provided each technique all of the IP addresses as inputs (for those which take addresses as input) or all of the true alias pairs as candidates (for those which require candidate pairs). Because some techniques may not find all combinations of pairs (identifying (A, B) and (B, C) as pairs, but not (A, C)), we take the transitive closure of all pairs confirmed, providing us with a complete set of implied aliases for each technique.

Figure 3 shows a CCDF demonstrating the per-router coverage provided by the techniques individually and by all techniques merged. The metric evaluates how completely the techniques identify all alias pairs for the routers in the dataset. The vertical axis provides a complementary cumulative fraction of the 1,635 `mrinfo` routers, and the horizontal axis provides the fraction of alias pairs per router successfully identified by the alias resolution technique. A given (x, y) point shows that a technique identified at least x fraction of pairs for y fraction of the routers. Some routers in the dataset had many more interfaces than others: 75% had six or fewer addresses, and the largest had 70 addresses.

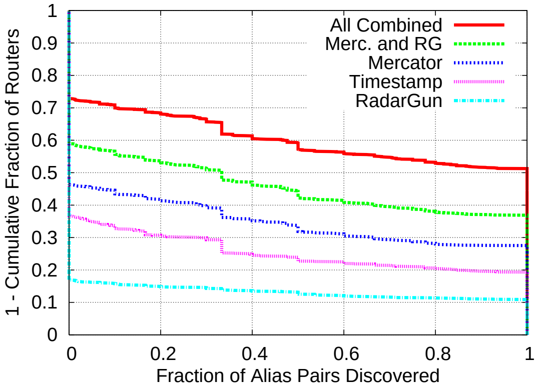


Figure 3: Comparison of coverage of alias resolution techniques over `mrinfo` dataset. Graph is a CCDF showing the fraction of routers for which the techniques identify a given fraction of alias pairs. Merging successive techniques improves the proportion of identifiable alias pairs, for this dataset.

The merger of the three techniques provides much greater coverage than any technique in isolation for this dataset. With all techniques merged together, 52.3% of routers have their aliases resolved entirely (100% of pairs identified) and 72.9% are resolved partially (some, but not all of their alias pairs are identified by the alias resolution). In aggregate, 68.3% of all alias pairs within the dataset are identified by one or more technique. Furthermore, the gains provided by including each successive technique are large. In particular, adding our technique to the merger of RadarGun and Mercator increases the number of routers that are completely resolved from 36.9% to 52.3%. In fact, 76.7% of the pairs discovered by our technique are unidentified by either RadarGun or Mercator.

5. RELATED WORK

We next discuss related work which has dealt with alias resolution or measurements using timestamps.

5.1 Alias Resolution

Many techniques exist to address the IP alias resolution problem, but none have complete coverage or offer a complete solution.

Mercator sends a UDP probe to a high numbered port, generating an ICMP Port Unreachable error message. On some routers, the error message is sourced from an address different from the destination of the original UDP probe, revealing that the two addresses reside on the same router [8]. Mercator cannot discover aliases for routers that respond from the original destination address.

Ally sends a succession of probes to suspected aliases and analyzes the IPID value in response packets. Since some routers use a common counter to set IPID values, Ally declares a pair of addresses aliases if subsequent probes interleaved between the two addresses remain in the same range [21]. RadarGun borrows Ally's IPID insight and improves on its technique to reduce measurements from $O(n^2)$ probes to $O(n)$ probes. RadarGun does this by probing each IP address several times to generate a 'velocity model' of how the address's IPID values change over time. It then compares the velocity models between all pairs of addresses to identify aliases [3]. However, many routers do not have well-behaved linear counters, leaving both Ally and RadarGun unable to identify their aliases.

Whereas our technique, as well as the others discussed so far, classify aliases by probing the IP addresses in question and inspecting the replies, DisCarte [20] and APAR [9] base their techniques on topological relationships observed in traceroute and similar probes. The IP record route option often discovers different IP addresses than traceroute, even along the same path, and DisCarte attempts to align the two types of probes to resolve aliases [20]. APAR (Analytic and Probe-based Alias Resolver) uses common IP address assignment schemes to analyze traceroutes and infer aliases. The developers of APAR found that approximately half of the alias pairs they discovered were also found by Ally [9], whereas we found that 76.7% of the pairs we discovered in the `mrinfo` dataset were not found by the Ally-based RadarGun or by Mercator. An interesting question for the future is how well DisCarte and APAR combine with our technique.

We are not the first to suggest combining techniques. While we showed in Section 4.2 that combining techniques can provide more complete alias information, challenges remain for such a technique to be successful. Efficient probing, resolving conflicts between techniques, and limiting the impact of false positives all remain unresolved questions for a complete aliasing system [13].

5.2 IP Timestamps

Fonseca et al. investigated whether packets with timestamp or other IP options were dropped on paths between PlanetLab sites. They found that approximately half the paths dropped options packets, but that most of the drops were at the edge of the network, concentrated in a small number of ASes [5]. We measure from multiple vantage points to counteract this, an approach we also used in earlier work [11]. Since we are targeting routers, rather than end hosts, many of our destinations are in the core, where Fonseca et al. found little filtering. Still, our technique will not work for destinations that do not provide timestamps or that are completely behind filters.

In earlier work, we used prespecified timestamps as part of our reverse traceroute [11] and iPlane [14] systems. Neither system used timestamps for alias resolution.

In addition to the IP timestamp option that we use, there is an ICMP Timestamp request message. With this message, a sender can request the timestamp of the destination, but not of other IP addresses, so it cannot be used for our alias resolution technique. Tulip uses ICMP Timestamp messages to help diagnose performance problems [17].

6. CONCLUSION

Despite numerous efforts to develop alias resolution techniques, comprehensive IP alias resolution remains a challenge. We have demonstrated a new technique for IP alias resolution using IP prespecified timestamp measurements, a standard IP option. Our technique requests timestamps from both addresses in a candidate alias pair in a single probe. We have shown that many routers honor IP timestamp requests, and 32.4% of routers in our dataset will provide two or more timestamps in response to a request for their own IP address multiple times in the same probe. We found that timestamp-based alias resolution provides accurate results for routers that support it: in our study, 94.7% of alias pairs identified were true positives. In addition, timestamp alias resolution provides gains in overall coverage by discovering alias pairs which existing techniques failed to identify.

Acknowledgments

We would like to thank the anonymous IMC reviewers for their valuable comments and feedback. We would also like to thank Adam Bender for modifying the RadarGun tool for our use and Colin Scott for comments on an early draft of this paper.

7. REFERENCES

[1] RFC 791: Internet Protocol, September 1981.

[2] Bavier, A., Bowman, M., Chun, B., Karlin, D. C. S., Muir, S., Peterson, L., Roscoe, T., Spalink, T., and Wawrzoniak, M. Operating system support for planetary-scale network services. In *Network Systems Design and Implementation* (2004).

[3] Bender, A., Sherwood, R., and Spring, N. Fixing Ally's growing pains with velocity modeling. In *Internet Measurement Conference* (2008).

[4] Costa, M., Castro, M., Rowstron, A., and Key, P. PIC: Practical Internet Coordinates for distance estimation. In *International Conference on Distributed Computing Systems* (2004).

[5] Fonseca, R., Porter, G., Katz, R., Shenker, S., and Stoica, I. IP options are not an option. Tech. rep., EECS Department, University of California, Berkeley, 2005.

[6] Francis, P., Jamin, S., Jin, C., Jin, Y., Paxson, V., Raz, D., Shavitt, Y., and Zhang, L. IDMaps: A global Internet host distance estimation service. In *INFOCOM* (2000).

[7] Gao, L. On inferring autonomous system relationships in the Internet. *IEEE/ACM Transactions on Networking* (2000).

[8] Govindan, R., and Tangmunarunkit, H. Heuristics for Internet map discovery. In *INFOCOM* (2000).

[9] Gunes, M., and Sarac, K. Resolving IP aliases in building traceroute-based Internet maps. *IEEE/ACM Transactions on Networking* (2009).

[10] Hyun, Y. Personal Communication, May 2010.

[11] Katz-Bassett, E., Madhyastha, H. V., Adhikari, V., Scott, C., Sherry, J., van Wesep, P., Anderson, T., and Krishnamurthy, A. Reverse traceroute. In *Network Systems Design and Implementation* (2010).

[12] Katz-Bassett, E., Madhyastha, H. V., John, J. P., Krishnamurthy, A., and Anderson, T. Studying black holes in the Internet with Hubble. In *Network Systems Design and Implementation* (2008).

[13] Keys, K. Internet-scale IP alias resolution techniques. *ACM SIGCOMM Computer Communication Review (CCR)* (2009).

[14] Madhyastha, H. V. *An Information Plane for Internet Applications*. PhD thesis, Univ. of Washington, 2008.

[15] Madhyastha, H. V., Isdal, T., Piatek, M., Dixon, C., Anderson, T., Krishnamurthy, A., and Venkataramani, A. iPlane: An information plane for distributed services. In *Operating Systems Design and Implementation* (2006).

[16] Madhyastha, H. V., Katz-bassett, E., Anderson, T., Krishnamurthy, A., and Venkataramani, A. iPlane Nano: Path prediction for peer-to-peer applications. In *Network Systems Design and Implementation* (2009).

[17] Mahajan, R., Spring, N., Wetherall, D., and Anderson, T. User-level Internet path diagnosis. In *Symposium on Operating Systems Principles* (2003).

[18] Mahajan, R., Zhang, M., Poole, L., and Pai, V. Uncovering performance differences among backbone ISPs with Netdiff. In *Network Systems Design and Implementation* (2008).

[19] Mérindol, P., den Schrieck, V. V., Donnet, B., Bonaventure, O., and Pansiot, J.-J. Quantifying ASes multiconnectivity using multicast information. In *Internet Measurement Conference* (2009).

[20] Sherwood, R., Bender, A., and Spring, N. DisCarte: A disjunctive Internet cartographer. In *SIGCOMM* (2008).

[21] Spring, N., Mahajan, R., and Wetherall, D. Measuring ISP topologies with Rocketfuel. In *SIGCOMM* (2002).

[22] Steenbergen, R. A. A practical guide to (correctly) troubleshooting with traceroute. In *NANOG 45* (2009).

[23] Subramanian, L., Agarwal, S., Rexford, J., and Katz, Y. H. Characterizing the Internet hierarchy from multiple vantage points. In *INFOCOM* (2002).

[24] Tangmunarunkit, H., Govindan, R., Jamin, S., Shenker, S., and Willinger, W. Network topology generators: Degree-based vs. structural. In *SIGCOMM* (2002).

[25] Wong, B., Stoyanov, I., and Sirer, E. G. Octant: A comprehensive framework for the geolocalization of Internet hosts. In *Network Systems Design and Implementation* (2007).

[26] Zhang, Y., Mao, Z. M., and Zhang, M. Effective diagnosis of routing disruptions from end systems. In *Network Systems Design and Implementation* (2008).