Toward the Practical Use of Network Tomography for Internet Topology Discovery

Brian Eriksson
University of Wisconsin
Madison, WI
bceriksson@wisc.edu

Gautam Dasarathy
University of Wisconsin
Madison, WI
dasarathy@wisc.edu

Paul Barford
Nemean Networks and
University of Wisconsin
Madison, WI
pb@cs.wisc.edu

Robert Nowak
University of Wisconsin
Madison, WI
nowak@ece.wisc.edu

***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. In this paper, we describe new techniques that aim toward the practical use of tomographic inference for accurate router-level topology measurement. Specifically, prior tomographic techniques have required an infeasible number of probes for accurate, large scale topology recovery. We introduce a Depth-First Search (DFS) Ordering algorithm that clusters end host probe targets 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 DFS Ordering topology recovery algorithm in simulation and find that our method uses 94% fewer probes than exhaustive methods and 50% fewer than the current state-of-the-art. We also present results from a case study in the live Internet where we show that DFS Ordering 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.

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.

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]. While multicast inference is attractive due to total number of probes necessary (probing complexity) is $O(N)$ (where N is the number of end hosts in the topology), 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 [9], [10], however these techniques are also impractical due to the quadratic number of probes ($O\left(N^2\right)$) needed to resolve the topology. Many unicast tomography techniques also require significant, coordinated measurement infrastructure.

The tomographic technique we will focus on for this paper 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 1. Both packets originating from end host a will encounter 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 ($cov(d_b, d_c)$) will inform us to the amount of shared logical topology between paths $\{a, b\}$ and $\{a, c\}$.

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 include 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 number of probes that are required in order to resolve a topology.

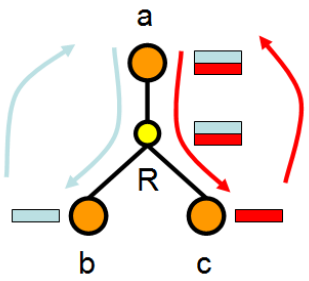

Fig. 1. Example of Network Radar on simple logical topology.

In this paper, we exploit the idea of arranging end host targets for RTT probes 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 end hosts (leaf nodes) of a tree structure 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 lower 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 will show how a DFS ordering clusters target end hosts based on the amount of shared infrastructure. Given this shared infrastructure clustering, we will demonstrate how the resulting covariance matrix has a special structure, and by exploiting this special covariance matrix structure the number of delay-based 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 over a factor of 2 using our new DFS Ordering-based methodology. To the best of our knowledge, our resulting probing complexity is the lowest probing complexity for any developed unicast tomography algorithm. We believe this reduction in the number of probes is an important step towards unicast tomography being considered a feasible and practical topology discovery mechanism.

The remainder of the paper is structured as follows. In Section II, we will describe previous delay-based tomographic methods for Internet logical topology discovery and other related work. Exploiting the Depth-First Search (DFS) of a tree, the idea of *DFS Ordering* is introduced in Section III with a description of its implications on topology reconstruction. In Section IV, an efficient logical topology discovery algorithm is described given an a priori *DFS Ordering* of the end hosts. Given that real world topologies will not have a proper a priori ordering known, in Section V we show how a valid DFS ordering of the end hosts can be found from relatively few topology measurements. Finally, in Section VI the results of our experiments on both synthetic and real world topologies will show the probe efficiency improvements of our procedure for estimating the logical topology of a network over previous results.

II. Related Work

The initial work most directly related to the research in this paper is the hierarchical clustering methodologies explored in [7], [8], [15], [16]. This method requires obtaining the entire covariance matrix (e.g., $O\left(N^2\right)$ measurements given N number of end hosts in the topology). The hierarchical clustering methodology will be considered the worst case probing bounds, as it performs an exhaustive probing on the set of end hosts in the network. This 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 hierarchical clustering, to $O\left(N\ell\log_\ell\left(N\right)\right)$ for the Sequential Topology Inference algorithm. In Section V, we will 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 will show how our methodology can further reduce the number of probes by roughly a factor of 2 compared to this current state-of-the-art.

III. Depth-First Search (DFS) Order

The foundation for the work in this paper is the idea of *Depth-First Search (DFS) Ordering*. 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 will 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 the logical tree structure ignoring the labeling of the internal nodes of the tree. In previous literature, this was considered a "topological sort" [12] on the leaf nodes of a tree structure.

For the tree structure in Figure 2, 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

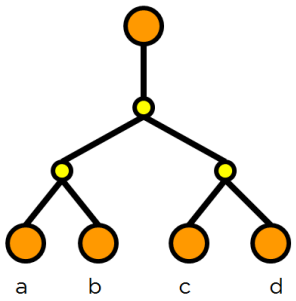

Fig. 2. Example simple logical topology in a proper DFS Order.

ordering $\{a, c, d, b\}$ does not satisfy a depth-first search of the end hosts.

The power of considering a depth-first search can be seen when examining the shared logical path matrix $\mathbf{S}$, where $S_{i,j}$ = 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. For a proper DFS ordering of the topology in Figure 2 ($\{a, b, c, d\}$), the matrix $\mathbf{S}_{proper}$ will be found as:

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

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

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

Using the intuition from this small example, we can state the following proposition that shows how a Depth-first Search ordering clusters end hosts based on the degree of shared infrastructure,

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

$$S_{i,i+j} \geq S_{i,i+k} \quad : \text{for } 0 \leq j \leq k$$

Proof - Consider the case where the end hosts are in a proper DFS ordering, but $S_{i,i+j} < S_{i,i+k}$ (for $0 \leq j \leq k$). 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). 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 proper DFS order. Given this proper ordering, we look to estimate the logical topology. Defining the unknown logical tree structure $\mathcal{T} = \{V, E\}$, and the router node path for end host x_i as $\mathbf{p}^{(i)} = \{v_1^{(i)}, v_2^{(i)}, ...\} \subset V$ the set of nodes from the root of the tree to end host x_i. We will denote the round-trip-time (RTT) delay variance along a single path as the sum of the delay variances of each router node along the path from the tree root to the end host,

$$\sigma_i^2 = \sigma^2\left(\mathbf{p}^{(i)}\right) = \sum_{j=1}^{\left|\mathbf{p}^{(i)}\right|} \sigma^2\left(v_j^{(i)}\right) \tag{1}$$

Where $\sigma^2(v)$ = the delay variance induced by router $v \in V$.

Using previous work on Network Radar in [11], we can state the covariance between two end hosts x_i, x_j,

$$\sigma_{i,j}^2 = cov\left(\mathbf{p}^{(\mathbf{i})}, \mathbf{p}^{(\mathbf{j})}\right) = \sigma^2\left(\mathbf{p}^{(i)} \bigcap \mathbf{p}^{(j)}\right) \tag{2}$$

is equivalent to the variance of the shared path from the root node to the two end hosts. Using this Network Radar probing technique, we obtain the delay covariance between any two end hosts x_i, x_j. We will define the covariance matrix Σ, such that $\Sigma_{i,j} = \text{cov}(x_i, x_j) = \sigma_{i,j}^2$. The first question we set about answering is, *Is there any inherent structure to a depth-first search ordered covariance matrix that can be exploited in order to efficiently estimate the logical topology?*

Using the ordering results from Proposition 1, we can state that the covariance matrix Σ has structure similar to the shared path matrix $\mathbf{S}$.

Proposition 2: Given the set of end hosts $\{x_1, x_2, ..., x_N\}$ in a proper DFS Ordering, the covariance matrix Σ will have the following property:

$$\sigma^2{}_{i,i+j} \geq \sigma^2{}_{i,i+k} \quad : \text{for } 0 \leq j \leq k$$

Proof - Given Proposition 1 and the fact that every router will induce positive delay variance, it is trivial to see this property of covariance matrix Σ.

The Hierarchical Clustering algorithm [7], [8] showed that in order to reconstruct the tree topology, the only information needed for each end host is the knowledge of which other end host, out of all the other end hosts, this end host has the most shared topology with. This is equivalent to finding the end host with the largest covariance magnitude. Unfortunately, to acquire this knowledge, it was previously necessary to obtain all possible covariance values. Given the end host DFS ordering assumption and the ordered covariance matrix structure as specified in Proposition 2, we will state that the

only covariance 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 covariance ($\sigma^2_{i-1,i}$) and the immediately successive end host covariance ($\sigma^2_{i,i+1}$). This is due to the proposition stating that the covariance $\sigma^2_{i,i+1} \geq \sigma^2_{i,i+j}$ 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 covariance 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 covariance values ($\sigma^2_{i-1,i}, \sigma^2_{i,i+1}$) will directly inform us as to the structure of the logical topology.

In order to distinguish between covariance differences caused by differences in topology and covariance differences induced by noise, we introduce the value δ here to denote the smallest possible delay covariance induced by a router in the topology.[1] We can now state the following proposition:

Proposition 3: Using the set of end hosts in a proper DFS Order, only $N-1$ pair probes (the covariance values $\sigma^2_{i,i+1}$ for $i = \{1, 2, ..., N-1\}$) are needed to reconstruct the unknown logical topology.

Proof: We will now show how every covariance magnitude combination will inform our reconstruction of the logical topology. Each of the following cases can be found in Algorithm 1. In constructing the tree topology, we will denote $f(x_i)$ as the assigned parent node of end host x_i.

A. Case A - $\left|\sigma^2_{i,i-1} - \sigma^2_{i-1,i-2}\right| < \delta$

Given a non-significant magnitude difference between the two covariance values, this implies that the shared path between the pairs $\{x_i, x_{i-1}\}$ and $\{x_{i-1}, x_{i-2}\}$ are the exact same set of routers. Therefore, as seen in Figure 3-(A), we can infer that the last logical hop is shared for the set of end hosts $\{x_i, x_{i-1}, x_{i-2}\}$. This assigns $f(x_i) = f(x_{i-1})$, the parent node of the current end host $f(x_i)$ is the same as the parent of the previous end host $f(x_{i-1})$.

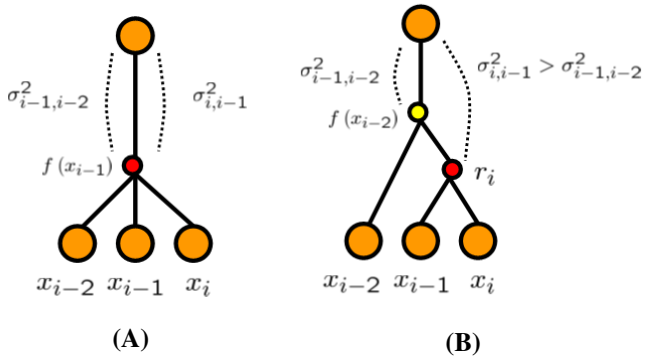

Fig. 3. **(A)** - Case A - $\left|\sigma^2_{i,i-1} - \sigma^2_{i-1,i-2}\right| < \delta$. The current end host x_i is attached to the parent of x_{i-1}. **(B)** - Case B - $\sigma^2_{i,i-1} \geq \sigma^2_{i-1,i-2} + \delta$. A new router r_i is created with children x_i, x_{i-1} with parent $f(x_{i-2})$.

[1]In a real-world experiment, we could find this term by cross-validation on a partition of known infrastructure. Assume here that it is known a priori.

B. Case B - $\sigma^2_{i,i-1} \geq \sigma^2_{i-1,i-2} + \delta$

This implies that there is more shared topology between the end host pair $\{x_i, x_{i-1}\}$ than the pair $\{x_{i-1}, x_{i-2}\}$. We will then insert a new interior logical router node, r_i, with children $\{x_i, x_{i-1}\}$ (this assigns $f(x_i) = f(x_{i-1}) = r_i$), with the new router r_i having the same parent as x_{i-2} (thereby assigning $f(r_i) = f(x_{i-2})$). The covariance value associated with the shared path to the new logical node, $\sigma^2_{r_i}$ must be recorded for future reference. An example of this structure can be see in Figure 3-(B).

C. Case C - $\sigma^2_{i,i-1} + \delta < \sigma^2_{i-1,i-2}$

In the case that the current covariance pair, $\sigma^2_{i,i-1}$, is less than the previous covariance pair, $\sigma^2_{i-1,i-2}$, this implies that end host x_i attaches to a logical router at some point in the topology higher in the tree than the current parent router ($f(x_{i-1})$) attached to end host x_{i-1}. But which logical router should it attach to? To find this router, we must traverse the current logical path from x_{i-1} to the root node (the set of nodes $\{f(x_{i-1}), f(f(x_{i-1})), f(f(f(x_{i-1}))), ...\}$) and discover the farthest logical router (r^*) from the end host (i.e., the router on the path closest to the root node) that has recorded covariance greater than or equal to the current covariance pair $\sigma^2_{r^*} \geq \sigma^2_{i,i-1}$. Once this logical router r^* is found, one of the following cases will occur.

1) Case C-1 - $\left|\sigma^2_{r^*} - \sigma^2_{i,i-1}\right| < \delta$: There is a non-significant difference between the covariance of the found router and the observed covariance. Simply assign the current end host x_i as having the parent r^* ($f(x_i) = r^*$) as seen in Figure 4-(A).

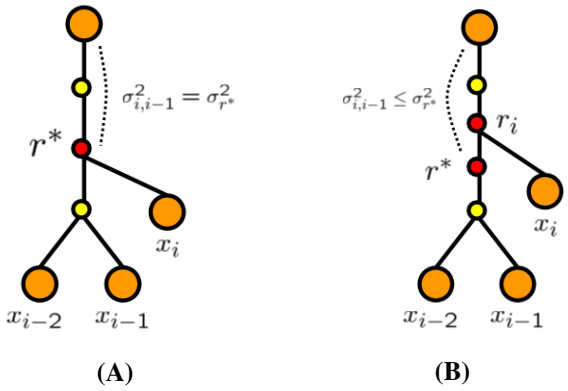

Fig. 4. **(A)** - Case C-1 - $\left|\sigma^2_{r^*} - \sigma^2_{i,i-1}\right| < \delta$. The current end host ($x_i$) is attached to router r^*. **(B)** - Case C-2 - $\sigma^2_{r^*} < \sigma^2_{i,i-1} + \delta$. A new router r_i is attached on the path between routers r^* and $f(r^*)$.

2) Case C-2 - $\sigma^2_{r^*} \geq \sigma^2_{i,i-1} + \delta$: This implies that there was a previously unseen logical router on the path between the router r^* and its parent node $f(r^*)$. We must add a new logical router r_i between these two nodes, such that, $f(r_i) = f(r^*)$ and $f(r^*) = r_i$. Finally, we attach the current end host x_i to the new router r_i, setting $f(x_i) = r_i$. An example of this is seen in Figure 4-(B). ■

Algorithm 1 - Ordered Logical Topology Discovery Algorithm

Initialize:

1) Given set of end hosts $(x_1, x_2, ..., x_N)$ in a proper DFS Order with unknown logical topology
2) δ = minimum possible delay covariance induced by a single router.
3) Initial reconstructed topology - $\widehat{\mathcal{T}} = (V, E)$.
4) Initial set of nodes - $V = \{r_1, x_1, x_2\}$.
5) Initial set of edges - $E = \{(r_1, x_1), (r_1, x_2)\}$

Main Body: For $i = \{3, 4, ..., N\}$.
Add the new end host x_i to the set of nodes in the reconstructed topology, $V = V \cup \{x_i\}$.

if $\left|\sigma^2_{i-1,i-2} - \sigma^2_{i,i-1}\right| < \delta$ **then**

1) Assign the parent of x_{i-1} to be the same parent as the current end host x_i, $E = E \cup \{(f(x_{i-1}), x_i)\}$.

else if $\sigma^2_{i-1,i-2} > \sigma^2_{i,i-1}$ **then**

1) Create new node r_i, $V = V \cup \{r_i\}$.
2) Set r_i as a child of the current parent $f(x_{i-1})$, $E = E \cup \{r_i, f(x_{i-1})\}$.
3) Remove the previous edge between x_{i-1} and the assigned parent, $E = E/\{f(x_{i-1}), x_{i-1}\}$.
4) Assign r_i as the parent of the end hosts x_i, x_{i-1}, $E = E \cup \{(r_i, x_i), (r_i, x_{i-1})\}$
5) Record the current covariance value for future reference $\sigma^2_{r_i} = \sigma^2_{i,i-1}$.

else

Find the parent router of x_{i-1} (denoted as r^*) such that $\sigma^2_{r^*} \geq \sigma^2_{i,i-1}$.

if $\left|\sigma^2_{r^*} - \sigma^2_{i,i-1}\right| < \delta$ **then**

1) Assign r^* as the parent of x_i, therefore $E = E \cup \{(r^*, x_i)\}$

else if $\sigma^2_{r^*} > \sigma^2_{i,i-1}$ **then**

1) Create new node r_i, $V = V \cup \{r_i\}$.
2) Place new node r_i between router r^* and its parent $f(r^*)$. First remove the current link $E = E/\{f(r^*), r^*\}$, then add the new edges, $E = E \cup (\{f(r^*), r_i\}, \{r_i, r^*\})$
3) Record the current covariance value for future reference $\sigma^2_{r^*} = \sigma^2_{i,i-1}$.

end if

end if

V. Depth-First Search Ordering Estimation

The major problem with the methodology in Section IV is that it is based around the assumption that the end hosts are already correctly arranged in a proper depth-first search order. In any non-trivial problem, this ordering will not be known. Instead, given no a priori knowledge of the topology, we must estimate a proper DFS Ordering from targeted measurements. *But how can we infer this ordering using as few targeted probes as possible?*

Given a random ordering of the set of end hosts, consider choosing a single end host (x_1) and obtaining the delay covariance between this end host and all other end hosts in the set ($= \{\sigma^2_{1,2}, \sigma^2_{1,3}, \sigma^2_{1,4}, ..., \sigma^2_{1,N}\}$). Some end hosts will have very high delay covariance, while others will have significantly less shared infrastructure with the chosen end host and therefore have low delay covariance. Consider sorting these obtained covariance 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. Can this be considered a proper DFS Ordering? No, as seen in Figure 5, a significant fraction of the end hosts will have the same observed delay covariance (in this case, σ^2_A) when compared against the chosen end host. While the end hosts with the same covariance values will be clustered together in this ordering, a proper DFS order inside this cluster is unknown using only this single vantage point.

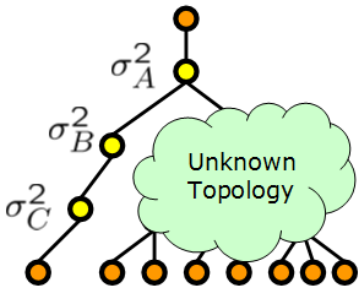

Fig. 5. Example of covariance values from a single end host not revealing the entire topology

This implies that delay covariances from more than a single end host vantage point will be required to correctly order the entire set of end hosts. But how many vantage points will be required? For Figure 5, consider that any covariance will take one of three values $\{(\sigma^2_A), (\sigma^2_A + \sigma^2_B), (\sigma^2_A + \sigma^2_B + \sigma^2_C)\}$. Having correctly ordered the end hosts by these three values, we now desire to order the subclusters (e.g., what should the ordering be of all the end hosts with covariance $= \sigma^2_A$ for the topology?). One could consider dividing the set of end hosts into constant covariance clusters (e.g., all the end hosts with covariance σ^2_A in one cluster, all the end hosts with covariance $\sigma^2_A + \sigma^2_B$ in another cluster, etc.) 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 delay covariance values with this new vantage point. While we could consider this methodology in a noise-free environment, when noise is present this division of the end hosts into multiple clusters introduces multiple possibilities for cluster misclassification error, thereby introducing error into the topology reconstruction.

Instead, we will look to a recursive methodology that at each iteration bisects the ordered set of end hosts into only two clusters. This reduces our objective to the single problem of finding the correct end host to bisect the set at each iteration of the algorithm. The simplest approach to this problem is

for a given value δ (where each router will induce at least δ delay covariance), sorting the covariance values and finding all the possible bisection candidate end hosts (denoted by set $\mathcal{I}$) where $i \in \mathcal{I}$ if the difference between the i-th and the $(i+1)$-th covariance value is more than δ,

$$\mathcal{I} = \{i : \sigma^2_{1,i+1} - \sigma^2_{1,i} \geq \delta\}$$

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

$$i^* = \arg\min_{i \in \mathcal{I}} \left| i - \frac{N}{2} \right| \tag{3}$$

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

Algorithm 2 - Bisection DFS Ordering Algorithm - bisect($\mathbf{X}$, δ)

Given:

1) Unordered set of end hosts with unknown logical topology $\mathbf{X}$
2) δ = minimum possible covariance induced by a single router

Main Body:

1) Find $\mathbf{Y}$, such that $Y_i = cov(X_1, X_i) = \sigma^2_{1,i}$ for $i = \{2, 3, ..., |\mathbf{X}| - 1\}$.
2) Sort the covariance vector $\mathbf{Y}$, obtaining the ordered index vector $\mathbf{I}$
3) Find $\mathbf{I}_\delta = \{i : \mathbf{Y}(\mathbf{I}(i+1)) - \mathbf{Y}(\mathbf{I}(i)) > \delta\}$, the indices where the difference between consecutive sorted covariance values is greater than δ.
4) Bisect the set of sorted end hosts $\mathbf{X}$ at the index of $\mathbf{I}_\delta$ that creates two sets most equal in size using Equation 3, creating sorted end host subsets $\mathbf{X}_1, \mathbf{X}_2$.
5) If $|\mathbf{X_1}| > 2$, then find $\mathbf{I}_1$ = bisect($\mathbf{X_1}$, δ)
6) If $|\mathbf{X_2}| > 2$, then find $\mathbf{I}_2$ = bisect($\mathbf{X_2}$, δ)
7) Reorder $\mathbf{X}_1$ using new indices $\mathbf{I}_1$.
8) Reorder $\mathbf{X}_2$ using new indices $\mathbf{I}_2$.
9) Return the final ordered list of indices $\mathbf{I} = [\mathbf{I}_1 \mathbf{I}_2]$

Proposition 4: Using Algorithm 2, the number of probes needed to correctly obtain a proper DFS Ordering for a balanced ℓ-ary tree (where each non-leaf node has ℓ children) with N end hosts is upper bounded by $p(\ell) N \log_\ell N$ probe pairs (where $p(\ell)$ is sublinear in ℓ).

Proof: Using Algorithm 2, consider the first step, end host x_1 will be chosen and the covariance values will be found between x_1 and $x_2, x_3, ..., x_N$. Given the ℓ-ary balanced property of the tree, after sorting the covariance 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 6-(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 $\frac{(\ell-1)N}{\ell} - 1$ covariance 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 6-(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.

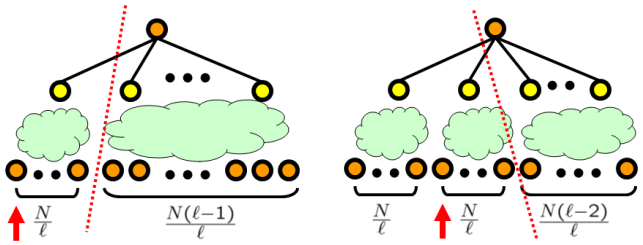

Fig. 6. **(Left)** The first split taken on a balanced ℓ-ary tree. **(Right)** The second split taken on a balanced ℓ-ary tree. Both splits indicated by the dotted line, the arrow indicates the randomly chosen end host covariance values are measured against.

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 $f_\ell(N)$.

$$\begin{aligned} f_\ell(N) &\leq N + \frac{\ell-1}{\ell}N + \frac{\ell-2}{\ell}N + \cdots + \frac{2}{\ell}N + \ell f_\ell\left(\frac{N}{\ell}\right) \\ &= \frac{N}{\ell}(2 + 3 + \cdots + \ell) + \ell f_\ell\left(\frac{N}{\ell}\right) \\ &\overset{(a)}{=} Np(\ell) + \ell f_\ell\left(\frac{N}{\ell}\right) \\ &\overset{(b)}{\leq} Np(\ell) + \ell\left\{\frac{N}{\ell}p(\ell) + \ell f_\ell\left(\frac{N}{\ell^2}\right)\right\} \\ &= 2Np(\ell) + \ell^2 f_\ell\left(\frac{N}{\ell^2}\right) \\ &\vdots \\ &\overset{(c)}{\leq} p(\ell)N(\log_\ell N) \end{aligned} \tag{4}$$

(5)

Where

$$p(\ell) := \frac{(2 + 3 + \cdots + \ell)}{\ell} = \left(\frac{\ell+1}{2} - \frac{1}{\ell}\right) \tag{6}$$

■

Given that once the ordering is found (using Algorithm 2), to resolve the logical topology only requires an additional N pair probes (using Algorithm 1). By intelligent bookkeeping

of covariance values in Algorithm 2, it is possible to obtain knowledge of these values without addition probing. It is then trivial to prove Proposition 5 for the total number of probes required by our new DFS Ordering Algorithm. Therefore using our new DFS Ordering methodology, we are using roughly half the probes needed by the current state-of-the-art tomography approach in [14] that would require at most $N\ell\left(\log_\ell N\right)$ pair probes.

Proposition 5: Using the DFS Ordering Algorithm, the number of probes needed to correctly obtain the logical topology from a balanced ℓ-ary tree (each non-leaf node has ℓ children) with N end hosts is upper bounded by $Np(\ell)\log_\ell N$ (where $p(\ell) = \left(\frac{\ell+1}{2} - \frac{1}{\ell}\right)$ is sublinear in ℓ).

VI. Experiments

A. Prior Methods

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

2) Sequential Logical Topology: What happens when acquiring all $\frac{N(N-1)}{2}$ covariance values is infeasible? 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. This methodology depends upon sequentially building the tree topology for each end host. For a given end host, the delay covariance 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 covariance (and thus the most shared topology), c_i^*, the delay covariance is found between the end host and the children of the specified child (c_i^*). The covariance 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 delay covariance 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 probes needed is upper bounded by $\ell N\log_\ell N$.

A comparison of the probing complexity for all three probing methodologies (hierarchical clustering, sequential, DFS ordering) is seen in Table I. The new DFS Ordering algorithm is found to have the smallest probing complexity upper bound of the three algorithms. In comparison with the Sequential Topology algorithm, from Equation 6 we can see that $\frac{p(\ell)}{\ell} \leq 0.5625$ for all choices of ℓ, therefore the new DFS Ordering will use at most 56.25% of the pairs probes needed by the Sequential Topology algorithm.

TABLE I
Upper bound probing complexity for the three probing methodologies for balanced ℓ-ary tree. (Where $p(\ell)$ is sublinear in ℓ)

Methodology	Probing Complexity
Hierarchical Clustering	$\frac{1}{2}N(N-1)$
Sequential	$\ell N\log_\ell N$
DFS Ordering	$p(\ell)N\log_\ell N$

B. Datasets

1) Synthetic Dataset: In this paper, the synthetic topologies are generated by Orbis [17]. Orbis is one of the latest and most realistic network topology generators. It creates graphs that have properties that are consistent with many of those observed in the Internet. The Orbis-generated synthetic networks enable us to analyze the capabilities of our methods with full ground truth and over a range of network and embedding sizes. For these experiments we will consider three different sized topologies, $N = \{768, 1497, 2261\}$.

2) Real World Data: 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 7 starting at the University of Wisconsin - Madison as the root node. Using the Network Radar methodology [11], the sample covariances were found between pairs of the 9 end hosts in the topology.

C. Synthetic Noise-Free Experiments

To test the performance of our algorithm in a noise-free environment, several different sized Orbis topologies were generated. With each topology, every node was assigned a random covariance value, with the estimated measured covariance being the sum of the random covariance values (with the smallest router covariance assigned, $\delta = 0.1$) along the shared shortest path from the root node to the two end hosts under consideration. In Table II, we present the resulting number of probes required to resolve the logical topology for both the new DFS Ordering methodology, the previous state-of-the-art Sequential Inference algorithm, and the exhaustive Hierarchical Clustering method. Due to this experiment being noise-free, all methodologies will perfectly reconstruct the topology. As seen in the table, the DFS Ordering methodology does significantly better than the exhaustive Hierarchical Clustering approach, with over 94% fewer probes needed to resolve the topology. In comparison with the more recent Sequential Inference algorithm, from these experiments it was seen that our new method requires on average 50% fewer probes to resolve the topology, matching the derived bounds in Table I.

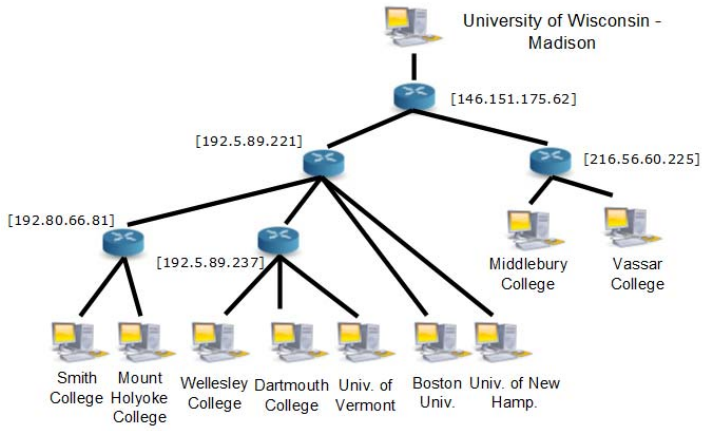

Fig. 7. Real world topology used to test tomography methods

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

	Hierarchical Algorithm	Sequential Algorithm		DFS Ordering Algorithm		
Number of End Hosts (N)	Probe Pairs Needed	Probe Pairs Needed	Percentage of Hierarchical Pairs	Probe Pairs Needed	Percentage of Hierarchical Probes	Percentage of Sequential Pairs
768	294,528	38,029	12.91%	16,768	5.69%	44.09%
1,497	1,119,756	127,750	11.41%	63,036	5.63%	49.34%
2,261	2,554,930	242,656	9.50%	135,531	5.30%	55.85%

D. Real World Experiments

Using the Network Radar methodology [11], we observed 1,000 back-to-back round-trip-time delay samples for every end host pair in our real-world topology (Figure 7). Due to imperfect round-trip-time measurements and other delay noise measured, the sample covariance was found to not be perfectly correlated to the `traceroute` observed shared path length. Therefore, for any estimation procedure based on the sample covariance, there will be potential errors in the reconstructed topology [2]. In order to determine the accuracy of our estimated topologies, we must develop a metric that compares our estimated topologies to the ground-truth topologies.

1) Shortest Path Estimation: Consider a triple of end hosts $\{a, b, c\}$ that exist 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 value can be found by,

$$p = \frac{1}{|\mathbf{X}|^3} \sum_{i \in \mathbf{X}} \sum_{j \in \mathbf{X}} \sum_{k \in \mathbf{X}} f(i, j, k)$$

Where the value $f(i, j, k) = 1$ if the reconstructed topology correctly classifies the triple $\{i, j, k\}$ and $f(i, j, k) = 0$ if the reconstructed topology incorrectly classifies the triple $\{i, j, k\}$:

$$f(i, j, k) = \mathcal{I}(\widehat{P}(i, j) \geq \widehat{P}(i, k))\mathcal{I}(P(i, j) \geq P(i, k)) + \mathcal{I}(\widehat{P}(i, j) < \widehat{P}(i, k))\mathcal{I}(P(i, j) < P(i, k))$$

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

The baseline for any topology reconstruction algorithm will be to outperform a naive random choice of the two possible triple combinations ($\{a, b\}$ greater than $\{a, c\}$, $\{a, b\}$ less than or equal to $\{a, c\}$), this would be roughly equivalent to $p = \frac{1}{2}$ (given equal distribution of the two cases in the topology). This value p will be the metric we use to assess how accurate the estimated topologies are.

2) Results: The performance of the three algorithms are averaged over 2,000 random permutations of the end hosts. Both the Sequential Algorithm and the new DFS Ordering algorithm have performance sensitive to initial choice of end hosts. Averaging over many random permutations eliminates any order bias from the results.

All three algorithms have a tunable parameter, δ, that must be chosen. To give the two comparison methodologies (sequential and hierarchical clustering) every possible advantage, for each restricted number of probes, the performance of

[2]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.

both algorithms is shown for *the best possible value of* δ at each level of probing. Meanwhile, our new DFS ordering methodology has a constant value of δ across all levels of probing.

For the real-world topology in Figure 7, the corresponding p values for the three topology reconstruction algorithms (DFS Ordering, Sequential, Hierarchical Clustering) can be seen in Figure 8 versus a restricted total number of delay probes available. All three topology reconstruction techniques do significantly better than the naive random classification technique (with performance $p = \frac{1}{2}$). As shown in the figure, the new DFS Ordering algorithm will reconstruct the the real-world topology with the highest accuracy for a restricted number of tomography probes, with improvements of several percent accuracy in classifying shared path lengths over all range of probing sizes. Given the probing complexity in Table I, it is very likely that the accuracy improvements for the new DFS Ordering algorithm will grow as the size of the topology increases. We leave extending these results from this small-scale Internet topology experiment to a large-scale Internet topology experiment as future work.

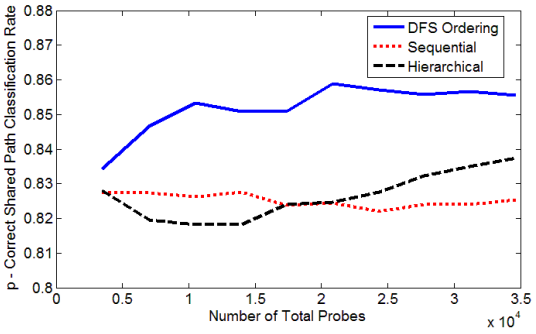

Fig. 8. Topology reconstruction results for the three algorithms (DFS Ordering, Sequential, and Hierarchical Clustering).

VII. 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 quadratic number of probes. In this paper, we describe algorithms that considerably reduce the number of probes needed to resolve logical topology. The ability to reduce the number of 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 improvements of over 94% fewer probes compared with an exhaustive methodology, and roughly 50% fewer probes compared with 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. In future work, we look to applying this methodology to a larger real-world environment and developing an integrated `traceroute`/tomographic algorithm that aims to eliminate undiscoverable portions of Internet maps.

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 *To appear in IEEE/ACM Transactions on Networking*, 2009.

[15] M. Shih and A. Hero, "Hierarchical inference of unicast network topologies based on end-to-end measurements," in *IEEE Transactions on Signal Processing*, vol. 55, 2007, pp. 1708–1718.

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

[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.