Efficient Network Tomography for Internet Topology Discovery

Brian Eriksson, Gautam Dasarathy, Paul Barford, and Robert Nowak

***Abstract*—Accurate and timely identification of the router-level topology of the Internet is one of the major unresolved problems in Internet research. Topology recovery via tomographic inference is potentially an attractive complement to standard methods that use TTL-limited probes. Unfortunately, limitations of prior tomographic techniques make timely resolution of large-scale topologies impossible due to the requirement of an infeasible number of probes to perform topology discovery. In this paper, we describe new techniques that aim toward the practical use of tomographic inference for accurate router-level topology measurement. We introduce methodologies based on a Depth-First Search (DFS) Ordering that clusters end hosts based on shared infrastructure, and enables the logical tree topology of the network to be recovered accurately and efficiently. We evaluate the capabilities of our algorithms in simulation and find that our methods will reconstruct topologies using less than 2% of the probes required by exhaustive methods and less than 15% of the probes needed by the current state-of-the-art tomographic approach. We also present results from a case study in the live Internet where we show our DFS-based methodologies can recover the logical router-level topology more accurately and with fewer probes than prior techniques.**

I. Introduction

Mapping the Internet's router-level topology is a compelling objective for network measurement. In addition to their appeal to network researchers, accurate and timely maps of the Internet have a wide range of applications and are of particular importance in network management, operations and security. A large number of prior studies have focused on efficient Internet router-level topology discovery using active probe-based, `traceroute`-like measurements *e.g.,* [1], [2]. However, when using TTL-limited, `traceroute`-like measurements for reconstructing topologies, one is faced with the serious challenges of resolving anonymous routers [3] and router aliases [4]. More recent research in [5], [6] has shown how a combination of `traceroute` and Record Route probes can improve the accuracy of topology estimation. However, Record Route probes are also limited in that only a small percentage of Internet routers respond to the Record Route option. Finally, TTL-limited measurements are unable to reveal Layer-2 hops or hops through MPLS clouds, which further reduces the accuracy of reconstructed topologies.

There are alternatives to TTL-limited measurements for Internet topology recovery. One technique that has shown promise is tomographic inference of router-level topology using end-to-end measurements of packet delay or loss. Initial work on network tomography methodologies focused on the use of multicast measurements [7], [8]. Multicast inference is attractive due to the total number of probes necessary (*i.e.,* probing complexity) is $O(N)$, where N is the number of end hosts in the topology. However, the extremely limited deployment of open, multicast-enabled nodes renders these techniques impractical for a wide-scale topology study of the Internet. More recent work has focused on network tomography using unicast probes to obtain a measure of similarity between pairs of end hosts [9], [10], [11]. Unfortunately, these techniques are also impractical due to the quadratic number of probes $\left(O\left(N^2\right)\right)$ needed to resolve the topology.

The goal of our work is to advance the capabilities of RTT-based tomography such that it can be used effectively and efficiently for router-level topology discovery in the Internet. We face a number of challenges in this work, including understanding how to construct RTT probes based on the limitations of typical end hosts for measurement and common case congestion characteristics of end-to-end paths. However, the specific focus of this paper is on reducing the total number of pairwise probes that are required in order to resolve the network topology.

In this paper, we exploit the idea of arranging end host targets in a *Depth-First Search (DFS) Order*. For a collection of end hosts in a tree topology, any of the non-unique ordinal lists found from a depth-first search on the leaf nodes of a tree structure (considered here as end hosts) can be defined as a DFS ordering. This can also be considered a topological sort [12] on only the end hosts of a logical topology. The idea of topological sort has been explored previously in sensor network literature in [13], where a topological sort of the nodes in a sensor network provides efficient routes through the network with low power consumption. Due to the focus on wire-line networks in this paper, we are not able to chose the routing. Instead we will use a modified version of topological sorting to efficiently reconstruct the logical routing from Internet measurements.

We show how a DFS ordering clusters target end hosts based on the amount of shared infrastructure. Given this shared infrastructure clustering, we demonstrate how the resulting similarity matrix has a special structure. By exploiting this special similarity matrix structure, the number of delay-based

B. Eriksson is with the Department of Computer Science, Boston University, Boston, MA, 02215 USA e-mail: eriksson@cs.bu.edu.

G. Dasarathy, and R. Nowak are with the Department of Electrical and Computer Engineering, University of Wisconsin - Madison.

P. Barford is with the Department of Computer Science, University of Wisconsin - Madison and Qualys.

This work was supported in part by the National Science Foundation (NSF) grants CCR-0325653, CCF-0353079, CNS-0716460 and CNS-0905186, and AFOSR grant FA9550-09-1-0140. Any opinions, findings, conclusions or other recommendations expressed in this material are those of the authors and do not necessarily reflect the view of the NSF or the AFOSR.

probes used to resolve the logical topology of a balanced ℓ-ary tree can be reduced from the current state-of-the-art tomography probing methodology ([14]) by using our new DFS Ordering-based methodologies. We then show how the similarity measurement condition assumed by the current state-of-the-art methodologies are too restrictive and that the DFS ordering can be exploited to resolve topologies using only $O(\ell N \log(N))$ pairwise probes under less restrictive pairwise similarity conditions for a balanced ℓ-ary tree. To the best of our knowledge, the resulting probing complexity of the developed DFS-based algorithms are the lowest probing complexity for any unicast tomography algorithm. We believe this reduction in the number of probes is an important step toward unicast tomography being considered a feasible and practical topology discovery mechanism.

The remainder of the paper is structured as follows. In Section II, we describe previous delay-based tomographic methods for Internet logical topology discovery and other related work. The topology discovery problem and the *DFS Ordering* idea are introduced in Section III. In Section IV, an efficient logical topology discovery algorithm is described given a *DFS Ordering* of the end hosts. Given that real world topologies will not have the DFS ordering known, in Section V we show how a valid DFS ordering of the end hosts can be found from relatively few topology measurements given a specified *margin condition* on the observed similarities between end hosts. In Section VI we show how the DFS ordering can be resolved when only a less restrictive *monotonic condition* is considered on the observed pairwise similarities. Finally, in Section VII the results of our experiments on both synthetic and real world topologies show the improvements of our procedure for estimating the logical topology of a network over previous techniques. We conclude and describe future work in Section VIII.

II. Related Work

The initial work most directly related to the research in this paper is the application of bottom-up agglomerative clustering-based methodologies explored in [7], [8], [15] for use on Internet topology reconstruction. Bottom-up agglomerative clustering resolves the hierarchical clustering of a set of objects with pairwise similarity values by finding the maximum similarity element and merging the rows/columns of the similarity matrix corresponding to those two end hosts, then finding the next maximum element and merging those rows/columns of the similarity matrix to the new maximum element. This process is repeated until all the rows/columns are merged and the hierarchical clustering of the entire set of objects is resolved. In terms of network tomography, this method requires obtaining the entire similarity matrix (e.g., $O\left(N^2\right)$ pairwise probes given N number of end hosts in the topology). The agglomerative clustering methodology will be considered the worst case probing bounds, as it performs an exhaustive probing of every possible pair of end hosts in the network. The large number of probes required is due to the decoupling of topology measurements and topology inference, where no information from prior measurements is used to inform new measurements, and topology inference is performed completely separate from the measurement process.

A more efficient probing methodology is the Sequential Topology Inference algorithm from [14]. This work sequentially builds the logical tree structure and leverages the current estimated logical tree structure to determine where the next probe pair measurements should be performed. This work couples topology inference and measurement into one process by exploiting the tree structure of the topology. For a balanced ℓ-ary tree (a balanced tree where each non-leaf node has exactly ℓ children), this reduces the number of probes needed from $O\left(N^2\right)$ for agglomerative clustering, to at most $\ell N \log_\ell(N)$. In Sections V and VI, we show how improvements to this performance can be obtained by exploiting the structure of not just the tree topology, but the structure of the topology *measurements*. We show how our methodologies can further reduce the number of probes compared to this current state-of-the-art. Additionally, the Sequential Topology methodology requires strict conditions on the observed pairwise similarities. In Section VI, we present an efficient tomography methodology that requires less restrictive conditions on the observed pairwise similarity values.

III. Depth-First Search (DFS) Order

We consider the standard tomography problem of resolving the logical tree topology rooted at an end host that is transmitting probes through the network. Common to any unicast tomography methodology is the ability to estimate a measure of similarity between pairs of end hosts. This similarity can vary depending on the probing methodology, such as observed covariance for the Network Radar technique [11] or observed delay deviation for the sandwich probes method from [15]. Our efficient topology reconstruction techniques presented in this paper are agnostic to the choice of tomography probing methodology. Therefore, for end host x_i, x_j, we denote the observed pairwise similarity as $s_{i,j}$, ignoring the mechanism used to generate the similarity. Regardless of the probing methodology used, we assume that the observed similarity measurements satisfy the *Monotonic Condition* for logical topology shared path, where $p_{i,j}$ is the number of logical routers shared between end hosts x_i and x_j in the paths from the root node to the two end hosts

Condition 1: The observed pairwise similarity matrix $\mathbf{S}$ and shared path matrix $\mathbf{P}$ satisfy the **Monotonic Condition** if for all end hosts i, j, k, the observed similarity satisfies $s_{i,j} > s_{j,k}$ if and only if the tree topology shared path satisfies $p_{i,j} > p_{j,k}$.

The foundation for the work in this paper is the idea of *Depth-First Search (DFS) Ordering* of the end hosts. A depth-first search (DFS) is a tree search that starts at the tree root and progresses down the tree labeling each node and backtracking only when a node has been explored fully (e.g., every child of that node has been labeled). We formally define a DFS Ordering as *any* ordinal list of the end hosts (which will be considered the leaf nodes of the logical routing tree) that would satisfy the ordering found by a depth-first search of that logical tree structure ignoring the labeling of the internal nodes of the tree.

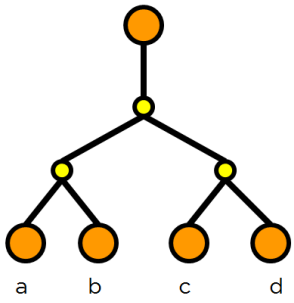

Fig. 1. Example simple logical topology in a proper DFS Order.

For the tree structure in Figure 1, we can find the following valid DFS orderings all of which would satisfy a depth-first search on the tree topology:

$$\begin{array}{cccc} \{a,b,c,d\} & \{a,b,d,c\} & \{b,a,c,d\} & \{b,a,d,c\} \\ \{c,d,a,b\} & \{d,c,a,b\} & \{c,d,b,a\} & \{d,c,b,a\} \end{array}$$

There are also many possible end host orderings that would violate a DFS ordering property of the tree. For example, the ordering $\{a,c,d,b\}$ does not satisfy a depth-first search of the end hosts.

The power of a depth-first search can be seen when examining the shared logical path matrix $\mathbf{P}$. For a proper DFS ordering of the topology in Figure 1 ($\{a,b,c,d\}$), the ordered shared path matrix $\mathbf{P}_{proper}$ will be found as:

$$\mathbf{P}_{proper} = \left[\begin{array}{c|cccc} - & a & b & c & d \\ \hline a & 2 & 2 & 1 & 1 \\ b & 2 & 2 & 1 & 1 \\ c & 1 & 1 & 2 & 2 \\ d & 1 & 1 & 2 & 2 \end{array}\right]$$

And for an improper DFS ordering ($\{a,c,d,b\}$), the out-of-order shared path matrix ($\mathbf{P}_{improper}$) will be found as:

$$\mathbf{P}_{improper} = \left[\begin{array}{c|cccc} - & a & c & d & b \\ \hline a & 2 & 1 & 1 & 2 \\ c & 1 & 2 & 2 & 1 \\ d & 1 & 2 & 2 & 1 \\ b & 2 & 1 & 1 & 2 \end{array}\right]$$

Using a set of end hosts in DFS order, we can state the following proposition,

Proposition 1: Given a set of end hosts $\{x_1, x_2, ..., x_N\}$ in a DFS Ordering, the resulting shared path matrix $\mathbf{P}$ has the following structure:

$$p_{i,i+1} \geq p_{i,i+k} \quad : \text{for all } k = \{1,2,...,N-i\}$$

For all $i = \{1,2,...,N\}$.

Proof: Consider the case where the end hosts are in a proper DFS ordering, but $p_{i,i+j} < p_{i,i+k}$ (for some $0 \leq j < k$). This states that end hosts x_i, x_{i+k} have more shared infrastructure than x_i, x_{i+j} (*e.g.,* a longer shared path length from the root). This implies the tree structure has x_i and x_{i+k} at some point of depth (*e.g.,* level of shared infrastructure), while x_{i+j} is located at some point in the tree structure at some shallower point in the structure in comparison to x_i (*e.g.,* at some level with less shared infrastructure than x_i and x_{i+k}). But by the depth-first search ordering, this requires $j > k$ as a depth-first search would encounter x_{i+k} before x_{i+j}, thus violating the setup of the problem. Therefore, if the end hosts are in a proper DFS order, Proposition 1 must hold. ■

IV. Logical Topology Discovery using DFS Ordering

Assume that all the end hosts in an unknown topology are already in a DFS order. Given this ordering, we look to estimate the logical topology using a non-exhaustive number of pairwise similarity observations. Using the results from Proposition 1 and the Monotonic Condition, we can state that the similarity matrix $\mathbf{S}$ has structure directly relating to the structure of the shared path matrix $\mathbf{P}$.

Proposition 2: Given a set of end hosts $\{x_1, x_2, ..., x_N\}$ in proper DFS Ordering with the similarity matrix $\mathbf{S}$ satisfying the Monotonic Condition, then the similarity matrix $\mathbf{S}$ will have the following property:

$$s_{i,i+1} \geq s_{i,i+k} \quad : \text{for all } k = \{1,2,...,N-i\}$$

For all $i = \{1,2,...,N\}$.

Proof - Given Proposition 1 and the monotonic condition, it is trivial to see this property of similarity matrix $\mathbf{S}$.

Using the results of this proposition, we can now devise an efficient logical tree reconstruction procedure in Algorithm 1 for a set of end hosts in DFS order with pairwise similarities satisfying the monotonic condition.

Algorithm 1 - DFS Ordered Logical Topology Discovery Algorithm

Given:

- Set of observed pairwise similarities for end hosts in DFS order, $\{s_{1,2}, s_{2,3}, ..., s_{N-1,N}\}$
- Set of tree nodes, $\widehat{V} = \{x_1, x_2, ..., x_N\}$.
- Set of tree edges, $\widehat{E} = \emptyset$.
- Reconstructed tree topology, $\widehat{\mathcal{T}} = \left(\widehat{V}, \widehat{E}\right)$.
- Set of merge nodes $V' = \{x_1, x_2, ..., x_N\}$.

Main Body:

For $k = \{1,2,...,N-1\}$

1) Find $\widehat{j} = \text{argmax}_{j=\{1,2,...,|V'|-1\}} s_{j,j+1}$.
2) Create new interior node, x^*.
3) Add new interior node, $\widehat{V} = \widehat{V} \bigcup x^*$.
4) Add new edges to interior node, $\widehat{E} = \widehat{E} \bigcup \{V'\left(\widehat{j}\right), x^*\} \bigcup \{V'\left(\widehat{j}+1\right), x^*\}$.
5) Update the merge nodes, set $V'\left(\widehat{j}\right) = x^*$, $V'\left(\widehat{j}+1\right) = \emptyset$.
6) Update the similarity values, set $s_{\widehat{j},\widehat{j}+1} = 0$.

Proposition 3: Using the set of end hosts in a proper DFS Order and pairwise similarities satisfying the monotonic condition, only $N-1$ pairwise probes (the similarity values

$s_{i,i+1}$ for $i = \{1, 2, ..., N-1\}$) are needed to reconstruct the unknown logical tree topology using Algorithm 1.

Proof: For each end host, bottom-up agglomerative clustering requires only knowledge of which other end host has the most shared topology. Given the monotonic condition, this is equivalent to finding the end host with the largest similarity magnitude. Unfortunately, to acquire this knowledge, it was previously necessary to obtain all possible similarity values (on the order of $O\left(N^2\right)$ pairs for N end hosts). Given both the DFS ordering of the end hosts and the monotonic condition, the only similarity values necessary to infer the logical topology will be (for each end host x_i, with $i = \{1, 2, ..., N\}$) the value of the immediately preceding end host similarity ($s_{i-1,i}$) and the immediately successive end host similarity ($s_{i,i+1}$). This is due to the Proposition 2 stating that the similarity $s_{i,i+1} \geq s_{i,i+j}$ for any $j > 1$. Therefore, end host x_i will share the most infrastructure in the topology with either x_{i+1} or x_{i-1}. In order to reconstruct the tree topology, only the similarity values associated with these two pairs of end hosts, x_i, x_{i-1} and x_i, x_{i+1} are needed. The magnitude of these two similarity values ($s_{i-1,i}, s_{i,i+1}$) will directly inform us as to the structure of the logical topology using a modified bottom-up agglomerative clustering procedure that only considers this subset of pairwise similarities, which is Algorithm 1. ■

V. Margin-Based DFS Ordering Estimation

A limitation of the methodology in Algorithm 1 is that it is based around the assumption that the end hosts are already correctly arranged in a proper depth-first search (DFS) order. In any non-trivial problem, this ordering will not be known. Instead, given no prior knowledge of the topology, we must estimate a proper DFS Ordering from targeted measurements. To resolve the ordering, in this section we require a more restrictive assumption on the similarities than the Monotonic Condition. We assume the observed tomographic measurements satisfy the *Margin Condition*.

Condition 2: The observed similarity matrix $\mathbf{S}$ and shared path matrix $\mathbf{P}$ satisfy the **Margin Condition** if for all end hosts i, j, k, the observed similarities satisfy $s_{i,j} > s_{j,k} + \delta$ (for some specified $\delta > 0$) if and only if the tree topology shared path satisfies $p_{i,j} > p_{j,k}$.

Remark: Delay-based unicast methods exploit the observation of shared queuing delay between pairs of end hosts. Consider the value δ to be the minimum queuing delay induced by a router in the network topology. Also, note that the monotonic condition is a special case of the margin condition where $\delta = 0$. Therefore, any tree reconstruction methodology that holds under the monotonic condition will also hold under the margin condition for any δ.

The intuition behind our ordering methodology is as follows. Given a random ordering of the set of end hosts, consider choosing a single end host (x_1) and obtaining the pairwise similarities between this end host and all other end hosts in the set ($= \{s_{1,2}, s_{1,3}, s_{1,4}, ..., s_{1,N}\}$). Some end hosts will have very high pairwise similarity, while others will have significantly less shared infrastructure with the chosen end host and therefore have low pairwise similarity. By sorting these obtained similarity values, this would place the end hosts that have more shared infrastructure at one end of the list, and the end hosts with little shared infrastructure at the other end of the list. We define ordering with respect to a single end host as the *partial ordering*, π, of the set of end hosts, such that $\pi : \{2, 3, ..., N\} \rightarrow \{2, 3, ..., N\}$ (with $s_{1,\pi_i} \geq s_{1,\pi_{i+j}}$: for all $j \geq 1$). This partial ordering cannot be considered a proper DFS ordering for the reasons shown in Figure 2-(Left). While a significant fraction of the end hosts will have observed pairwise similarity within some margin δ when compared against the chosen end host (in this case, s_A), a proper DFS order inside this cluster is unknown using only this single end host vantage point.

This implies that pairwise similarities from more than a single end host vantage point will be required to correctly order the entire set of end hosts. From the example Figure 2-(Right), consider that any similarity will be within a δ deviation of one of three values $\{s_A, s_B, s_C\}$. Having correctly ordered the end hosts, we now look to order the subclusters (*e.g.,* what should the ordering be of all the end hosts with similarity within a δ deviation of s_A for the topology?). One could consider dividing the set of end hosts into similarity clusters and for each cluster repeating this probing process. This would be performed by taking a new intracluster vantage point, and then reordering the intracluster end hosts based on the pairwise similarity values with this new vantage point. Therefore, we look to a recursive methodology that at each iteration bisects the ordered set of end hosts into two topologically significant clusters. This reduces our objective to the single problem of finding the correct end host to bisect the set of end hosts at each iteration of the algorithm.

The simplest approach to this bisection problem is for a given margin value δ, sorting the similarity values and finding all the possible bisection candidate end hosts (denoted by set $\mathcal{I}$) where $i \in \mathcal{I}$ if the similarity difference between the partial ordered i-th (denoted as π_i) and the partial ordered $(i+1)$-th (denoted as π_{i+1}) similarity value is more than δ, thereby indicating a topology difference between the two end hosts with respect to end host x_1,

$$\mathcal{I} = \{i : s_{1,\pi_i} - s_{1,\pi_{i+1}} > \delta\}$$

The bisection point will be the end hosts in $i^* \in \mathcal{I}$ that results in the two bisected end host sets $\mathbf{X}_1 = [x_1, x_2, ..., x_{\pi_{i^*}}]$ and $\mathbf{X}_2 = \left[x_{\pi_{i^*+1}}, ..., x_{\pi_N}\right]$ to be closest in size to each other for all choices of $i^* \in \mathcal{I}$.

$$i^* = \underset{i \in \mathcal{I}}{\arg\min} \left| \pi_i - \frac{N}{2} \right| \tag{1}$$

Using this intuition, we present Algorithm 2 to find a proper DFS Ordering for a set of end hosts using this recursive bisection methodology.

Proposition 4: Using Algorithm 2, the number of pairwise similarities needed to correctly obtain a proper DFS Ordering for a balanced ℓ-ary tree (where each non-leaf node has ℓ children) satisfying the margin condition with N end hosts is

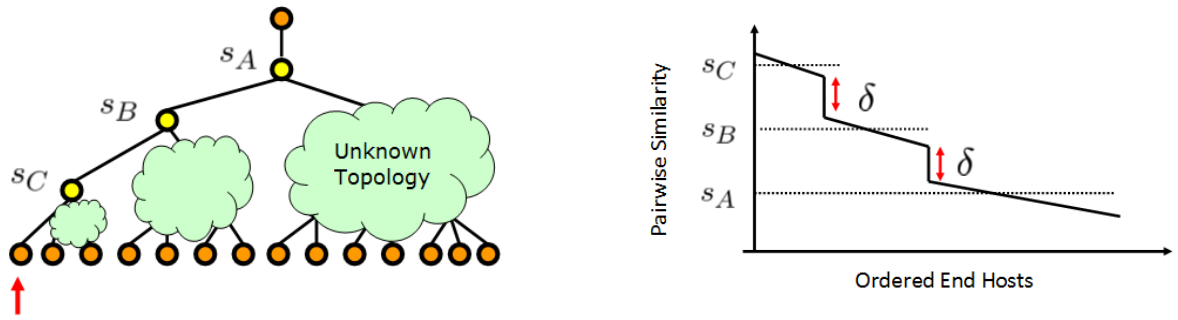

Fig. 2. Margin-based End Host Ordering - (Left) Example of similarity values from a single end host revealing partial ordering, (Right) - Resulting ordered pairwise similarity values given the margin condition.

Algorithm 2 - Margin-Based DFS Ordering Algorithm - marginOrder($\mathbf{X}$, δ)

Given:

- Unordered set of N end hosts with unknown logical topology $\mathbf{X} = \{x_1, x_2, ..., x_N\}$
- Margin condition, $\delta > 0$.

Main Body:

1) Find the pairwise similarity values with respect to end host x_1, $\{s_{1,2}, s_{1,3}, ..., s_{1,N}\}$.
2) Sort the set of pairwise similarities, obtaining the partial ordering with respect to end host x_1, $\pi : \{2, 3, ..., N\} \rightarrow \{2, 3, ..., N\}$.
3) Find $\mathcal{I}$, the set of indices where the difference between consecutive sorted similarity values is greater than δ.

$$\mathcal{I} = \{i : s_{1,\pi_i} - s_{1,\pi_{i+1}} \geq \delta\}$$

4) Bisect the set of sorted end hosts $\mathbf{X}$ at the index of i^* that creates two sets most equal in size using Equation 1, creating sorted end host subsets $\mathbf{X}_1 = \{x_1, x_{\pi_1}, ..., x_{\pi_{i^*}}\}$, $\mathbf{X}_2 = \{x_{\pi_{i^*+1}}, ..., x_{\pi_{N-1}}\}$.
5) If $|\mathbf{X_1}| > 2$, then find $\mathbf{X}_1 = \text{marginOrder}(\mathbf{X_1}, \delta)$
6) If $|\mathbf{X_2}| > 2$, then find $\mathbf{X}_2 = \text{marginOrder}(\mathbf{X_2}, \delta)$
7) Return the reordered set of end hosts, $\mathbf{X} = [\mathbf{X}_1 \mathbf{X}_2]$

upper bounded by $p(\ell) N \log_\ell N$ probe pairs (where $p(\ell) = \left(\frac{\ell+1}{2} - \frac{1}{\ell}\right)$).
Proof of this proposition is found in the appendix.

Our margin-based DFS Ordering topology reconstruction methodology consists of finding the DFS ordering of the end hosts using Algorithm 2, and then resolving the logical topology using Algorithm 1. Using Proposition 4 and Proposition 3, it is trivial to bound the total number of pairwise probes required by the margin-based topology reconstruction methodology.

Proposition 5: Using Algorithm 2 and Algorithm 1, the logical topology for a balanced ℓ-ary tree with N end hosts can be reconstructed requiring at most $N\left(p(\ell)\log_\ell N + 1\right)$ pairwise similarities that satisfy the margin condition.

Note that our margin-based DFS Ordering topology reconstruction methodology has a pairwise probing upper bound that requires fewer pairwise similarities than the current state-of-the-art efficient tomography approach in [14], which also requires the margin condition on the pairwise similarities.

VI. Monotonic-Based DFS Ordering Estimation

In many real world tomography problems, the margin condition will be too restrictive. Instead, we look to efficiently reconstruct the tree topology when only the monotonic condition holds, where for any triple of end hosts x_i, x_j, x_k, the pairwise similarities satisfy $s_{i,j} > s_{i,k}$ if and only if the shared path values satisfy $p_{i,j} > p_{i,k}$. Our ordering estimation methodology begins similar to the margin-based approach where the ordering is found by a recursive bisection of the set of end hosts, but we also require a small number of additional pairwise similarities to reinforce each bisection choice.

Given a random ordering of the set of end hosts ($\{x_1, x_2, ..., x_N\}$), consider choosing a single end host (x_1) and obtaining the similarity measurements between this and all other hosts in the set ($= \{s_{1,2}, s_{1,3}, ..., s_{1,N}\}$) to find the partial ordering, π. Common to the methodology in Section V, we look to reduce the total number of pairwise measurements needed by bisecting the partial order π and repeatedly taking pairwise measurements only with respect to each bisected subset. While the margin condition allows us to easily find the bisection point x_i^* through the use of the similarity difference (where, $s_{1,\pi_i} - s_{1,\pi_{i+1}} > \delta$), under just the monotonic condition this deviation will not indicate changes in the tree topology. Therefore, we must devise a different methodology for finding the bisection point that represents a split in the tree, as a large similarity difference will not necessarily imply a topology change assuming only the monotonic condition holds on the pairwise similarity values.

Imagine performing bottom-up agglomerative clustering on the set of partial ordered end hosts. Pairs of end hosts are repeatedly merged together until all are combined into a single cluster by the final merge operation. Prior to the final merge operation, the set of ordered end hosts are bisected into two sets separated by a split in the tree topology. When all the end hosts are in DFS order, this split can be defined by a single point that bisects these two sets in the ordering, x^*. Using standard bottom-up agglomerative clustering, this would require examination of all possible pairwise similarity values

(on the order of N^2 for N end hosts). By exploiting the partial ordering of the end hosts, π, this split can be found using a modified agglomerative clustering procedure which requires significantly fewer than all the possible pairwise similarities.

Consider performing agglomerative clustering on only a subset of m end hosts evenly spaced in the partial ordering (*e.g.,* such that there are $\frac{N}{m}$ between each agglomerative clustering end host) in Figure 3-(Left). The final merge of the agglomerative clustering on these m end hosts (Figure 3-(Center)) will reveal which subset of $\frac{N}{m}$ contain the bisection point x^*. Once this subset is revealed, we choose m new end hosts inside this subset of $\frac{N}{m}$ (set $\mathbf{X}_R$ in Figure 3-(Right)) and again perform agglomerative clustering to find a subset of interest, this time of size $\frac{N}{m^2}$ (again, containing the bisection point x^*). This process is repeated until the subset of interest is of size less than or equal to m, where our modified agglomerative clustering will resolve the top most split in the tree and bisection point x^*.

This recursive agglomerative clustering algorithm is stated in Algorithm 3. The power of this methodology is found by the distillation of agglomerative clustering measurements, requiring at most $m^2 \log_m N$ pairwise measurements, while standard agglomerative clustering would require $O\left(N^2\right)$ pairwise measurements.

Algorithm 3 - Recursive Agglomerative Clustering Algorithm - recursiveAgg($\mathbf{X}$,m)

Given:

- Set of N end hosts in partial order $\mathbf{X} = \{x_1, x_2, ..., x_N\}$.
- Number of clustering hosts, m.

Main Body:

1) Pick a subset of end host indices $\mathbf{U} \subset \{1, 2, \ldots, N\}$ such that $|\mathbf{U}| = m$ and these indices are uniformly spaced in $\{1, 2, \ldots, N\}$.
2) Using bottom-up agglomerative clustering, reconstruct the tree structure of the subset of end hosts indexed by $\mathbf{U}$.
3) Find the last merge of the agglomerative clustering algorithm, between end host subsets $\mathbf{U}_L, \mathbf{U}_H$ (such that $\mathbf{U}_L \bigcup \mathbf{U}_H = \mathbf{U}$ and $\mathbf{U}_L \bigcap \mathbf{U}_H = \emptyset$).
4) Define the largest similarity in the 'lower' set $s_L^{max} = \max_{i \in \mathbf{U}_L} s_{1,i}$, related to end host x_L^{max}. And the smallest similarity in the 'higher' set, $s_H^{min} = \min_{i \in \mathbf{U}_H} s_{1,i}$, related to end host x_H^{min}.
5) Divide the set X into three subsets: the 'higher' set $X_H = \{x_i : s_{1,i} \geq s_H^{min}\}$, the 'lower' set $X_L = \{x_i : s_{1,i} \leq s_L^{max}\}$ and the 'residual' set $X_R = (\mathbf{X} \setminus (\mathbf{X}_L \cup \mathbf{X}_H)) \bigcup \{x_L^{max}, x_H^{min}\}$.
6) If $|\mathbf{X}| \leq m$, return end host x_H^{min}
7) Else, return end host x^* = recursiveAgg($\mathbf{X}_R$,m)

The recursive agglomerative clustering methodology is only to find the bisection object at a single iteration of the ordering algorithm. This methodology must be repeated multiple times to find a true DFS ordering on the end hosts. The complete methodology for finding the DFS ordering under the monotonic condition is described in Algorithm 4.

Algorithm 4 - Monotonic-Based DFS Ordering Algorithm - monotonicOrder($\mathbf{X}$)

Given:

- Unordered set of N end hosts with unknown logical topology $\mathbf{X} = \{x_1, x_2, ..., x_N\}$
- Number of clustering nodes, m.

Main Body:

1) Find the pairwise similarity values with respect to end host x_1, $\{s_{1,2}, s_{1,3}, ..., s_{1,N}\}$.
2) Sort the set of pairwise similarities, obtaining the partial ordering with respect to end host x_1, $\pi : \{2, 3, ..., N\} \rightarrow \{2, 3, ..., N\}$.
3) Using the Recursive Agglomerative Clustering methodology (Algorithm 3), find the bisection index, i^*.
4) Create sorted end host subsets $\mathbf{X}_1 = \{x_1, x_{\pi_1}, ..., x_{\pi_{i^*}}\}$, $\mathbf{X}_2 = \{x_{\pi_{i^*+1}}, ..., x_{\pi_{N-1}}\}$.
5) If $|\mathbf{X_1}| > 2$, then find $\mathbf{X}_1$ = monotonicOrder($\mathbf{X_1}$, δ)
6) If $|\mathbf{X_2}| > 2$, then find $\mathbf{X}_2$ = monotonicOrder($\mathbf{X_2}$, δ)
7) Return the reordered set of end hosts, $\mathbf{X} = [\mathbf{X}_1 \mathbf{X}_2]$

Proposition 6: Using Algorithm 4 and Algorithm 1, the logical topology for a balanced ℓ-ary tree with N end hosts can be reconstructed requiring at most $N\left((\ell + 9)\log_2 N + 1\right)$ pairwise similarities that satisfy the monotonic condition. Proof of this proposition is found in the appendix.

VII. Experiments

To assess performance of our efficient network tomography methodologies, we perform experiments on both synthetic and real-world Internet topologies.

A. Prior Methods

1) Agglomerative Clustering: Consider having access to every pairwise similarity value for all N end hosts in the topology. Given complete knowledge of the similarity matrix we would have knowledge of which set of end hosts have the largest similarity in the entire topology, and hence, knowledge of which set of end hosts have the most shared infrastructure from the root node. For the bottom-up Agglomerative Clustering algorithm (applied in a networking context in [7], [8], [15]), at each step of the algorithm the current set of end hosts with the largest similarity are found, and a logical router is inserted connecting this set of end hosts together. The corresponding rows/columns in the similarity matrix for these two end hosts are then merged together. This process is repeated until there are no more rows/columns in the matrix are left to merge. The main disadvantage to this methodology is that it requires knowledge of all $\frac{N(N-1)}{2}$ similarity values, which is effectively exhaustive probing of all the end hosts in the network.

2) Sequential Logical Topology Reconstruction: Informed by the generic tree structure of the topology, the work in [14] shows that the number of probes needed to reconstruct the topology can be considerably reduced when the observed similarities satisfy the margin condition. This methodology

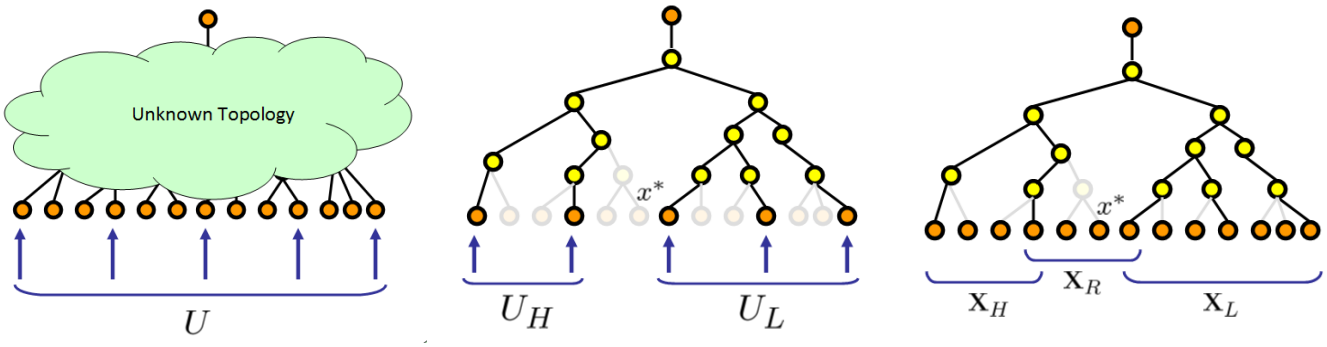

Fig. 3. **(Left)** Set of end hosts in partial order, with subset of end hosts U given $m = 5$. **(Center)** Tree structure found through agglomerative clustering on the set U, with top-level split partitioning into U_H, U_L. **(Right)** Resulting end host sets X_R, X_L, X_H.

depends upon sequentially building the tree topology for each end host. For a given end host, the pairwise similarity for this end host and all the nodes that are children of the root node are found. Given the child of the root node with the largest similarity (and thus the most shared topology), c_i^*, the pairwise similarity is found between the end host and the children of the specified child (c_i^*). The similarity value (and margin δ) determines whether the end host is a sibling, child, or descendant of c_i^*. This process is repeated until the leaf node with the largest pairwise similarity is found. On a balanced ℓ-ary tree (a balance tree where each non-leaf node has ℓ children), each end host requires at most $\ell \log_\ell N$ pair probes, thus for the entire topology the number of pairwise probes needed is upper bounded by $\ell N \log_\ell N$.

A comparison of the probing upper bounds for all four topology reconstruction methodologies (agglomerative clustering, sequential, margin-based DFS ordering, and monotonic-based DFS ordering) is seen in Table I. The margin-based DFS ordering algorithm is found to have the smallest probing complexity upper bound of all algorithms, but requires the restrictive margin condition on the observed similarity values. In comparison with the Sequential Topology algorithm, which also requires the margin condition, from Equation 5 we can see that $\frac{p(\ell)}{\ell} \leq 0.5625$ for all choices of ℓ, therefore the margin-based DFS ordering will require fewer pairwise probes than the Sequential Topology algorithm for any feasible ℓ, N topologies. The monotonic-based DFS ordering algorithm has upper bounds requiring more probes than the two margin-based methodologies, but will be guaranteed to succeed when the similarity margin δ is not required on the observed pairwise similarities. The monotonic-based DFS ordering algorithm also requires significantly fewer pairwise probes than the agglomerative clustering methodology, which is the only other methodology that is also guaranteed to reconstruct topologies under the monotonic condition.

B. Synthetic Noise-Free Experiments

Synthetic topologies enable us to analyze the capabilities of our methods with full ground truth and over a range of network sizes. The synthetic topologies are generated using the Heuristically Optimized Topology framework [16]. Heuristically Optimized Topology is one of the latest and most realistic network topology generators that create graphs that have properties that are consistent with many of those observed in the Internet, incorporating societal, engineering, and economic constraints. Using the Orbis topology generator [17], we scale the Heuristically Optimized Topologies to create three different sized topologies, with $N = \{768, 1497, 2261\}$.

To test the performance of our algorithms in a noise-free environment, for each synthetic topology every node is assigned a random similarity value, with synthesized pairwise similarities being the sum of the node similarity values (with the smallest router similarity assigned, $\delta = 0.1$) along the shared shortest path from the root node to the two end hosts under consideration. Due to this experiment being noise-free with similarities that satisfy the margin condition, all topology reconstruction methodologies will perfectly reconstruct the topologies from the pairwise measurements.

TABLE II
COMPARISON OF NUMBER OF PROBES NEEDED TO ESTIMATE LOGICAL TOPOLOGY USING SYNTHETIC ORBIS TOPOLOGIES.

	Agglomerative Clustering	Sequential [14] Algorithm	
Number of End Hosts (N)	# Pairwise Sim. Needed	# Pairwise Sim. Needed	Percentage of Agglomerative Pairs
768	294,528	52,774	17.9%
1,497	1,119,756	112,375	10.0%
2,261	2,554,930	128,104	5.0%

In Table II, we present the number of pairwise probes needed by the prior tomography methodologies, agglomerative clustering and Sequential. By exploiting the tree structure, the state-of-the-art Sequential method requires at most 20% of the pairwise probes the agglomerative cluster methodology requires. In Table III, we present the resulting number of pairwise similarities required to resolve the logical topology for our monotonic-based and margin-based DFS Ordering methodologies. As seen in the tables, both DFS Ordering methodologies do significantly better than the exhaustive agglomerative clustering approach. In terms of the 768-end host topology, we obtain a pairwise probe savings with both DFS methodologies requiring at most 2% of the pairwise probes used by the exhaustive agglomerative clustering approach. As expected, due to the more restrictive conditions, the margin-based DFS methodology consistently requires fewer pairwise

TABLE I
THE TREE RECONSTRUCTION METHODOLOGIES A FOR BALANCED ℓ-ARY TREE. (WHERE $p(\ell) = \left(\frac{\ell+1}{2} - \frac{1}{\ell}\right)$ IS SUBLINEAR IN ℓ)

Methodology	Pairwise Probe Upper Bounds	Satisfies Margin-based Reconstruction?	Satisfies Monotonic-based Reconstruction?
Sequential [14]	$\ell N \log_\ell N$	Yes	No
Margin-Based DFS Ordering	$N\left(p\left(\ell\right)\log_\ell N + 1\right)$	Yes	No
Agglomerative Clustering	$\frac{1}{2}N\left(N-1\right)$	Yes	Yes
Monotonic-Based DFS Ordering	$N\left(\left(\ell+9\right)\log_2 N + 1\right)$	Yes	Yes

similarities than the monotonic-based DFS methodology. From these experiments it was seen that our new margin-based DFS method requires at most 10% of the number of pairwise similarities that the Sequential method require. Meanwhile, even the Monotonic DFS methodology (which does not require the margin condition) outperforms the Sequential methodology by requiring at most 15% of the number of pairwise probes used by the state-of-the-art approach.

C. Real World Experiments

To observe the performance of our algorithm on real-world topologies, we chose 9 DNS servers located at small-to-medium sized colleges in the New England geographic area. Using the DNS server addresses and `traceroute` probes we discovered the following logical tree topology in Figure 4 starting at the University of Wisconsin - Madison as the root node. Using delay-based unicast tomographic probes, the pairwise similarities were found between pairs of the end hosts in the topology.

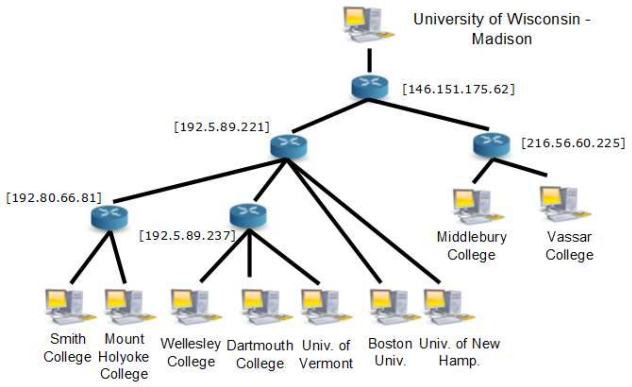

Fig. 4. Real world topology used to test tomography methods

The delay-based unicast tomographic technique we will focus on is *Network Radar* [11]. Network Radar uses round trip time (RTT) measurements as the basis for topology inference and was developed as an attempt to obviate the need for significant coordinated measurement infrastructure. Consider the simple logical topology in Figure 5. Back-to-back packets originating from end host a will travel along the same path until router R. It can be assumed that any delays encountered before router R induced by router queuing delays will cause highly correlated delays for both back-to-back packets (due to both packets being in the same router queues). Assuming that any delays encountered between the two packets past router R are uncorrelated, then the level of covariance between the RTT delays found from a series of back-to-back packets ($cov\left(d_b, d_c\right)$) will inform us to the amount of shared logical topology between paths $\{a, b\}$ and $\{a, c\}$. Thus, for our real world experiments the pairwise similarity will be considered as, $s_{b,c} = cov\left(d_b, d_c\right)$.

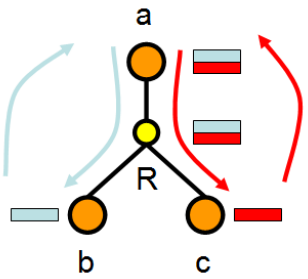

Fig. 5. Example of Network Radar on simple logical topology.

Using the Network Radar methodology, we observed 1,500 back-to-back round-trip-time delay samples for every end host pair in our real-world topology (Figure 4). Due to imperfect round-trip-time measurements and other delay noise measured, the sample similarity was found to not be perfectly correlated to the `traceroute` observed shared path length. Therefore, for any estimation procedure based on the sample similarity, there will be potential errors in the reconstructed topology [1]. In order to determine the accuracy of our estimated topologies, we must develop a metric that compares our estimated topologies to the ground-truth topology.

Informed by our monotonic condition, we consider the following accuracy measure for our reconstructed tree topologies. Consider a triple of end hosts $\{a, b, c\}$ that exists in our estimated topology. From our estimated logical topology, we can predict whether there is a longer shared path between end hosts $\{a, b\}$ or end hosts $\{a, c\}$. For the estimated logical topology $\widehat{\mathcal{T}}$, these two paths will be denoted $\widehat{p}_{a,b}$ and $\widehat{p}_{a,c}$ respectively. And for the true topology, these two true path lengths will be denoted as $p_{a,b}$ and $p_{a,c}$ respectively. The more accurate our estimated topology, the more often our estimated topology will return the correct answer for whether $\{a, b\}$ has more shared infrastructure than $\{a, c\}$. The percentage of times we are correct with this problem will be denoted as p. For all possible triples in our set of end hosts ($\mathbf{X}$), this *shared path classification rate* can be found by,

[1] This could be improved upon by taking more back-to-back sample probes or using a DAG card to obtain more accurate time information, but for this paper we will focus on the case where neither improvement is available.

TABLE III
COMPARISON OF NUMBER OF PROBES NEEDED TO ESTIMATE LOGICAL TOPOLOGY USING SYNTHETIC ORBIS TOPOLOGIES.

	Margin-Based DFS Ordering Algorithm			Monotonic-Based DFS Ordering Algorithm		
End Hosts (N)	# Pairwise Sim. Needed	Percentage of Agglomerative Probes	Percentage of Sequential Pairs	# Pairwise Sim. Needed	Percentage of Agglomerative Probes	Percentage of Sequential Pairs
768	3,604	1.22%	6.83%	5,511	1.87%	10.44%
1,497	7,771	0.69%	6.92%	10,571	0.94%	9.41%
2,261	11,848	0.46%	9.25%	17,929	0.70%	14.0%

$$p = f(\mathbf{X}) \sum_{a \in \mathbf{X}} \sum_{b \in \mathbf{X}} \sum_{c \in \mathbf{X}} \mathbf{1}(\widehat{p}_{a,b} > \widehat{p}_{a,c}) \mathbf{1}(p_{a,b} > p_{a,c}) \quad (2)$$

With the value,

$$f(\mathbf{X}) = \left(\sum_{a \in \mathbf{X}} \sum_{b \in \mathbf{X}} \sum_{c \in \mathbf{X}} \mathbf{1}(p_{a,b} > p_{a,c}) \right)^{-1},$$

Where $\mathbf{1}(x) = 1$ if the condition x holds while $\mathbf{1}(x) = 0$ if the condition x does not hold.

The baseline for any topology reconstruction algorithm will be to outperform a naive randomly reconstructed topology with end hosts and interior nodes connected at random into a tree topology. Our shared path classification rate will be the metric we use to assess how accurate the estimated topologies are.

1) Results: Due to the Sequential Algorithm and both DFS Ordering algorithms having performance sensitive to initial ordering of end hosts, the performance of the three algorithms are averaged over 500 random permutations of the end hosts. Averaging over many random permutations eliminates any order bias from the results.

The two margin-based methodologies (Sequential and margin-based DFS ordering) have a tunable margin parameter, δ, that must be chosen. To give the prior margin-based methodology (Sequential) every possible advantage, for each experiment the performance of the Sequential algorithm is shown for *the best possible value of* δ at each level of probing. Meanwhile, our new margin-based DFS ordering methodology has a constant value of δ across all levels of probing (with $\delta = 0.1$).

For the real-world topology in Figure 4, the corresponding shared path classification rate (from Equation 2) for the two margin-based topology reconstruction algorithms (margin-based DFS Ordering and Sequential) and a baseline random random methodology can be seen in Figure 6-(Left) versus a restricted total number of delay probes available. We find that the margin-based DFS ordering methodology performs significantly better than both the Sequential and random topologies. Surprisingly, the Sequential methodology requires a large number of pairwise probes to outperform the random topologies, we believe this is due to a heavy reliance on the margin δ between the similarity measurements. While our margin-based methodology also requires the δ margin condition, through the exploitation of DFS ordering, our methodology will be more robust to violations of the margin condition (as evident by the improved performance). Given the probing complexity in Table I, it is very likely that the accuracy improvements for the new DFS Ordering algorithm will further grow as the size of the topology increases.

The shared path classification rate (from Equation 2) for the two monotonic-based topology reconstruction algorithms (monotonic-based DFS Ordering and agglomerative clustering) can be seen in Figure 6-(Right) versus a restricted total number of delay probes available. As seen in the figure, for a wide range of available pairwise probes, our monotonic-based DFS ordering methodology results in a more accurate tree topology than the standard bottom-up agglomerative clustering methodology. For example, to obtain the same tree reconstruction accuracy ($p = 0.7$), the monotonic-based DFS methodology requires 1,700 fewer delay-based measurements sent through the network compared with the exhaustive agglomerative clustering methodology (3,800 delay probes for DFS ordering, while agglomerative clustering requires 5,600 delay probes).

VIII. CONCLUSIONS / FUTURE WORK

Despite concerted efforts, generating accurate maps of the router-level topology of the Internet remains a compelling objective in Internet measurement. Standard TTL-based and Record Route methods for discovering router-level network topology have well known limitations that motivate development of alternative topology measurement methods. One such method is the application of tomographic inference to network delay measurements in order to recover the underlying topology. While network tomography for topology discovery has been examined in the past, it has yet to be widely used in practice due to it's own set of limitations.

The goal of our work is to address the shortcomings of RTT measurement-based network tomography for discovering Internet logical topology. Tomographic methodologies described in prior work required an impractical number of probes. In this paper, we describe algorithms that considerably reduce the number of pairwise probes needed to resolve logical topologies. The ability to reduce the number of pairwise probes is reliant on exploiting the idea of a *Depth-First Search (DFS) Ordering* of the end hosts. We analyze the capabilities of our algorithms on a set of large-scale synthetically generated topologies. The experiments on these topologies show our new methodologies require only 2% of the probes used by an exhaustive methodology, and roughly 15% of the probes used by the current state-of-the-art. Results from a small-scale real-world Internet experiment further validate the performance of our algorithms. The significant reduction in the number of probes needed opens delay-based tomographic topology discovery techniques to new avenues of applications.

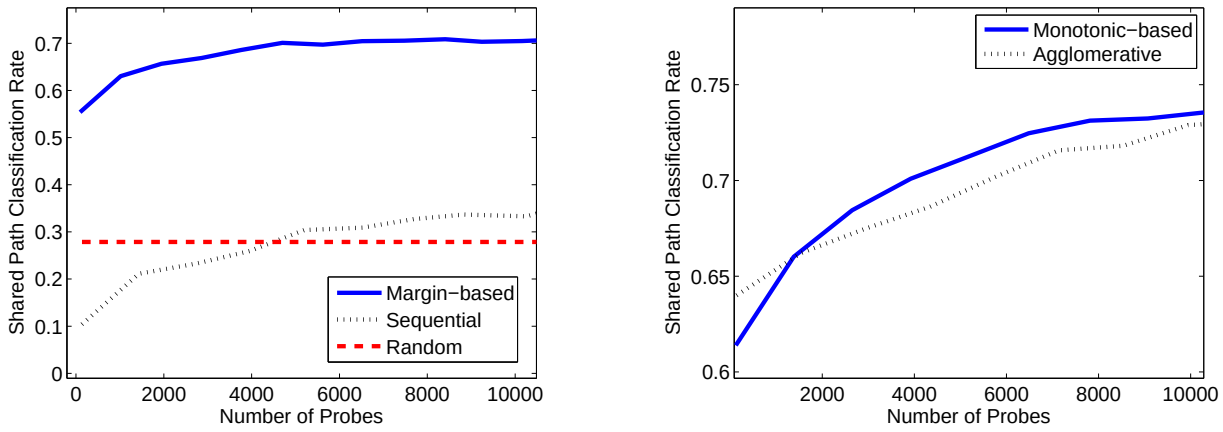

Fig. 6. (Left) - Topology reconstruction results for the two margin-based algorithms (margin-based DFS Ordering, Sequential) and a baseline random topology. (Right) - Topology reconstruction results for the two monotonic-based algorithms (monotonic-based DFS Ordering and agglomerative clustering).

IX. Appendix

A. Proof of Proposition 4

Using Algorithm 2, consider the first step, end host x_1 will be chosen and the similarity values will be found between x_1 and $x_2, x_3, ..., x_N$. Given the ℓ-ary balanced property of the tree and the margin condition, after sorting the similarity values this implies that the first iteration of the algorithm will divide the set of end hosts into a group of $\frac{N}{\ell}$ end hosts and a group of $\frac{(\ell-1)N}{\ell}$ end hosts corresponding to the first branch on the first level of the tree as seen in Figure 7-(Left). Consider further subdividing the set of $\frac{(\ell-1)N}{\ell}$ end hosts, where a random end host is chosen in the set and $\frac{(\ell-1)N}{\ell}-1$ similarity measurements are taken. Our bisection algorithm would then subpartition into a group of $\frac{N}{\ell}$ end hosts and a group of $\frac{(\ell-2)N}{\ell}$ end hosts, again corresponding to the first level of the tree as seen in Figure 7-(Right). In these initial steps of the algorithm each iteration is resolving a branch off the first level of this tree, clustering into ℓ sets of $\frac{N}{\ell}$ end hosts each relating to a branch off the first level of the tree.

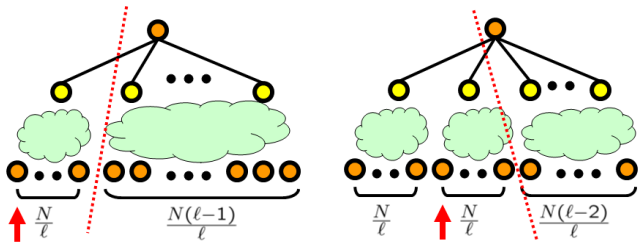

Fig. 7. **(Left)** The first split taken on a balanced ℓ-ary tree. **(Right)** The second split taken on a balanced ℓ-ary tree. Both splits indicated by the dotted line, the arrow indicates the randomly chosen end host similarity values are measured against.

After the tree has been divided past the first level, the problem can now be considered ordering the ℓ number of subtrees each with $\frac{N}{\ell}$ end hosts. Using this recursive property, we can state the number of probes needed for a balanced ℓ-ary tree with N leaf nodes as $f_\ell(N)$.

$$\begin{aligned} f_\ell(N) &\leq N + \frac{\ell-1}{\ell}N + \cdots + \frac{2}{\ell}N + \ell f_\ell\left(\frac{N}{\ell}\right) \\ &= \frac{N}{\ell}(2+3+\cdots+\ell) + \ell f_\ell\left(\frac{N}{\ell}\right) \\ &\overset{(a)}{=} Np(\ell) + \ell f_\ell\left(\frac{N}{\ell}\right) \\ &\overset{(b)}{\leq} Np(\ell) + \ell\left\{\frac{N}{\ell}p(\ell) + \ell f_\ell\left(\frac{N}{\ell^2}\right)\right\} \\ &= 2Np(\ell) + \ell^2 f_\ell\left(\frac{N}{\ell^2}\right) \\ &\vdots \\ &\overset{(c)}{\leq} p(\ell)N(\log_\ell N) \end{aligned} \tag{3}$$

(4)

Where

$$p(\ell) := \frac{(2+3+\cdots+\ell)}{\ell} = \left(\frac{\ell+1}{2} - \frac{1}{\ell}\right) \tag{5}$$

B. Proof of Proposition 6

Consider a balanced ℓ-ary tree with a set of N end hosts $X = \{x_1, x_2, ..., x_N\}$. Let $f_\ell(N)$ denote the number of pairwise measurements required to discover a valid DFS ordering on the set X and let $s_m(N) \leq m^2 \log_m N$ be the number of measurements required to find the exact split by agglomerative clustering m chosen end hosts out of N and proceeding recursively (as indicated by Algorithm 3) until the split point x^* is found. Using the recursive property of the

algorithm, we can state:

$$\begin{aligned}
f_\ell(N) &\leq (N + s_m(N)) + \\
&\quad \left(\frac{\ell-1}{\ell}N + s_m\left(\frac{\ell-1}{\ell}N\right)\right) + \cdots \\
&\quad \cdots + \left(\frac{2}{\ell}N + s_m\left(\frac{2}{\ell}N\right)\right) + \ell f_\ell\left(\frac{N}{\ell}\right) \\
&\overset{(a)}{=} Np(\ell) + \sum_{i=2}^{\ell} s_m\left(\frac{i}{\ell}N\right) + \ell f_\ell\left(\frac{N}{\ell}\right) \\
&\overset{(b)}{\leq} Np(\ell) + \ell m^2 \log_m N + \ell f_\ell\left(\frac{N}{\ell}\right) \\
&\leq Np(\ell) + \ell m^2 \log_m N + \\
&\quad \ell\left\{\frac{N}{\ell}p(\ell) + \ell m^2 \log_m\left(\frac{N}{\ell}\right) + \ell f_\ell\left(\frac{N}{\ell^2}\right)\right\} \\
&\leq 2Np(\ell) + \left(\ell + \ell^2\right) m^2 \log_m N + \ell^2 f_\ell\left(\frac{N}{\ell^2}\right) \\
&\leq Np(\ell)\log_\ell N + \left(\sum_{j=1}^{\log_\ell N} \ell^j\right) m^2 \log_m N \\
&\leq Np(\ell)\log_\ell N + \frac{N-1}{\ell-1} m^2 \log_m N \\
&\leq N\ell \log_\ell N + Nm^2 \log_m N
\end{aligned}$$

In (a), $p(\ell) := \frac{2+3+\cdots+\ell}{\ell} = \frac{\ell+1}{2} - \frac{1}{\ell}$.

Given that we control the agglomerative clustering procedure in Algorithm 4, we can reduce the total pairwise measurements needed by setting $m = 3$ (as $m \geq 3$). This results in the total pairwise measurements needed by Algorithm 4 to resolve DFS ordering to be less than $N(\ell + 9)\log_2 N$ measurements for N end hosts clustered in a ℓ-ary balanced tree. Combined with Proposition 3, we see that to resolve the tree topology requires only $N((\ell + 9)\log_2 N + 1)$ pairwise similarities that satisfy the monotonic condition.

REFERENCES

[1] B. Donnet, P. Raoult, T. Friedman, and M. Crovella, "Deployment of an Algorithm for Large-Scale Topology Discovery," in *IEEE Journal of Selected Areas in Communications, Special Issue on Sampling the Internet*, 2006, pp. 2210–2220.

[2] N. Spring, R. Mahajan, and D. Wetherall, "Measuring ISP Topologies with Rocketfuel," in *Proceedings of ACM SIGCOMM '02*, Pittsburgh, PA, August 2002.

[3] B. Yao, R. Viswanathan, F. Chang, and D. Waddington, "Topology Inference in the Presence of Anonymous Routers," in *In IEEE INFOCOM*, 2003, pp. 353–363.

[4] M. H. Gunes and K. Sarac, "Resolving IP aliases in building traceroute-based Internet maps," in *Technical Report*, 2006.

[5] R. Sherwood and N. Spring, "Touring the internet in a TCP sidecar," in *IMC '06: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, 2006, pp. 339–344.

[6] R. Sherwood, A. Bender, and N. Spring, "DisCarte: A Disjunctive Internet Cartographer," in *Proceedings of ACM SIGCOMM*, Seattle, WA, August 2008.

[7] N. Duffield and F. L. Presti, "Network tomography from measured end-to-end delay covariance," vol. 12, no. 6, 2004, pp. 978–992.

[8] N. Duffield, J. Horowitz, and F. L. Presti, "Adaptive Multicast Topology Inference," in *in Proceedings of IEEE INFOCOM '01*, 2001, pp. 1636–1645.

[9] N. Duffield, F. L. Presti, V. Paxson, and D. Towsley, "Network loss tomography using striped unicast probes," vol. 14, no. 4, 2006, pp. 697–710.

[10] R. C. M. Coates and R. Nowak, "Maximum Likelihood Network Topology Identification from Edge-Based Unicast Measurements," June 2002.

[11] Y. Tsang, M. Yildiz, P. Barford, and R. Nowak, "Network radar: tomography from round trip time measurements," in *IMC '04: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, 2004, pp. 175–180.

[12] A. B. Kahn, "Topological sorting of large networks," in *Communications of the ACM*, vol. 5, 1962, pp. 558–562.

[13] M. Qiu, C. Xue, Z. Shao, Q. Zhuge, M. Liu, and E. Sha, "Efficent Algorithm of Energy Minimization for Heterogeneous Wireless Sensor Network," in *Embedded and Ubiquitous Computing, Lecture Notes in Computer Science*, 2006, pp. 25–34.

[14] J. Ni, H. Xie, S. Tatikonda, and Y. R. Yang, "Efficient and dynamic routing topology inference from end-to-end measurements," in *IEEE/ACM Transactions on Networking*, vol. 18, no. 1, February 2010, pp. 123–135.

[15] R. Castro, M. Coates, and R. Nowak, "Likelihood Based Hierarchical Clustering," in *IEEE Transactions on Signal Processing*, vol. 52, August 2004, pp. 2308–2321.

[16] D. Alderson and J. Doyle, "Toward an optimization-driven framework for designing and generating realistic internet topologies," in *In ACM HotNets-I*, 2002, pp. 41–46.

[17] P. Madadevan, C. Hubble, D. Krioukov, B. Huffaker, and A. Vahdat, "Orbis: Rescaling Degree Correlations to Generate Annotated Internet Topologies," in *Proceedings of ACM SIGCOMM '07*, Kyoto, Japan, August 2007.