Resolving Anonymous Routers in Internet Topology Measurement Studies

Mehmet H. Gunes and Kamil Sarac
Department of Computer Science, University of Texas at Dallas
Email: {mgunes, ksarac}@utdallas.edu

***Abstract*—Internet measurement studies utilize traceroute to collect path traces from the Internet. A router that does not respond to a traceroute query is referred to as an anonymous router and is represented by a '*' in the traceroute output. Anonymous router resolution refers to the task of identifying the occurrences of '*'s that belong to the same router in the underlying network. This task is an important step in building traceroute-based topology maps and obtaining an optimum solution is shown to be NP-complete. In this paper, we use a novel technique from graph data mining field to build an efficient solution. The results of our experiments on both synthetic and genuine topologies show a significant improvement in accuracy and effectiveness over the existing approaches.**

I. Introduction

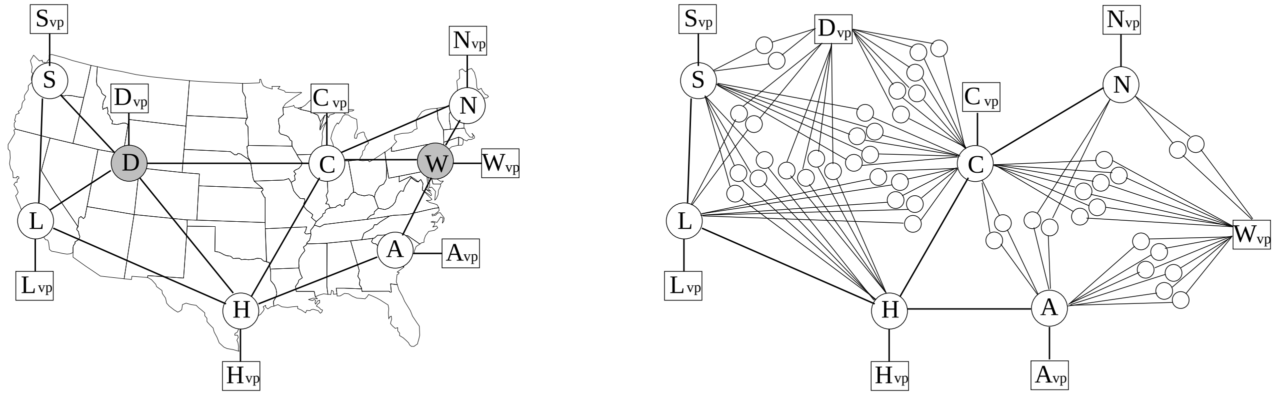
Router level Internet topology maps are useful in various contexts including analyzing the topological characteristics of the Internet and designing topology generators that can produce Internet-like synthetic network topologies. Due to privacy and security reasons, Internet Service Providers (ISPs) keep their router level topology information confidential and do not share it with others. This policy introduces a practical challenge for the research community and requires them to use other means to collect Internet topology data. Most measurement studies utilize *traceroute* to collect a large number of path traces from topologically diverse set of vantage points and use this data in their studies.

After collecting path traces, the information needs to be processed to build the corresponding topology map. This task involves several steps including (1) verifying the correctness of the collected paths, (2) resolving IP addresses belonging to the same router, (3) inferring IP addresses that are connected over the same subnet, and (4) resolving anonymous routers that are represented by '*'s in traceroute outputs. The accuracy and the completeness of these tasks may significantly affect the representativeness of the resulting topology maps [9], [16]. Hence, topology measurement studies should handle these tasks to obtain a representative topology map.

The first task in topology construction is concerned with the fact that certain traffic engineering practices may cause traceroute to return IP addresses that do not correspond to a real end-to-end path in the network. A recent study by Augustin et al. [1] proposes *Paris traceroute* tool to address this problem. The second task, alias resolution, is due to the fact that routers have multiple IP addresses that may appear in different path traces. Several tools exist to resolve alias IP addresses [7], [8], [15]. The third task, subnet resolution, identifies subnet relations among IP addresses and reveals connectivity that is not observed in collected path traces [10]. The last task, anonymous router resolution, emerges due to the fact that not all routers respond to traceroute probes during topology collection and is the main focus of this paper.

An anonymous router is represented by a '*' in a traceroute output and the same router may appear as a '*' in multiple different traceroute outputs. Depending on the number of anonymous routers and the topology collection scenario, the collected set of path traces may include a large number of occurrences of '*'s. Consider the National Lambda-Railway network backbone presented in Figure 1-a. Assume that the routers D and W are configured to be anonymous and path traces are collected among all vantage points, represented as squares in the figure. Under these assumptions, the topology that is constructed from the collected path traces corresponds to the one shown in Figure 1-b (with no resolution). This example demonstrates that even a small number of anonymous routers may significantly distort the constructed network topology and presents the importance of anonymous router resolution task. Additionally, the iPlane-collected genuine Internet topology that we use in our evaluations in Section V-B includes over 9M anonymous nodes (1M after the initial pruning) along with 230K known routers. The mere volume of anonymous nodes in the collected data set introduces additional challenges in building an efficient solution.

The anonymous router resolution problem has not received much attention from the research community and most topology measurement studies tend to ignore this problem or use simple mechanisms to work around it (see Section II for details). To the best of our knowledge, there are only two studies that attack this problem [11], [17]. In [17], Yao et al. propose a graph minimization approach with a computational complexity of $O(n^5)$ where n is the number of anonymous nodes in the topology. In [11], Jin et al. proposes an ISOMAP-based dimensionality reduction technique with a computational complexity of $O(n^3)$. Considering the volume of anonymous nodes in topology mapping studies (e.g., 10^6 in our iPlane data set), neither of the solutions is practical as they incur high computational complexity (e.g., 10^{30} or 10^{18} operations, respectively). Jin et al. also propose a simple neighbor matching heuristic as a more practical alternative solution. However, as the authors mention in their paper, this approach has accuracy problems as it may yield a high rate of false positives and false negatives.



(a) National LambdaRail Backbone Network

(b) Induced Topology

Fig. 1. Sample network

In this paper, we develop and utilize a *graph based induction* technique to resolve anonymous routers in traceroute based topology mapping studies. Graph based induction is a technique to obtain information from a graph in data mining field [13]. In our work, we define a new induction approach from the practical context of the anonymous router resolution problem and develop an efficient implementation. We first analyze the nature of anonymous routers and identify different types of anonymity. We observe that anonymity due to ICMP rate limiting is an important case that needs to be handled. However, neither of the previous studies considered this type of anonymity in their algorithms. Then, we visually examine a number of topology maps that are constructed from traceroute data with anonymous routers. From this visual study, we identify a number of graph structures that are formed among anonymous nodes and their known router neighbors. We then develop efficient algorithms to detect these structures in the graph and reduce the anonymous nodes (i.e., the occurrences of '*'s) into their corresponding anonymous routers[1].

In our evaluations on synthetic topologies, we observe that graph based induction approach has better resolution than previously proposed neighbor matching approach. Due to their high complexity, we did not compare our solution with graph minimization approach of [17] and dimensionality reduction approach of [11]. Our approach produces topologies that are closer to the actual topology in terms of different topological characteristics. In addition, we demonstrate the practicality of our approach on a large genuine Internet topology data (iPlane data [12]) with 18M path traces, 9M anonymous and 230K known nodes. Previous studies work with much smaller topology data (e.g., [17] works with up to 50 node synthetic topologies) to resolve anonymous nodes.

The rest of this paper is organized as follows. Section II presents related work and Section III formally defines the anonymous router resolution problem. Section IV presents our graph based induction approach. Section V presents our experimental evaluations. Section VI concludes the paper.

II. Related Work

A. Anonymous Router Resolution

Anonymous router resolution is an inherent problem in traceroute based topology mapping studies. Most of the early work in the area ignored this problem or used simple heuristics to work around it [2], [3], [4]. In [4], authors avoid the problem by stopping a trace toward a destination on encountering an anonymous router on the path. In [3], authors handle anonymous routers by replacing them either (1) with arcs (to connect the known routers at two ends) or (2) with unique identifiers to treat them as separate nodes. Finally, in [2], authors use a sandwich approach to merge a chain of anonymous nodes between the same pair of known nodes with each other. These approaches cause either loss of potentially useful connectivity information (as in [4]), or inaccuracies in the resulting topology maps (as in [3]), or limited resolution in the resulting topology maps (as in [2]).

Yao et al. formulate the anonymous router resolution problem as an optimization problem [17]. Their goal is to build a minimum size topology by combining anonymous nodes with each other under two conditions: (1) trace preservation condition and (2) distance preservation condition. The first condition corresponds to the trace preservation condition that we define in the next section and we briefly discuss the second condition below. They prove that the optimum topology inference under these conditions is NP-complete and then propose a heuristic to minimize the constructed topology by identifying anonymous nodes that, when merged, satisfy the two conditions. The main limitation of this approach is its high complexity, i.e. O(n^5) where n is the number of anonymous nodes, that significantly limits its practicality in real life scenarios. A second limitation is its use of the distance preservation condition stating that the anonymous router resolution process should not reduce the length of a shortest path between any two nodes in the resulting topology map. This condition is not necessarily accurate as inter domain routes may not always be the shortest routes in the Internet.

Jin et al. propose two heuristics to address the problem in [11]. First one is an ISOMAP based dimensionality reduc-

[1] In this paper, we use the term *anonymous node* to refer to a '*' in a traceroute output and *anonymous router* to refer to the actual router that is represented by this anonymous node (i.e., by this '*') in the traceroute output.

tion approach that uses link delays or node connectivity as attributes in the dimensionality reduction process. The main limitation of this approach is its high complexity, i.e., O(n^3) where n is the size of the topology. In addition, their link delay based approach suffers from practicality problems as they ignore the difficulty of estimating individual link delays from round trip delays in path traces [6]. The authors also propose a simple neighbor matching heuristic with a smaller time complexity, i.e., O(n^2). As the authors mention in their paper, this approach suffers from accuracy problems as it may introduce a high rate of false positives and false negatives.

In this paper, we propose a practical approach using graph data mining techniques. We analyze the graph structures generated by anonymous routers and introduce a graph based induction technique that considers trace preservation condition to resolve anonymous routers. Our approach has a small time complexity, resolves any type of anonymous routers, and is effective in large topologies.

B. Mining Graph Data

Graph data mining techniques are utilized in many application domains to extract useful information from the graph representation of large data sets [5]. Among various graph mining techniques, graph-based induction technique shows similarity to our problem. Graph-based induction is a technique to find frequent substructures, a common problem in biological and chemical networks, [13]. It extracts typical patterns by stepwise identification of recurring node pairs and minimizes the graph size by replacing identified patterns with a node. In our work, we propose a similar induction technique to identify subgraphs of some common structures within the topology graph and use them to resolve anonymous routers.

III. Anonymous Router Resolution Problem

Anonymous router resolution problem emerges due to the fact that not all routers respond with ICMP error messages to traceroute queries all the time. In our topology measurement studies, we observe five different scenarios that cause routers to stay anonymous as follows:

- **Type 1:** A router may be configured to ignore traceroute queries causing it to be anonymous in all trace outputs.
- **Type 2:** A router may apply ICMP rate limiting and stay anonymous if the rate of the incoming queries exceed the preset limit.
- **Type 3:** A router may be configured to ignore ICMP messages when it is congested but may respond to traceroute queries when it is not congested.
- **Type 4:** A border router may be configured to filter all outgoing ICMP responses coming from routers within its administrative domain. This causes all the routers in its domain to be anonymous.
- **Type 5:** A router may have a private (publicly unroutable) IP address. Such private IP addresses can not guarantee node uniqueness as they may be used by multiple routers in different networks. Hence, when a response with a private IP address reaches the vantage point, the ICMP sending router should be handled as an anonymous router.

We now introduce a number of definitions and conditions that will help us formally define the problem. The notations introduced in this section will be used in the development of the algorithms in the next section.

Definition (Router-Level Graph): Let $G = (V, E)$ be a router level network graph where V represents the set of vertices (i.e., routers and end-hosts) and E represents the set of edges (i.e., communication links) connecting the vertices in V. Each vertex $v \in V$ has one or more interfaces i_e^v and each interface i_e has a unique identifier $i_e.address$ that corresponds to a globally unique IP address. □

For the ease of presentation, we use i_e to represent both an interface and its corresponding identifier, i.e., its IP address.

Definition (Preferred Path): A *preferred path* $PP(v_i, v_j)$=($V_{PP(v_i,v_j)}, E_{PP(v_i,v_j)}$) is a subgraph of G where $V_{PP(v_i,v_j)}$=$\{v_i, v_{i+1}, v_{i+2}, \ \ldots, \ v_j\}$ represents the sequence of the 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 the edges in E connecting the vertices in $V_{PP(v_i,v_j)}$. □

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 not be equal to $PP(v_i, v_j)$.

Definition (Trace): A 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 *nodes* representing the interfaces (one for each vertex) as its output, i.e., $trace(v_i, v_j) = (i_x^{v_i}, \ldots, i_e^{v_p}, i_f^{v_r}, \ldots, i_y^{v_j})$. Note that a $trace$ may visit anonymous routers on the preferred path $PP(v_i, v_j)$. If a router $v_p \in PP(v_i, v_j)$ is an anonymous router, the $trace$ output will have a *anonymous node*, $*_e^{v_p}$, instead of a *known node*, $i_e^{v_p}$. □

Definition (Alias Set): An alias set is a set of known (e.g., $i_e^{v_p}$) and anonymous (e.g., $*_f^{v_p}$) nodes that belong to a router v_p. □

The alias set of known nodes is handled by IP alias resolution process [7], [8], [15] and is not focus of this paper.

Definition (Substring): A substring is a continuous segment of a trace output and is denoted as $s_{(a,b,l)} = (a, v_1, v_2, ..., v_l, b)$ where a and b are known nodes and each v_i is a known (e.g., i_e) or an anonymous (e.g., $*_f$) node. □

Definition (*-Substring): A *-substring $s^*_{(a,b,l)}$ is a continuous segment of length $l + 2$ of a trace output that starts and ends with known nodes a and b, and all intermediate nodes are anonymous nodes. Given a path trace $trace(v_i, v_j) = (i^{v_i}, \ldots, i_a, *_1, *_2, \ldots, *_l, i_b, \ldots, i^{v_j})$, a *-substring is $s^*_{(i_a,i_b,l)} = (i_a, *_1, *_2, \ldots, *_l, i_b)$. □

Note that a given trace may have zero or more *-substrings.

Definition (Trace Preservation Condition): If $(*_e, *_f) \in \ trace(v_i, v_j)$, then $*_e$ and $*_f$ cannot be in the same alias set, i.e., they cannot belong to the same anonymous router, for any $trace(v_i, v_j)$ in the data set. □

Trace preservation condition serves as an accuracy condition during topology construction. This condition arises from the fact that there should be no routing loops in collected path traces. Path traces with loops should be filtered during trace verification task prior to the anonymous router resolution.

Definition (Mergeable): Two anonymous nodes $*_e$ and $*_f$ are mergeable, $mergeable(*_e, *_f)$, if they (or any other node in their alias sets) do not appear in the same path trace. □

Node mergeability can be found in $O(1)$ by keeping an $O(n^2)$ data structure or in $O(log(m))$ by keeping an $O(n.m)$ data structure where m is the average number of non-mergable nodes per node (m was below 5 in our experiments). Note that such a data-structure is required for previous algorithms as well.

Definition (Anonymous Router Resolution Problem): Given path traces, $\bigcup trace(v_i, v_j)$, and IP address alias sets of *known* nodes from a network graph $G = (V, E)$, anonymous router resolution problem is to build a graph $\bar{G} = (\bar{V}, \bar{E})$

- $\bar{V} = \bigcup V_{PP(v_i,v_j)}$ and $\bar{E} = \bigcup E_{PP(v_i,v_j)}$.
- If $*_e^{v_p} \in trace(v_k, v_l)$ and $*_f^{v_p} \in trace(v_m, v_n)$, then there should be one and only one anonymous node $*^{v_p} \in \bar{V}$ corresponding to anonymous router $v_p \in V$.
- If $*_e^{v_p} \in trace(v_k, v_l)$ and $i_f^{v_p} \in trace(v_m, v_n)$, then there should be one and only one node $i^{v_p} \in \bar{V}$ corresponding to the router $v_p \in V$.
- $\forall *_e^{v_p}$ and $\forall i_e^{v_p}$ of a node $v_p \in \bar{V}$ and a node $v_r \in \bar{V}$, there should be at most one $e_{(v_p,v_r)} \in \bar{E}$. Note that parallel links between *known* nodes are preserved.

In the above problem definition, we require that the IP alias resolution process to be performed beforehand such that the following condition is already satisfied:

- If $i_e^{v_r} \in trace(v_k, v_l)$ and $i_f^{v_r} \in trace(v_m, v_n)$, then there should be one and only one $v_r \in \bar{V}$.

IV. Graph Based Induction Approach

In this section, we present a graph based induction technique to resolve anonymous routers that introduce a large number of artificial nodes in traceroute-based topology maps. In this study, we first identify the anonymity types that are presented in Section III. We then formulate a number of graph structures that can be found in traceroute-based topologies collected from the Internet. Next, we visually examine a number of sample topology maps to verify the occurrences of these types of structures. These structures are shown in Figure 1-b and Figure 4-b,-d,-f, and the corresponding connectivity relations in the underlying actual network are shown in Figure 1-a and Figure 4-a,-c,-e, respectively.

Based on this formulation, we introduce a graph based induction technique where we search for structures similar to the identified ones in a traceroute-collected topology data and then reduce the occurrences of anonymous nodes in them into their corresponding routers. In this process, we first start with a router-level topology graph $\bar{G} = (\bar{V}, \bar{E})$ that is constructed from a set of traceroute-collected path traces. We assume that IP alias resolution process is completed beforehand. We then apply graph based induction to resolve anonymous routers based on the structures as shown in Figure 4. In the rest of this section, we present each of the identified structures, their underlying topology, and how we use graph based induction to reduce them to their corresponding actual routers. Note that multiple network topologies with anonymous routers can result in the same observed topology. Yao et al. analyzes this issue and proposes to use the minimal topology as the underlying topology for an observed network topology [17]. Similarly, we assume that the minimal topology is the underlying topology for observed structures.

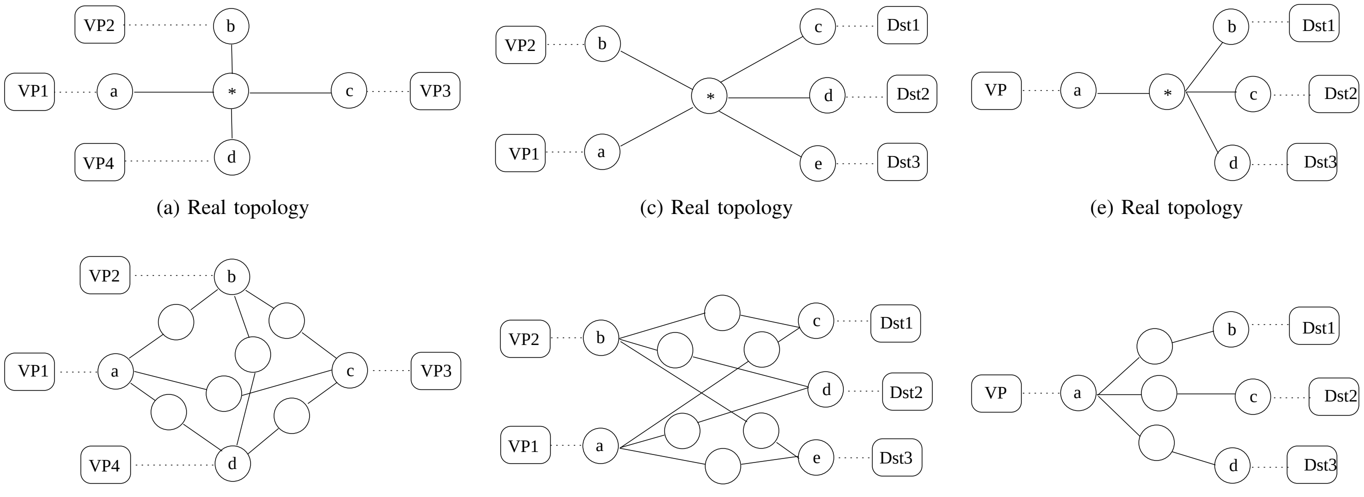
Structure 1 (Parallel/Symmetric *-substrings): One common pattern that we observe in traceroute-based topology maps is the occurrences of same length *-substrings with the same known nodes at the ends of each *-substrings. While collecting path traces from a vantage point, an anonymous router may appear as '*' in multiple path traces resulting in multiple parallel *-substrings between the same known nodes. As an example in Figure 1-a, traceroute queries from S_{vp} to H_{vp} and to A_{vp} may return path traces including *-substrings as $(S, *_1, H)$ and $(S, *_2, H)$ respectively. Similarly, traceroutes from H_{vp} and A_{vp} to S_{vp} may result in additional *-substrings as $(H, *_3, S)$ and $(H, *_4, S)$. This may then result in four parallel *-substrings between S and H in the resulting topology map as shown in Figure 1-b. Note that the above example uses *-substrings of length three, i.e., *-substrings that include only one anonymous router. A similar pattern can be observed for *-substrings of larger lengths.

Resolution of anonymous routers in this type of structures requires detection of similar *-substrings (i.e., same length *-substrings with the same known nodes at their end points). The algorithm in Figure 2 provides a graph search module to implement this task. In this algorithm, we extract all *-substrings from the path traces and identify same length *-substrings with the same known end nodes to merge them with each other.

The above structure is the most frequently seen structure in traceroute based topology maps and the proposed reduction removes the highest number of anonymous nodes from the constructed graph. Note that due to the nature of the observed structure, the resulting topology map will include anonymous nodes that will have a degree of two irrespective of the actual degrees of these routers in the underlying topology. That is, this step will reduce the number of artificial nodes in the

```
Let $\bar{G} = (\bar{V}, \bar{E})$; $\bar{V} \leftarrow \phi$; $\bar{E} \leftarrow \phi$; $S \leftarrow \phi$;
for (each $trace$ in $\bigcup trace(v_i, v_j)$)
  $\bar{V} \leftarrow \bar{V} \cup \{a, b\}$; $\bar{E} \leftarrow \bar{E} \cup \{e_{(a,b)}\}$ $\forall s_{(a,b,0)} \in trace$
  for (each $s^*_{(a,b,l)} = (a, *_1, *_2, ..., *_l, b) \in trace$)
    if ($\neg\exists$ $s^*_{(a,b,l)} \in S$)
      $S \leftarrow S \cup s^*_{(a,b,l)}$
      $\bar{V} \leftarrow \bar{V} \cup \{*_i\}$ for $1 \leq i < l$
      $\bar{E} \leftarrow \bar{E} \cup \{e_{(a,*_1)}, e_{(*_l,b)}, e_{(*_i,*_{i+1})}\}$ for $1 \leq i < l$
```

Fig. 2. Alg.1: Finding parallel and symmetric *-substrings



(a) Real topology

(c) Real topology

(e) Real topology

(b) Observed clique-like topology

(d) Observed complete bipartite-like topology

(f) Observed star topology

Fig. 4. Structures (genuine and observed)

```
INPUT: $\bar{G} = (\bar{V}, \bar{E})$ and $S$ from Alg.1 in Figure 2
for (each $a \in trace(v_i, v_j) \in \bigcup trace(v_i, v_j)$)
  for (each $s^*_{(a,b,l)} = (a, *_1, *_2, ..., *_l, b) \in S$)
    if ($\exists\ s_{(a,b,l)} = (a, p_1, p_2, ..., p_l, b) \in trace(v_i, v_j)$)
      set $*_1 \leftarrow p_1$, $*_2 \leftarrow p_2$, ..., $*_l \leftarrow p_l$ in $\bar{G}$
      $S \leftarrow S - s^*_{(a,b,l)}$
```

Fig. 3. Alg.2: Finding anonymous nodes due to type 2 and 3 routers

resulting graph but may not help resolve anonymous routers that appear on *-substrings with different known end nodes.

Another related common pattern is caused by routers that apply ICMP rate limiting or that stay anonymous when congested (type 2 & 3 anonymity). Such a router may appear as a known node in some path traces and may appear as a '*' in some other path traces. For instance, an ICMP rate limiting router c may cause occurrences of related substrings in the form of $(\ldots, a, c, *_3, b, \ldots)$ and $(\ldots, a, *_1, *_2, b, \ldots)$ in different traceroute outputs. In this case, we resolve $*_1$ to c and $*_2$ to $*_3$. We use the algorithm in Figure 3 to resolve anonymous nodes in such cases.

We now analyze the complexity of both algorithms. In a naive implementation of the algorithms, we would compare *-substrings with each other requiring O(n_s^2) comparisons where n_s is the number of *-substrings. However, an efficient implementation will reduce the time complexity. While reading the traces, each $s^*_{(a,b,l)}$ is stored based on the known end nodes a and b in a sorted table. Subsequently read $s^*_{(c,d,k)}$'s are then compared to the ones with the same starting known nodes c and d in the table. This then results in a complexity of O($n_s.log(n_s)$). Processing *-substrings in this manner significantly reduces the run time for large data sets. Moreover, an efficient implementation of Alg.2 in Figure 3 takes O($n_t.log(n_s)$) time where n_t is the number of path traces. We use the table built in the previous step and make a second run of path traces looking for known nodes that appear as end points in the table.

Structure 2 (Clique): The second structure that is formed between an anonymous router $*^v$ and its known neighbors $\{n_1, n_2, \ldots, n_k\}$ can be seen as a complete subgraph, i.e., K_k clique, among the known neighbors of $*^v$. This type of structure occurs when the topology map before anonymous router resolution process includes all *-substrings $(n_i, *^v, n_j)$ where $i, j \in [1, k], i \neq j$. Figure 4-b presents an example structure of this type where the data set includes *-substrings among all known neighbors of the anonymous node. This structure is referred to as a 4-clique and corresponds to the topology presented in Figure 4-a. To resolve anonymous nodes in clique structures, we first create a new graph $G^* = (V^*, E^*)$ to identify this type of structures. For this, for each *-substring of type $(a, *_e, b)$, we have $V^* \leftarrow V^* \cup \{a, b\}$ and $E^* \leftarrow E^* \cup \{e_{(a,b)}\}$ in G^*. Finally, we process G^* to find cliques and resolve anonymous nodes in G. In general, this type of structures appear in data set that are collected among all vantage points, i.e., in (n,n)-traceroute data.

In this process, we prefer to identify the cliques starting from the largest one to merge the corresponding anonymous nodes. However, given a graph, finding the maximum complete subgraph is an NP-complete problem. Due to the practical context of the problem, we are not interested in maximal cliques. That is, in order to observe a clique, the data set should include path traces through all pairs of neighboring routers of an anonymous router. Hence, instead of searching maximal cliques, we first identify 4-cliques and grow them by adding nodes that are connected to at least three of the nodes in the existing substructure. This approach also helps in tolerating few missing links in large cliques. After processing all 4-cliques and larger clique-like structures, we process all 3-cliques among the remaining nodes and resolve the corresponding anonymous routers. Figure 5 presents the corresponding algorithm.

```
INPUT: $S$ from Alg.2 in Figure 3
FUNCTION Θ: INPUT: $I = \{a, b, ...\} \subset V^*$
            OUTPUT: $\{*_e \mid (a, *_e, b) \in S, \forall\{a, b\} \in I\}$
Let $G^* = (V^*, E^*)$; $V^* \leftarrow \phi$; $E^* \leftarrow \phi$
for (each $s^*_{(a,b,1)} \in S$)
  $V^* \leftarrow V^* \cup \{a, b\}$; $E^* \leftarrow E^* \cup \{e_{(a,b)}\}$

for (each pair of edges$\{e_{(a,b)}, e_{(c,d)}\} \in E^*$)//$K_4$ or larger
  if ($\exists \{e_{(a,c)}, e_{(a,d)}, e_{(b,c)}, e_{(b,d)}\} \in E^*$)
    $C \leftarrow \{a, b, c, d\}$
    if ($mergeable(\Theta(C))$)
      while ($\exists \{e_{(f,x)}, e_{(f,y)}, e_{(f,z)}\} \in E^* | \{x, y, z\} \in C$)
        if ($mergeable(\Theta(C \cup \{f\}))$)
          $C \leftarrow C \cup \{f\}$
    In $\bar{G}$, merge all $*_e \in \Theta(C)$

for (each pair of edges $(e_{(a,b)}, e_{(b,c)}) \in E^*$)          //$K_3$
  if ($\exists e_{(a,c)} \in E^*$)
    $C \leftarrow \{a, b, c\}$
    if ($mergeable(\Theta(C))$)
      In $\bar{G}$, merge all $*_e \in \Theta(C)$
```

Fig. 5. Alg.3: Finding clique substructures

Finding 4-cliques and larger clique-like structures takes O(n_e^2) where n_e is the number of edges , i.e., $n_e = |E^*|$, in the transformation graph $G^* = (V^*, E^*)$. Note that we operate on G^* and not on $\bar{G}$. By construction, there is a significant size difference between G^* and $\bar{G}$. For instance, in iPlane data set in Section V-B, G^* has 17K nodes and 67K edges while $\bar{G}$ has 445K nodes and 1M edges. Therefore, comparisons of the complexity figures for $\bar{G}$ and G^* should consider this important difference for practical utilization of the corresponding approaches. In the first part of the algorithm, we take ($n_e(n_e-1)/2$) pair of edges and look for the existence of other 6 edges among the end points using a connectivity matrix. Once we find a 4-clique, we pick two other edges connected to the nodes in the 4-clique and check if they also form a 4-clique or not. In either case, all of the processed edges are removed from further consideration. Additional queries do not increase the time complexity since we only change the search order while the search space is the same. Similarly, finding all 3-cliques takes O(n_e^2). To find a 3-clique, the algorithm simply takes edge pairs of a node and looks for the existence of an edge between the other two nodes.

Structure 3 (Complete Bipartite): The third structure that is formed between an anonymous router $*^v$ and its known neighbors can be seen as a complete bipartite subgraph, $K_{k,l}$, among the known neighbors of $*^v$. This type of structure occurs when the topology map before anonymous router resolution process includes *-substrings $(n_i, *^v, n_j)$ where $i = [1, k]$ and $j = [1, l]$ among all known neighbors of $*^v$. Figure 4-d presents an example structure of this type where the data set includes *-substrings among all known neighbors of the anonymous node. This structure is referred to as a $K_{2,3}$ complete bipartite graph and corresponds to the topology presented in Figure 4-c. In general, this type of structures appear in traceroute data set that are collected using (k,m)-traces (where $k \ll m$).

```
INPUT: $G^* = (V^*, E^*)$ from Alg.3 in Figure 5
for (each pair of nodes $(a, b) \in V^*$)      //$K_{2,3}$ or larger
  if ($\exists \{e_{(a,c)}, e_{(a,d)}, e_{(a,e)}, e_{(b,c)}, e_{(b,d)}, e_{(b,e)}\} \in E^*$)
    $B2 \leftarrow \bigcup x$ where $\exists \{e_{(a,x)}, e_{(b,x)}\} \in E^*$
    $B \leftarrow \{a, b\} \cup B2$
    if ($mergeable(\Theta(B))$
      while ($\exists e_{(f,x)} \in E^* \ \forall x \in B2$)
        if ($mergeable(\Theta(B \cup \{f\}))$)
          $B \leftarrow B \cup \{f\}$
      In $\bar{G}$, merge all $*_e \in \Theta(B)$

for (each pair of edges $(e_{(a,b)}, e_{(c,d)}) \in E^*$)      //$K_{2,2}$
  if ( ($\exists \{e_{(a,d)}, e_{(b,c)}\} \in E^*$ or $\exists \{e_{(a,c)}, e_{(b,d)}\} \in E^*$)
    $B \leftarrow (a, b, c, d)$
    if ($mergeable(\Theta(B))$)
      In $\bar{G}$, merge all $*_e \in \Theta(B)$
```

Fig. 6. Alg.4: Finding complete bipartite subgraphs

In this step, we are interested in identifying complete bipartite graph structures among neighbors of anonymous nodes. Similar to the previous case, we work on the newly built graph $G^* = (V^*, E^*)$. We identify complete bipartite graphs on G^* and use them to resolve the corresponding anonymous routers on the original graph G. Given a graph, finding the largest complete bipartite subgraph is an NP-complete problem. In our work, we are interested in not only the largest but also all other complete bipartite subgraphs in G^*. Therefore, we use a heuristic in which we first search for a small size, i.e., $K_{2,3}$, complete bipartite subgraph in G^* and then grow it to a larger complete bipartite subgraph[2]. Once we get a large size complete bipartite subgraph in G^*, we resolve the corresponding anonymous nodes in the original graph G. Figure 6 presents the corresponding algorithm.

Finding $K_{2,3}$ and larger complete bipartite graphs takes O($n_v^2.n_n$) where n_v is the number of vertices, i.e., $n_v = |V^*|$, and n_n is the average node degree in the transformation graph $G^* = (V^*, E^*)$. In the algorithm, we take each pair of nodes and look whether they are in a $K_{2,3}$. After identifying a $K_{2,3}$, we look for larger complete bipartite graphs $K_{2,m}$ and then $K_{n,m}$ that contain the identified $K_{2,m}$. Growing from $K_{2,3}$ to $K_{2,m}$ can be done while checking the existence of $K_{2,3}$ and hence it does not add to the overall complexity. Growing from $K_{2,m}$ to $K_{n,m}$ is done by identifying the common neighbors of the second set, i.e., m nodes, by taking intersection of their neighbor set using the connectivity matrix in O(n_n). Similarly, finding $K_{2,2}$ takes O(n_e^2) where n_e is the edges, i.e., $n_e = |E^*|$, in G^*. Algorithm finds $K_{2,2}$ by taking two edges and checking for the existence of two other edges connecting both ends.

[2] Note that a given complete bipartite graph, e.g., $K_{2,3}$, may be a subgraph of another complete bipartite subgraph, e.g., $K_{3,4}$.

```
INPUT: $\bar{G} = (\bar{V}, \bar{E})$ from Alg.4 in Figure 6
$K \leftarrow \bigcup\{a \mid (|E^*_a| > 1)\}$ where $E^*_a = \{e_{(a,*_e)} \in \bar{E}\}$

while $(K \neq \phi)$
  Pick $a \in K$ where $|E^*_a|$ is minimum
  $N^*_a \leftarrow \{*_e \mid e_{(a,*_e)} \in \bar{E}\}$
  while $(N^*_a \neq \phi)$
    Pick $*_e$ from $N^*_a$
    $N^*_a \leftarrow N^*_a \ - \ *_e$
    while $(\exists *_f \in N^*_a$ and $mergeable(*_e, *_f))$
      In $\bar{G}$, merge $*_e$ and $*_f$
      $N^*_a \leftarrow N^*_a \ - \ *_f$
    $K \leftarrow \{K \cup *_e\} - a$
```

Fig. 7. Alg.5: Finding star substructures

Structure 4 (Star): The last structure aims at resolving anonymous routers as shown in Figure 4-e. This structure typically appears in path traces collected from a single vantage point toward a number of destinations. The observed topology then corresponds to the one presented in Figure 4-f. Note that a similar structure is obtained when multiple vantage points trace toward the same destination.

We identify this type of structures by grouping anonymous neighbors (e.g., $*_e$) of nodes (e.g., a, the head node of *-substrings in Figure 4-f). We then combine all anonymous neighbors of a node a into the same anonymous router in the topology map under trace preservation condition. In the process, we first sort the nodes based on their number of anonymous neighbors. We then start processing from the node with the smallest number of neighbors. Figure 7 presents the corresponding algorithm.

Resolving these structures takes O($n_k.n_d$) where n_k is the number of nodes with anonymous node neighbors and n_d is the average number of anonymous node neighbors per node. In the algorithm, the size of K grows and we need to consider how fast it grows. Note that in the worst case there may be $n_k/2$ new nodes after processing all n_k nodes with anonymous node neighbors. Therefore, there might be a series of new nodes, i.e., $n_k/2 + n_k/4 + ... + 1$, added to set K. However, this series is less than the following geometric series

$$n_k. \sum_{r=0}^{\infty} \left(\frac{1}{2}\right)^r = \frac{n_k}{1 - 1/2} = 2.n_k$$

Therefore, the complexity remains to be O($n_k.n_d$).

V. Evaluations

In this section, we use simulations and experiments to evaluate the accuracy and the performance of Graph Based Induction (GBI) approach. In our simulations, we use both synthetic and genuine topologies to compare the accuracy of GBI with that of Initial Pruning (IP) and Neighbor Matching (NM) approaches. IP is a commonly used technique in previous studies and corresponds to the Alg.1 in Figure 2. NM is used in previous study by Jin et al. and is similar to Alg.5 in Figure 7. Note that we did not compare GBI with graph minimization approach of [17] and dimensionality reduction appraoch of [11] due to their high complexities. In our experiments, we use a genuine Internet topology data to assess the effectiveness of GBI in a more practical setting.

A. Simulation-based Evaluations

In this section, we compare GBI with IP and NM based on their accuracy and completeness. We also consider the impact of omitting anonymous router resolution completely in our presented results. For our comparisons, we use two network topologies as follows:

- A *genuine network* obtained from AMP measurement infrastructure [14] on August 31, 2006. This topology consists of path traces among 130 vantage points and includes 2,376 routers and 3,770 links after the IP alias resolution process.
- A *synthetic transit-stub (TS) network* generated by GT-ITM topology generator [18]. This network consists of 50,000 nodes and 138,500 links.

In our work, we use the above topologies as the underlying actual networks. We randomly select a number of nodes in the actual networks as anonymous (between 2% to 14% of them) and collect a number of (k,m) path traces to represent traceroute-collected path traces. We use (10,500) and (10,1000) path traces from the genuine network and (10,1000), (10,2000), and (10,3000) path traces from the synthetic network. Note that in this study, we only consider anonymous routers of Type 1, 4 & 5 as IP and NM approaches are completely ineffective for Type 2 & 3 anonymous routers.

The below metrics are used for our evaluations. First two metrics assess the accuracy and the rest provide insight into topological characteristic of obtained networks.

Edit distance: Edit distance is the number of primitive operations required to transform the sampled graph to the actual graph. The primitive operations we define are *node split* and *node merge*. During node split, a node v is split into two nodes and a new vertex v' is added to the set of vertices. This operation fixes false positives. During node merge, two separate nodes v and v' are merged into a single node v and v' is removed from the set of vertices. This operation fixes false negatives. Note that this metric is similar to the widely used graph distance. Edit distance however fits better in our context as it explicitly counts both false positives and false negatives.

We analyze edit distance of all samples and observe that on average edit distance of NM and IP are 39% and 379% higher than that of GBI, respectively. Figure 8 shows the edit distance for (10,500) AMP and (10,2000) TS samples. In some cases, NM algorithm does not improve over the initial pruning considerably. As an example, in 14% case for (10,2000) TS sample, edit distance reduces from 1,800 with IP to 1,784 with NM. Overall, in all samples, GBI has the smallest edit distance, i.e., GBI has least number of errors. Finally, Table I presents the average edit distances for all samples (including AMP and TS samples). According to the results, GBI-based topologies have least number of errors in all cases.

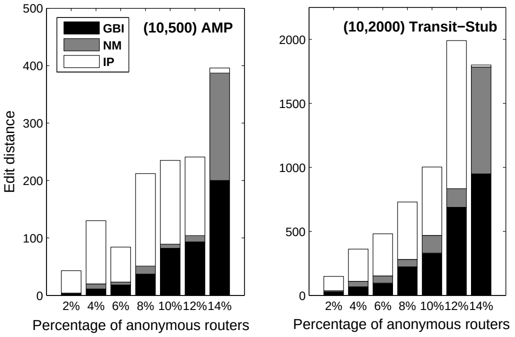


Fig. 8. Edit distances for two samples.

	2%	4%	6%	8%	10%	12%	14%
Initial	3,798	4,576	8,093	10,519	11,045	16,383	19,079
IP	135	229	501	666	718	967	1,252
NM	32	58	189	272	319	502	1,190
GBI	23	37	146	215	274	430	633

TABLE I
AVERAGE EDIT DISTANCES

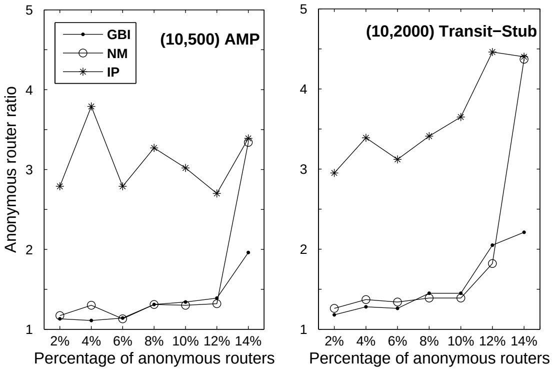


Fig. 9. Anonymous router ratio for two samples.

	2%	4%	6%	8%	10%	12%	14%
Initial	59.8	48.6	57.4	52.0	45.9	53.9	58.2
IP	3.4	3.3	3.9	3.9	3.9	3.7	4.2
NM	1.3	1.3	1.4	1.5	1.4	1.6	3.7
GBI	1.2	1.2	1.5	1.6	1.6	1.7	2.1

TABLE II
AVERAGE ANONYMOUS ROUTER RATIOS

Anonymous router ratio: Anonymous router ratio is the ratio of the number of anonymous routers in the induced topology to the ones in the actual topology. It helps observe the inflation caused by the anonymous routers and assess the effectiveness of the resolution algorithms. Note that an anonymous router ratio of 1 does not necessarily mean that anonymous routers are correctly resolved without considering the edit distance.

We analyze anonymous router ratio to assess the effectiveness of each approach. On average, anonymous router ratios are 53.7, 3.72, 1.75, and 1.56 for initial, IP, NM, and GBI approaches respectively. Figure 9 shows the anonymous router ratio for (10,500) AMP and (10,2000) TS samples. Note that in some cases NM results in smaller anonymous router ratios indicating that NM results in smaller size topology graphs. However, close examination of the results indicates that NM has higher edit distance in each of these cases as compared to GBI. The achieved reduction in graph size is basically due to higher number of false positives. Finally, Table II presents the average anonymous router ratios for all samples (including AMP and TS samples). According to the results, GBI-based topologies have small ratios. In some cases, NM results in smaller anonymous router ratios but, given the edit distance comparisons in Table I, these cases are related to inaccurate merging (i.e., false positives).

In the rest of our analysis, we compare different anonymous router resolution approaches based on their impact on several topological characteristics in the resulting graphs. These characteristics include topology size, clustering coefficient, node degree, and characteristic path length.

Topology size, in terms of number of nodes and number of links, reveals the basic information about a network. Due to anonymous routers, a traceroute-based topology map may include artificial nodes and artificial links. Anonymous router resolution is a process that tries to eliminate such nodes and links with a minimal error. The first two rows in Table III compare the different approaches based on the average percentage increase in the number of nodes and the number of links in their resulting topologies. According to the results, on average, GBI based topologies have the least number of artificial nodes and artificial links. In addition, as the table shows, the initial pruning (IP) step eliminates the highest number of artificial nodes and artificial links.

Clustering coefficient is the ratio of the number of triangles to the number of triplets in a graph. This metric characterizes the connectivity density of a given graph. This metric is a useful metric to compare different anonymous router resolution techniques as the resolution process modifies a given graph and changes its connectivity density. The last row in Table III compares the different approaches based on the percentage difference of clustering coefficients as compared to that of the original graph. According to the table, most of the initial topologies have approximately zero clustering coefficient. On average, the clustering coefficient of the GBI based graph is closest to that of the original graph suggesting that the resulting graph resembles the most to the original graph in terms of its connectivity density. In addition to above metrics, we study node degree and path length related characteristics but do not see a significant difference between the original topologies and the topologies obtained by IP, NM, and GBI.

In summary, in this section, we have compared GBI with practical approaches IP and NM, and have shown that GBI performs better or comparable in terms of edit distance or anonymous router ratio. We have also shown that GBI based

	Initial	IP	NM	GBI
Number of nodes	+262%	+21%	+5%	+4%
Number of edges	+593%	+83%	+55%	+49%
Clustering coefficient	-98%	-40%	-19%	-17%

TABLE III
CHANGES IN GRAPH CHARACTERISTICS.

graphs are closer to the original graphs in terms of important topological characteristics. In addition, as analyzed in the next section, run time of GBI is three orders of magnitude smaller than NM. Finally, GBI handles all five types of anonymity whereas NM is completely ineffective for Type 2 & 3 anonymity, which are common in the Internet but not considered in the simulations.

B. Experimental Results

In this section, we use a genuine topology data to analyze the practicality of our algorithm. We use a data set that has 18M path traces, 426K IP addresses, and 9M anonymous nodes after filtering inaccurate and incomplete path traces. This data set is collected on June 2, 2007 from 190 vantage points to approximately 90K destinations by iPlane measurement system [12]. After resolving IP aliases the number of known nodes reduces from 426K to 229,425 along with 8,974,939 anonymous nodes.

In this experiment, we apply our algorithms to resolve anonymous routers in the data set. Table IV presents the results for each step. Alg.1 applies the initial pruning to reduce the number of anonymous nodes significantly. Alg.2 then identifies anonymous nodes due to Type 2 & 3 anonymity (i.e., the ones due to ICMP rate limiting or due to congestion at the router). This step resolves over 73% of the existing anonymous nodes. Note that previous approaches do not handle this type of anonymity. Next, Alg.3 handles clique-like structures by using a transformation graph G^*. Note that due to the (k,m) nature of the iPlane data set, the number of observed clique-like structures and the number of resolved anonymous nodes is small. In the following step, Alg.4 detects complete bipartite structures in G^* and resolves corresponding anonymous nodes in $\bar{G}$. Finally, Alg.5 detects star structures in $\bar{G}$ to resolve more anonymous nodes. Overall, our algorithm reduces 8,972,939 anonymous nodes to 98,610 anonymous nodes.

We briefly examine the operational overhead of our approach. Table IV presents the complexities of each step of our algorithm and the size of the corresponding variables in the data set. We use these numerical values to estimate comparative run time overhead of our algorithm with that of NM proposed in [11]. Based on these complexity values, highest complexity is introduced by Alg.3, i.e., approximately 4.5*10^9 operations in the worst case. The NM approach has time complexity O(n^2) where n is the total number of nodes in the data set after initial pruning. For iPlane data n is 1,031,004 requiring approximately 10^{12} steps for NM in the worst case. Note that the dimensionality reduction approach of [11] would take 10^{18} operations while the graph minimization approach of [17] would take 10^{30} operations on this data set.

	#Anonymous	#Resolved	Complexity	# Operations
Alg.1	8,972,939	8,171,360	O($n_s.log(n_s)$)	7,238,749 * 6.86
Alg.2	801,579	585,887	O($n_t.log(n_s)$)	5,773,796 * 6.86
Alg.3	215,692	533	O(n_e^2)	67,219^2
Alg.4	215,159	61,968	O($n_v^2.n_n$)	17,439^2 * 3.85
Alg.5	153,191	54,581	O($n_k.n_d$)	118,285 * 3
Final	98,610	8,874,329		

TABLE IV
GRAPH BASED INDUCTION TECHNIQUE ON IPLANE DATA SET.

VI. CONCLUSION

In this paper, we have developed and applied a graph based reduction approach to resolve anonymous routers. Our work improves the state of the art in anonymous router resolution in terms of accuracy and practicality. Regarding accuracy, our solution addresses all five anonymity types (see Section III) whereas the previous approaches handle only three anonymity types. Regarding practicality, the run time complexity of our algorithm is significantly less than that of the existing algorithms. Our experiments on iPlane data has shown a significant reduction in the practical run time overhead of our approach (approximately, 4.5*10^9 operations) as compared to the previous approaches (approximately, 10^{12}, 10^{18}, or 10^{30} operations), in the worst case.

REFERENCES

[1] B. Augustin, X. Cuvellier, B. Orgogozo, F. Viger, T. Friedman, M. Latapy, C. Magnien, and R. Teixeira, *Avoiding traceroute anomalies with Paris traceroute*, in IEEE IMC, Rio de Jenario, Brazil, Oct 25-27, 2006.

[2] S. Bilir, K. Sarac, and T. Korkmaz, *Intersection characteristics of end-to-end Internet paths and trees,* in IEEE ICNP, Boston, MA, Nov 6-9, 2005.

[3] A. Broido and KC. Claffy, *Internet topology: Connectivity of IP graphs,* in SPIE ITCom, Denver, CO, Aug 19-24, 2001.

[4] B. Cheswick, H. Burch, and S. Branigan, *Mapping and visualizing the Internet*, in ACM USENIX Annual Technical Conference, San Diego, CA, June 18-23, 2000.

[5] D.J. Cook and L.B. Holder, *Mining graph data*, John Wiley & Sons, Hoboken, NJ, 2006.

[6] D. Feldman and Y. Shavitt, *An Optimal Median Calculation Algorithm for Estimating Internet Link Delays from Active Measurements*, in IEEE E2EMON, Munich, Germany, May 21, 2007.

[7] R. Govindan and H. Tangmunarunkit, *Heuristics for Internet map discovery*, in IEEE INFOCOM, Tel Aviv, Israel, Mar 26-30, 2000.

[8] M. Gunes and K. Sarac, *Analytical IP alias resolution*, in IEEE ICC, Istanbul, Turkey, Jun 11-15, 2006.

[9] M. Gunes and K. Sarac, *Importance of IP alias resolution in sampling Internet topologies*, in IEEE Global Internet, Anchorage, AK, May 11-12, 2007.

[10] M. Gunes and K. Sarac, *Inferring Subnets in Router-level Topology Collection Studies*, in IEEE IMC, San Diego, CA, Oct 24-26, 2007.

[11] X. Jin, W.-P. K. Yiu, S.-H. G. Chan, and Y. Wang, *Network topology inference based on end-to-end measurements*, in IEEE J-SAC, 24(12) pp. 2182–2195, Dec 2006.

[12] H. V. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani, *iPlane: An information plane for distributed services*, in USENIX OSDI, Seattle, WA, Nov 6-8, 2006.

[13] T. Matsuda, H. Motoda, and T. Washio, *Graph-based induction and its applications*, in Advanced Engineering Informatics, 16(2) pp 135–143, Apr 2002.

[14] A. McGregor, H.-W. Braun, and J. Brown, *The NLANR network analysis infrastructure*, in IEEE Communications Magazine, 38(5) pp. 122-128, May 2000

[15] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson, *Measuring ISP topologies using rocketfuel*, IEEE/ACM ToN, 12(1) pp. 2–16, Feb 2004.

[16] R. Teixeira, K. Marzullo, S. Savage, and G. Voelker, *In Search of Path Diversity in ISP Networks*, in ACM IMC, Miami, FL, Oct 27-29, 2003.

[17] B. Yao, R. Viswanathan, F. Chang, and D. Waddington, *Topology inference in the presence of anonymous routers*, in IEEE INFOCOM, San Francisco, CA, Mar 30 - Apr 3, 2003.

[18] E.W. Zegura, K.L. Calvert, and M.J. Donahoo, *A quantitative comparison of graph-based models for Internet topology*, in IEEE/ACM ToN, 5(6) pp. 770-783, 1997.