Analytical IP Alias Resolution

Mehmet Gunes
Department of Computer Science
University of Texas at Dallas
Richardson, TX 75080, USA
Email: mgunes@student.utdallas.edu

Kamil Sarac
Department of Computer Science
University of Texas at Dallas
Richardson, TX 75080, USA
Email: ksarac@utdallas.edu

***Abstract*— IP alias resolution is an important step in generating sample Internet topologies from collected path traces. Inaccuracies in IP alias resolution may significantly affect the characteristics of the resulting sample topologies. This in turn affects the accuracy of measurement results obtained using such topologies. Existing tools for alias resolution use an active probing approach. They induce significant traffic overhead into the network and critically depend on the participation of the routers. Recent studies have reported on the limited effectiveness of these approaches [1], [2].**

In this paper, we present a novel approach, called Analytical Alias Resolver (AAR), for IP alias resolution. Given a set of path traces, AAR utilizes the common IP address assignment scheme to infer IP aliases from the collected path traces. AAR does not incur traffic overhead due to active probing for alias resolution. Our experimental evaluations on a set of collected Internet path traces show that, compared to existing approaches, AAR can detect significantly more number of IP aliases.

I. Introduction

Internet's core is a web of interconnected backbone networks of high speed. Thousands of small and medium size Autonomous Systems (ASes) interconnect individuals, businesses, universities, and agencies over this backbone. Global overview of the Internet infrastructure is rather complex yet Internet continues its growth with no central authority to regulate.

Understanding the topological and functional characteristics of the Internet is an important research issue. This understanding is essential to verify, correct, and improve various desirable properties of the Internet including its robustness, reliability, efficiency, and security. Moreover, this understanding helps in designing and introducing new applications and services without breaking or negatively affecting the existing ones. As a result, several Internet measurement studies have been conducted to investigate numerous characteristics of the Internet topology and its functionality.

Most Internet measurement studies require the availability of a representative Internet map. Nonetheless, no complete map depicting Internet's router level structure or topology exists. There are several projects/efforts that collect large scale Internet maps. These include the Skitter [3] project of CAIDA, AMP [4] project of NLANR, Scriptroute [5] measurement infrastructure of PlanetLab, DIMES project of Tel Aviv University (`www.netdimes.org`), and several others [6], [7].

Most of these projects utilize the well-known *traceroute* software for topology collection. Traceroute returns the path to a given destination by tracing the routers in between. Traceroute uses TTL-scoped probe packets to obtain ICMP error messages from the routers on the path. Typically, each ICMP error message includes the IP address of one of the interfaces of the router as its source IP address. By collecting the source IP addresses from the incoming ICMP error messages, traceroute returns the path information as a sequence of IP addresses each representing a router between the local system and the remote destination.

After collecting the path traces, the next step is to build a network map from the collected information. This step requires identification and grouping of IP addresses belonging to the same router, a task often referred to as *IP alias resolution*. Alias resolution is an important yet often overlooked component of traceroute-based Internet map construction process. Inaccuracies in alias resolution may result in a network map that (1) includes artificial links/nodes and/or (2) misses existing links/nodes in the sample topology. Consequently, flaws in the resulting network map cause inaccuracies in the conclusions derived from the studies that use this map [1], [2], [8], [9].

As an example, the authors of [1] identified IP alias resolution as the main reason for the limited success of the topology discovery method used in the Rocketfuel study [7]. Similarly, in our recent study on the intersection characteristics of Internet paths and trees, we observed considerable effect of alias resolution on the final results [2]. In the same study, we demonstrated the effect of IP alias resolution on the validity of recently proposed sampling bias tests [8] that are used to check the representativeness of collected network topologies.

In this paper, we present a novel approach for IP alias resolution. We develop an analytical approach, *Analytical Alias Resolver (AAR)*, that utilizes the common IP address assignment scheme to infer IP aliases from the collected path traces. Compared to the existing approaches that use active probes to find alias pairs, AAR does not introduce any probe traffic into the network. In addition, AAR can be used together with the existing approaches to significantly improve the overall success rate and reduce the probing overhead in resolving IP aliases. Our preliminary evaluations that compare the success rate of AAR with that of the existing approaches show that AAR can perform significantly better than the others depending on the collected path traces.

The rest of the paper is organized as follows. Section II summarizes the existing approaches to IP alias resolution. Section III presents some observations that are used in the development of AAR. Section IV describes Analytical Alias Resolution and Section V presents an evaluation of AAR approach. Finally, the paper concludes in Section VI.

II. Related Work

A. Existing Approaches for IP Alias Resolution

In this section we present the existing IP alias resolution approaches in four groups:

Address Based Method

This approach uses active probes to resolve IP address aliases. It is based on a certain implementation of ICMP error reporting mechanism at the routers [10]. It assumes that when a router generates an ICMP error message to be sent to a remote system S, the router uses the IP address of the interface that is on the shortest path to S as the source IP address in the ICMP error message. In order to check if two given IP addresses, say IP_A and IP_B, belong to the same router, one can generate and send probe packets to each IP address and compare the source IP addresses in the returned ICMP error messages.

Address based method is fairly simple and is quite powerful in resolving IP aliases. On the other hand, it depends on a particular implementation of ICMP error reporting mechanism which may change from one router vendor to another. In addition, network operators can configure their routers to use alternative approaches in generating and sending ICMP error messages.

Mercator [11] and *iffinder* (`www.caida.org/tools`) are two well known IP alias resolution tools that use address based method. They send probe packets to each IP address. If replies from interfaces with different IP address have the same source IP address, these IP addresses are classified as aliases. *iffinder* may discover additional aliases comparing (1) outgoing interface addresses stored in the packet by enabling IP record route option (RFC-791) and (2) incoming interface address of the router replying to port unreachable error.

IP Identification Based Method

This method relies on the IP identification field values in IP protocol header of the returned ICMP error messages for alias resolution. When an IP packet is generated, the kernel puts a 16-bit value into the IP identification field in the IP header. Typically IP identification is implemented as a monotonically increasing counter. It is mainly used for identifying multiple fragments of the packet at the destination in the case of IP packet fragmentation. Since IP identification counter is monotonically incremented, successive packets originating from the same router have consecutive IP identification numbers.

This mechanism is used to identify if the ICMP error messages are coming from the same router after sending probe messages to different IP addresses. If the returned ICMP error messages have consecutive or close by IP identification numbers, this is an evidence that the messages are created by the same router. The experiment can be repeated several times to increase the level of confidence in the decision.

This approach is used in the recent *ally* [7] tool. *Ally* combines several techniques that seek peculiar similarities between responses to probes. Mainly, *ally* extends *mercator* by checking the IP identification field values in the IP protocol header of the returned ICMP response messages. *Ally* also compares the TTL values of the responses to narrow search space and tests for potential ICMP rate limits of the routers.

Given two IP addresses, *ally* successively sends a probe packet to each of the two address, and a third packet to the address that responds first. If the responses have IP identification values in sequence with a small difference in between, they are likely to be aliases and are classified as *alias*. IP addresses are classified as *non-alias* when identification values are distant or as *unknown* when one or both of the IP addresses do not respond to probe. The latter can happen as some ISPs configure their routers to ignore probes directed to them. There is also possibility that two responses from non-alias IP addresses will have close identification values. Thus, repeating the alias resolution test may reduce false positives.

DNS Based Method

DNS based method relies on the similarities in the host names of routers and works when an ISP uses a systematic naming convention in assigning IP addresses to router interfaces. By analyzing the structure of host names, one can decide whether IP addresses are alias or not. This method can be successful even if a router does not respond to probes directed to itself. However, the DNS based approach can be used only when a systematic naming scheme is used by the ISP and the naming template is present in the database. *Ally* uses this technique against unresponsive routers with the aid of *undns* tool which parses DNS names and reveals the geographic location of an IP address [5].

Graph Based Method

Graph based method builds a directed graph of linked IP addresses from traceroute outputs and observes patterns in the resulting graph to infer likely aliases and unlikely aliases [12]. Firstly, if two IP addresses directly precede a common successor IP address, then the two IP addresses are likely to be alias. The main assumption is that a router uses the IP address of the incoming interface as the source IP address for its response message. The second observation is that the addresses found in the same traceroute are unlikely to be alias. This approach mainly serves as a preprocessing step to reduce the number of probe pairs for an active probe based approach.

B. Limitations of Existing Approaches

Among the above approaches, probing based techniques (address based and IP identification based techniques) are the most effective methods for alias resolution. However, these approaches suffer from several disadvantages. First of all, active probing introduces traffic overhead in the network. Both approaches require one probe for each IP address in question. Additionally, IP identification based method may need repeated probes to increase the confidence of the decision.

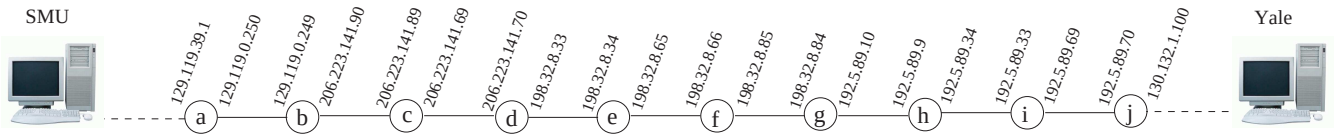


Fig. 1. Detecting IP aliases on a symmetric path segment.

In addition to introducing traffic overhead, probing based techniques suffer from lack of support by ISPs for probing. That is, most ISPs consider their network topology information as private information and hide it by configuring their routers not to respond probes directed to themselves. Furthermore, some operating systems assign random IP identification values to minimize the release of information as part of their protection mechanism against attackers. These situations make the IP identification based technique ineffective. For example, in our recent Internet measurements related study [2] we utilized *ally* to resolve IP aliases in a collected traceroute data set. At the end of alias resolution procedure, we realized that *ally* probes to 3122 unique IP addresses out of 7073 unique IP addresses did not get a response. Similarly, [1] reports on the limited success of probing based alias resolution on another real network topology.

III. Observations

In this section, we present several important observations about some Internet characteristics that helped us in the development of our alias resolution technique.

A. IP Address Assignment Practices

IP address space is a very scarce commodity and is used in a systematic way with a great care. The IP address assignment mechanism adheres to the guidelines presented in the Internet Registry IP Allocation Guidelines (RFC-2050). Basically, IP addresses belonging to a domain or an ISP network are divided into subnet ranges. Each subnet has a network address and each device, e.g., an end host or a router within the subnet, gets an IP address from the range of the network address given to the subnet. For example, if a subnet has a network address of 192.168.0.240/28, then the last 4 bits are used to identify the individual IP addresses to be assigned to the devices in this subnet. These four bits can identify at most 14 devices. The remaining two IP addresses, namely 192.168.0.240 and 192.168.0.255, have special meanings and are not typically used for host assignment.

For point-to-point links between two router interfaces, a /30 subnet address is defined and used for the IP address assignment to the interfaces. Recently, the use of /31 is made possible for point-to-point links (RFC-3021). For instance, the subnet of the point-to-point link in Fig. 2 can be either /30 or /31. For both cases, the IP address assignment to the interfaces of routers A and B are shown in the Fig. 2 where x's denote the network address.

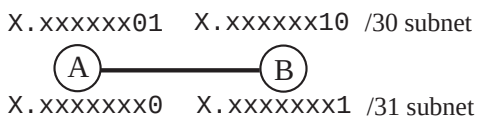


Fig. 2. IP address assignment for a point-to-point link

B. Inferring IP Aliases on Point-to-Point Links

The common IP address assignment practice for a point-to-point link results in $consecutive$ IP addresses on the point-to-point (See Fig. 2). Consecutiveness of IP addresses on a point-to-point link can be used to track potential path symmetry in path traces between two end points. That is, given path traces between two end points X and Y, one can look for two consecutive IP addresses IP_A and IP_B where IP_A is an IP address in the path trace from X-to-Y and IP_B is an IP address in the path trace from Y-to-X. Once such a match is observed, IP aliases can be inferred from the proper alignment of the path traces.

For instance, we use the above relation to infer IP aliases on a pair of path traces between the Yale University (Yale) and Southern Methodist University (SMU). Table I displays the traceroute output from SMU to Yale and the *reverse* of the traceroute output from Yale to SMU. Closely examining these two traces, we observe correlations between IP addresses in the 2^{nd} row till the 10^{th} row. Assuming point-to-point links with /30 or /31 subnets, we can construct the path segment corresponding to the traces as in Fig. 1. This arrangement then can be used to detect IP aliases: 129.119.0.249 and 206.223.141.90 are IP aliases representing router B, 206.223.141.89 and 206.223.141.69 are IP aliases representing router C, etc.

TABLE I
Traceroute results between SMU and Yale

#	SMU to Yale (Direct path)	Yale to SMU (Reverse path)
1	129.119.39.1	129.119.223.249
2	**129.119.0.249**	**129.119.0.250**
3	**206.223.141.89**	**206.223.141.90**
4	**206.223.141.70**	**206.223.141.69**
5	**198.32.8.34**	**198.32.8.33**
6	**198.32.8.66**	**198.32.8.65**
7	**198.32.8.84**	**198.32.8.85**
8	**192.5.89.9**	**192.5.89.10**
9	**192.5.89.33**	**192.5.89.34**
10	**192.5.89.70**	**192.5.89.69**
11	130.132.1.19	130.132.1.100
12	130.132.252.244	130.132.23.1

IV. Analytical Alias Resolution

In this section, we introduce a novel algorithm, Analytical Alias Resolver (AAR), that uses the above presented methodology to resolve IP aliases. Given two path traces between two nodes A and E (one path trace from A to E and the other from E to A), AAR uses an analytical approach to identify IP aliases. In the following, we first define Two-Path Alias Resolution Problem (TPARP) and then present our algorithm to solve this problem.

Definition ($\longleftrightarrow$): Let $\longleftrightarrow$ be a logic operator such that $\text{IP}_A \longleftrightarrow \text{IP}_B$ results in $true$ only if IP_A and IP_B are in a /30 subnet or /31 subnet as in Fig. 2.

Definition: Let $G = (V, E)$ be a connected graph where each vertex $v \in V$ has one or more interfaces $(i_1^v, i_2^v, ..., i_{degree(v)}^v)$. Let i_k^v denote the k^{th} interface of vertex v where each i_k^v has an *address*, $i_k^v.address$, that is unique in G. An edge $e \in E$ connects two adjacent vertices v_s and v_{s+1} by connecting interfaces $i_b^{v_s}$ of v_s and $i_a^{v_{s+1}}$ of v_{s+1} and satisfies the relation

$$i_b^{v_s}.address \longleftrightarrow i_a^{v_{s+1}}.address \tag{1}$$

Definition (PP): Let a *preferred path* $PP(v_i, v_j)$=$(V_{PP(v_i,v_j)}, E_{PP(v_i,v_j)})$ be a subgraph of G such that $V_{PP(v_i,v_j)}$=$\{v_i, v_{i+1}, v_{i+2}, \ldots, v_j\}$ represents the sequence of vertices between v_i and v_j, and $E_{PP(v_i,v_j)}$=$\{e_{(v_i,v_{i+1})}, e_{(v_{i+1},v_{i+2})}, \ldots, e_{(v_{j-1},v_j)}\}$ represents the sequence of edges in E connecting the vertices. An edge $e_{(v_k,v_{k+1})} \in E_{PP(v_i,v_j)}$ connects $v_k \in V_{PP(v_i,v_j)}$ and $v_{k+1} \in V_{PP(v_i,v_j)}$ via interfaces $i_b^{v_k}$ and $i_a^{v_{k+1}}$.

A path is a preferred path based on some application specific criteria, e.g., shortest path, minimum cost path, etc. Note that $PP(v_j, v_i)$ may or may not be equal to $PP(v_i, v_j)$.

Definition (Trace): $Trace(v_i, v_j)$ is a function of a preferred path $PP(v_i, v_j)$ where the trace visits each vertex $v_k \in V_{PP(v_i,v_j)}$ starting from v_i all the way to v_j and returns a list of interfaces (one for each vertex) as its output.

The interface representing a visited vertex in trace output is called an *incoming interface*. A trace function is said to arrive at a visited vertex from its incoming interface and the incoming interface for a trace function is identified based on some application specific criteria.

Definition (TPARP): Given two path traces $trace(v_i, v_j)$ and $trace(v_j, v_i)$, Two-Path Alias Resolution Problem (TPARP) is the problem of building a subgraph $\bar{G}$ of G such that $\bar{G}$=$(\bar{V}, \bar{E})$ where $\bar{V}$=$(V_{PP(v_i,v_j)} \cup V_{PP(v_j,v_i)})$ and $\bar{E}$=$(E_{PP(v_i,v_j)} \cup E_{PP(v_j,v_i)})$. If $i_a^{v_s} \in trace(v_i, v_j)$ and $i_b^{v_s} \in trace(v_j, v_i)$, then there should be one and only one $v_s \in \bar{V}$. And, for $v_s, v_t \in \bar{V}$, if $i_a^{v_s} \in trace(v_i, v_j)$ and $i_b^{v_t} \in trace(v_j, v_i)$ and $i_a^{v_s}.address \longleftrightarrow i_b^{v_t}.address$, then there should be one and only one $e_{(v_s,v_t)} \in \bar{E}$.

Analytical Alias Resolver (AAR)

Having defined the TPARP problem, we now present the Analytical Alias Resolver (AAR) algorithm as a solution to this problem. AAR benefits from previously mentioned IP address assignment mechanism for point-to-point links. Given a path pair (i.e., $trace(v_i, v_j)$ and $trace(v_j, v_i)$) between vertices v_i and v_j in V, AAR uses the relation in Equation 1 to identify symmetric path segments between the two paths. By symmetry, AAR locates the point-to-point links connecting vertices in the preferred paths $PP(v_i, v_j)$ and $PP(v_j, v_i)$. Identifying point-to-point links, AAR checks for the existence of interface address aliases and returns the found alias pairs. Each alias pair helps to remove a potential artificial vertex and an artificial link from the resulting network graph $\bar{G}$.

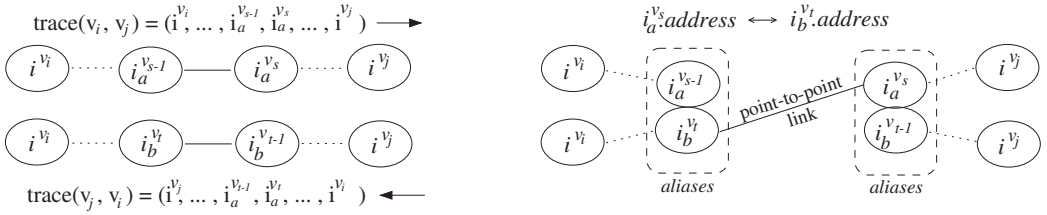
The algorithm in Fig. 3 solves the TPARP and produces the address aliases using aforementioned observations. The algorithm operates as follows. First, it populates $\bar{V}$ by including a vertex for each interface address and populates $\bar{E}$ by including an edge between two consecutive interfaces in the trace outputs $trace(v_i, v_j)$ and $trace(v_j, v_i)$. After this step, the graph $\bar{G}$ includes all connections between the vertices, but potentially has redundant vertices and edges as well. Then, alias resolution phase minimizes the graph by finding IP aliases and removing redundant vertices and edges. In each iteration of the alias resolution phase, the algorithm considers an interface $i_a^{v_s}$ of $trace(v_i, v_j)$ and compares it with the interfaces in $trace(v_j, v_i)$. When a match is found based on Equation 1, a point-to-point link is identified between the matching vertices. Then, vertices in $\bar{V}$ representing the matched interfaces are unified by $v_s = v_{t-1}$ and $v_t = v_{s-1}$. This also merges the corresponding edges in $\bar{E}$. Finally, as a byproduct of the algorithm, alias pairs are recorded in a set called $Alias$.

INPUT: $trace(v_i, v_j)$ and $trace(v_j, v_i)$
OUTPUT: $\bar{G} = \{\bar{V}, \bar{E}\}$; *Alias* = $\bigcup\ (i_a^{v_k}, i_b^{v_l})$
INITIALIZE: $\bar{V}$ = $\emptyset$; $\bar{E}$ = $\emptyset$; *Alias* = $\emptyset$

/* populate $\bar{V}$ and $\bar{E}$ */
for ($\forall\ i^{v_k} \in (trace(v_i, v_j)$ or $trace(v_j, v_i))$)
  $\bar{V} \leftarrow \bar{V} \cup v_k$
  if ($\exists\ i^{v_{k-1}}$) **then** $\bar{E} \leftarrow \bar{E} \cup e(v_k, v_{k-1})$

/* alias resolution phase */
for ($\forall\ i_a^{v_s} |\ i_a^{v_s} \in trace(v_i, v_j)$)
  if ($\exists\ i_b^{v_t} \in trace(v_j, v_i)$ |
      $(i_b^{v_t}.address \longleftrightarrow i_a^{v_s}.address)$) **then**
    if ($\exists\ v_{t-1} \in trace(v_j, v_i)$) **then**
      $v_s = v_{t-1}$ /* merge into one vertex */
      Alias $\leftarrow$ *Alias* $\cup\ (i_a^{v_s}, i_b^{v_{t-1}})$
    if ($\exists\ v_{s-1} \in trace(v_i, v_j)$) **then**
      $v_t = v_{s-1}$ /* merge into one vertex */
      Alias $\leftarrow$ *Alias* $\cup\ (i_a^{v_{s-1}}, i_b^{v_t})$

Fig. 3. AAR: An approach to solve TPARP problem.



a) Symmetric segments of traces

b) Merging aliases

Fig. 4. Reducing graph $\bar{G}$ by merging aliases

Fig. 4 demonstrates an example case to identify IP address aliases on a point-to-point link. Interfaces $i_a^{v_s-1}$ and $i_a^{v_s}$ are on the forward path trace and interfaces $i_b^{v_t-1}$ and $i_b^{v_t}$ are on the reverse path trace between two end points v_i and v_j. Since $i_a^{v_s}.address \longleftrightarrow i_b^{v_t}.address$, according to Equation 1, there is an edge connecting the two vertices v_s and v_t. Due to the edge between v_s and v_t, $i_a^{v_s}$ and $i_b^{v_t-1}$ are aliases and v_s and v_{t-1} are merged. Similarly, $i_a^{v_s-1}$ and $i_b^{v_t}$ are aliases and v_{s-1} and v_t are merged.

V. Evaluations

In this section, we present our experimental evaluations of the Analytical Alias Resolver (AAR) in two phases. In the first part, we compare AAR with the existing IP alias resolution approaches. For our evaluations, we use traceroute tool to collect a set of end-to-end path pairs on the Active Measurement Project (AMP) [4] infrastructure of NLANR. Our data set includes 351 pairs of path traces among 27 vantage points and includes a total of 503 unique IP addresses.

In the second part, we use a data set obtained from the Abilene Internet2 backbone network to assess the accuracy and the effectiveness of AAR. Here, we compare the set of alias pairs that AAR returns with the set of actual alias pairs that we obtain from an interface-level map of the Abilene network topology.

A. Comparison to Existing Approaches

In this subsection, we compare our approach with the existing probing approaches, namely address based approach, IP identification based approach, and the current state-of-the-art tool *ally*, which combines both approaches.

Qualitatively speaking, we can compare the above approaches based on their overhead and effectiveness as follows. AAR is a passive approach where it does not require active probe messages for resolving IP aliases. However, AAR assumes availability of path traces to perform alias resolution. On the other hand, probing based approaches require active probing to resolve IP aliases. Active probing incurs additional measurement traffic overhead into the network. In addition, it strongly depends on routers to respond probes directed to themselves. But, some of the ISPs configure their routers not to respond probes directed to themselves due to performance, privacy, and security reasons. This consequently reduces the effectiveness of probing based alias resolution approaches.

On the quantitative site, we use our data set to compare the number of alias pairs returned by different approaches. First, we use AAR algorithm on 351 path pairs to identify potential alias pairs and use the probing based approaches to see if they agree with AAR on this set of potential alias pairs. Note that without having the underlying network topology information, we cannot really be sure if a pair of IP addresses are alias. Our motivation here is to see the level of agreement and disagreement among the alternative approaches. Table II presents the number of alias pairs identified by each approach. AAR identifies 180 pairs of aliases in the data set. Being a passive approach, results in the form of 'Not Alias' or 'No Decision' are not applicable for AAR. The comparison of the second and the third rows suggests that IP identification based approach is more effective than address based approach. The last row indicates that *ally* and AAR agree on 77 cases, disagree on 5 cases.

In the next step, we run *ally* on all IP address pairs ($\binom{502}{2}$ = 126,253 pairs) to observe the number of address pairs that *ally* identifies as alias but AAR does not. As shown in Fig. 5, *ally* identifies a total of 100 alias pairs out of which 77 match with alias pairs returned by AAR. AAR does not report any result for the remaining 23 pairs. Assuming that the 23 alias pairs returned by *ally* are accurate, the reasons for the failure of AAR in returning these alias pairs may be that either (1) the two IP addresses in an alias pair may not belong to a path pair between two vantage points (i.e., the two IP addresses may appear on two independent traceroutes) or (2) the relevant routers are connected by a medium other than a point-to-point link.

TABLE II
Identified IP Aliases

Method	Alias	Not Alias	No Decision
AAR	180	n/a	n/a
Source-Address (SA)	42	40	98
IP Identification (IPI)	72	6	102
Ally (uses SA&IPI)	77	5	98

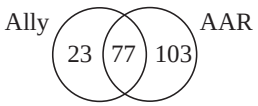


Fig. 5. Set comparison of IP aliases found by AAR and *ally*.

Fig. 6. Abilene Backbone (from http://abilene.internet2.edu)

B. Verification of AAR

In this subsection, we experiment with a small data set to measure effectiveness of AAR. We use the interface level topology map of the Abilene network, shown in Fig. 6, to verify the correctness and completeness of alias pairs returned by AAR. We carefully choose 11 AMP vantage points such that each router in the Abilene network appears as an entry or an exit point in at least one of the paths among these vantage points. After collecting path traces among 11 vantage points, we obtain path segments where each router on the path belongs to Abilene network (i.e., IP addresses are in 198.32.8.0/22). Hence, our data set includes partial path traces belonging to Abilene Internet2 backbone network.

After preparing the data set, we run AAR and *ally* on path pairs to identify the number of alias pairs corresponding to Abilene backbone routers. Table III presents the results where the first column indicates the geographical location of the routers, the rest of the columns give the number of alias pairs as obtained from the real topology map; from the AAR approach; and from *ally* approach, respectively. According to the results, *ally* remains completely ineffective for this experiment. A careful examination of *ally* outputs shows that Abilene routers do not respond to *ally* probes causing it to return 'No Decision' for all the cases.

TABLE III
NUMBER OF IP ALIASES IN ABILENE BACKBONE

Node Location	From the Map	From AAR	From *ally*
Atlanta	3	1	0
Chicago	1	1	0
Denver	3	3	0
Houston	3	1	0
Indianapolis	3	3	0
Los Angeles	1	1	0
Kansas City	3	3	0
New York	1	1	0
Seattle	1	0	0
Sunnyvale	3	3	0
Washington	1	1	0

In this experiment, AAR successfully identifies 18 alias pairs out of 23 aliases that exist in the network without any false positives. AAR fails to detect the IP address aliases at the Seattle router because this router does not appear on any path as an internal router. In other words, the Seattle router does not provide transit between Sunnyvale and Denver routers. Similarly, AAR fails to detect alias pairs at Atlanta and Houston routers since Atlanta-Indianapolis and Houston-Kansas City links do not appear in any of the collected paths. Based on these observations, we conclude that due to unicast routing preferences, our data set does not include necessary information to detect related alias pairs on these routers.

VI. CONCLUSION

In this paper we have focused on IP alias resolution problem. First, we have emphasized the importance of alias resolution for Internet measurement studies. Then, we have presented the existing probing based approaches and discussed their limitations. Next, we have introduced a graph theoretic formulation of a simplified version of the alias resolution problem and have presented an algorithm, AAR, to solve this problem. Finally, using a collected Internet topology data, we have shown the superiority of our approach over the existing alternatives with respect to their overhead and effectiveness.

ACKNOWLEDGMENTS

The authors would like to thank Tony McGregor for providing the AMP data.

REFERENCES

[1] R. Teixeira, K. Marzullo, S. Savage, and G. Voelker, "In search of path diversity in ISP networks," in *Proceedings of the USENIX/ACM Internet Measurement Conference*, (Miami, FL, USA), October 2003.
[2] S. Bilir, K. Sarac, and T. Korkmaz, "End to end intersection characteristics of Internet paths and trees," in *IEEE International Conference on Network Protocols (ICNP)*, (Boston, MA, USA), November 2005.
[3] D. McRobb, K. Claffy, and T. Monk, *Skitter: CAIDA's macroscopic Internet topology discovery and tracking tool*, 1999. Available from http://www.caida.org/tools/skitter/.
[4] A. McGregor, H.-W. Braun, and J. Brown, "The NLANR network analysis infrastructure," *IEEE Communications Magazine*, vol. 38, pp. 122–128, May 2000.
[5] N. Spring, D. Wetherall, and T. Anderson, "Scriptroute: A public internet measurement facility," in *North American Network Operator's Group*, October 2002.
[6] V. Paxson, J. Mahdavi, A. Adams, and M. Mathis, "An architecture for large-scale Internet measurement," *IEEE Communications*, vol. 36, pp. 48–54, August 1998.
[7] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson, "Measuring ISP topologies using rocketfuel," *IEEE/ACM Transactions on Networking*, vol. 12, pp. 2–16, February 2004.
[8] A. Lakhina, J. Byers, M. Crovella, and P. Xie, "Sampling biases in IP topology measurements," in *Proceedings of IEEE INFOCOM*, (San Francisco, CA, USA), March/April 2003.
[9] D. Alderson, L. Li, W. Willinger, and J. Doyle, "Understanding Internet Topology: princIPles, models, and validation," vol. 13, December 2005.
[10] J. Pansiot and D. Grad, "On routes and multicast trees in the Internet," *ACM Computer Communication Review*, vol. 28, pp. 41–50, January 1998.
[11] R. Govindan and H. Tangmunarunkit, "Heuristics for Internet map discovery," in *IEEE INFOCOM*, March 2000.
[12] M. R. Neil Spring, Mira Dontcheva and D. Wetherall, "How to resolve IP Aliases," tech. rep., University of Washington, May 2004.