

Estimating Hop Distance Between Arbitrary Host Pairs

Brian Eriksson
University of Wisconsin
Madison, WI
bceriksson@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—Establishing a clear and timely picture of Internet topology is complicated by many factors including the vast size and dynamic nature of the infrastructure. In this paper, we describe a methodology for estimating an important characteristic of Internet topology - the hop distance between arbitrary pairs of end hosts. Our goal is to develop an approach to pairwise hop distance estimation that is accurate, scalable, timely and does not require a significant measurement infrastructure. Our methodology is based on deploying a small set of landmark nodes that use traceroute-like probes between each other to establish a set of accurate pairwise hop distances. The landmark nodes are also configured to collect source IP addresses and TTL values from passively monitored network packet traffic. We develop a novel multidimensional scaling algorithm that can be applied to both the passive and active measurements to generate pairwise hop distance estimates for all of the observed source host addresses. The basic algorithm is then enhanced to consider the autonomous system membership of source hosts via BGP routing information. We investigate the capabilities of our estimation algorithms using a set of synthetic network topologies. The results show that our method can generate highly accurate pairwise hop distance estimates over a range of network sizes and configurations, and landmark infrastructure sizes.

I. INTRODUCTION

Over the past quarter century, the Internet has grown into a gigantic and fantastically complex infrastructure that connects over a billion users world wide. The ability to measure, map and analyze the structural characteristics of this artifact accurately would, for example, facilitate network design and network management processes by exposing potential weaknesses in the infrastructure and opportunities to improve its robustness and performance. Similarly, the ability to generate accurate maps of the Internet or portions of the Internet in near real time could help in the process of diagnosing anomalous events, or in constructing and maintaining overlay infrastructures among other things.

There are significant challenges in any approach to measurement and characterization of Internet topology. First, the lack of inherent support for topology measurement coupled with ISPs desire to keep much of this information private calls for a distributed measurement infrastructure and structural inference

methods that are reliable and robust. Next, the vast size and global footprint of the Internet suggest that a potentially significant number of measurement systems will be required in order to gather sufficient data to generate comprehensive maps. Finally, the well known dynamic nature of the Internet means that measurements must be taken almost continuously in order to identify changes in a timely fashion.

Most prior work on measuring Internet topology has been based on using active probe-based tools similar to `traceroute` to gather data. Great strides have been made in solving the thorny problems associated with active probe-based topology measurement (*e.g.*, interface disambiguation [1], [2]), which has enabled accurate mapping of ISP topologies (*e.g.*, [2]) and of the Internet's core (*e.g.*, [3], [4]). However, the active probe-based approaches are inherently limited in their ability to capture characteristics of the edges of the network and by the fact that they introduce an additional traffic load on the infrastructure.

In our prior work, we described a new approach to identifying the Internet's structural characteristics based on simple *passive measurements* of end host address¹ and TTL information from packets observed on a link or opportunistically gathered *e.g.*, via network honeypots [5], [6]. We developed a set of inference algorithms that can be used to establish end host address clusters and to identify shared path characteristics sufficient to recover important characteristics of network topology. The passive measurement-based approach offers the benefits of no additional traffic in the network, simplified management of the measurement infrastructure, and the potential for broader and more timely coverage.

In this paper, we extend our prior work by describing a new method for accurately estimating the shortest hop distances between *arbitrary pairs* of hosts in a network. We argue that this characteristic of Internet topology is important and useful in its own right. For example, if link failures are independently and identically distributed, then identifying the shortest paths between nodes is important to robust overlay network design. The shortest pairwise hop distance estimation between arbitrary nodes is also an important step toward our

This work was supported in part by NSF grants CNS-0347252, CNS-0646256, CNS-0627102 and CCR-0355653. 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.

¹We assumed IP addresses to be equivalent to individual Internet hosts although we understand that this could introduce error into our estimates.

overall objective of accurate and timely generation of Internet-wide topology maps using passive measurements. While it is known that routes do not always take the shortest path [7], we argue that the assumption is reasonable for this work since our primary objective is to demonstrate the feasibility of our methods, and not to draw conclusions about the empirical characteristics of the Internet.

We address this problem by creating an embedding of m landmark nodes, which send active traceroute-like probes between each other. These nodes are also configured to collect passive measurements of $n \gg m$ source IP addresses and TTL values. The difficulty is that hop counts between any pair of the n nodes are not available and packets from all n nodes will not be observed at all m landmark nodes. We apply a multi-dimensional scaling (MDS) algorithm to generate estimates of hop distances between all nodes using the combined active and passive measurements. This algorithm uses an iterative approach to minimize the error between the embedding of the nodes and observed distances from the passive and active measurements. We emphasize the fact that the number of landmarks is far smaller than the number of end host addresses ($m \ll n$). In our experiments, m is orders of magnitude smaller than n . Thus, the overhead of the active measurement component of the process is almost negligible.

We evaluate the feasibility and capabilities of our pairwise hop distance estimation method using a set of synthetic topologies and an empirical data set from Skitter [4]. We consider varying degrees of missing passive measurement data and find that the algorithm is able to consistently produce highly accurate pairwise hop distance estimates. We also consider the impact of different sizes of landmark infrastructures and find that even low dimensional embeddings (*e.g.*, < 7) are able to produce highly accurate estimates.

The remainder of this paper is organized as follows. In Section II, we discuss prior studies that are related to our own. In Section III, we describe the datasets that are used in the evaluation of our pairwise hop distance estimation method. We provide details on the multi-dimensional scaling estimation method using active measurements in Section IV, and on our extension to this method using incomplete passive measurements in Section V and using BGP information in Section VI. We demonstrate the capabilities of our methods in Section VII, and we conclude in Section VIII.

II. RELATED WORK

Most previous network embedding methods have considered the problem of latency estimation between nodes. In [8], [9], [10], [11], methods are proposed in which a set of m landmark nodes are embedded in a low-dimensional Euclidean space, and then mn measurements are made between each landmark node and all n other nodes. While past studies have identified difficulties with some of the basic assumptions of embeddings (*e.g.*, [7]) more recent work has shown them to perform quite well in practice [12]. Embeddings have also been proposed as a mechanism for topological inference [13], [14]. These approaches are based on hop-count measurements obtained

using active probes between landmarks and all other nodes. In contrast, our proposed approach relies primarily on passive measurements and involves a negligible number of active probes, resulting in a significantly lighter weight approach to the problem. Our emphasis on passively collected data avoids the problems of using a large number of active measurements, which includes the difficulty in generating real-time Internet topologies from these measurements and the prevalence of active blocking of standard probes by ISPs.

To the best of our knowledge this is the first network embedding method that uses passive measurements to reduce the number of active probes. The total number of active probes needed for our method will be shown to grow quadratically in the embedding dimension. As the embedding dimension is shown to be typically about 10 (see Section VII), the number of probes needed will be on the order of around 100 (regardless of the number of nodes in the network), making the method almost completely dependent on passive measurements.

This is also the first network embedding study designed to handle large amounts of missing measurement data. Our embedding methodology, unlike the prior work in [14], [13], [8], is designed to embed IP sources given very incomplete network measurements. In fact, due to their reliance on complete measurements, the previous work in the area of network embedding is incomparable to our methodology. Our previous work in the area of incomplete network measurements [5], [6] focused on statistical clustering of end hosts given incomplete passive measurement data. This paper describes a more advanced methodology that resolves the distance in router hops between each pair of end hosts, a problem not addressed in our previous research.

III. HOP-COUNT DISTANCE MEASUREMENTS

The basis for our approach is illustrated by the diagram in Figure 1. Packets sent from source S_i depart from the network edge and eventually enter the densely-connected core component through a border router. The packets then traverse the core, exit through another border router and eventually are intercepted by a passive monitor M_j . The number of hops between S_i and M_j , deduced from the TTL field, is denoted by h_{ij} . A complete system with n such end hosts and m passive monitors results in an $n \times m$ hop-count matrix denoted by H . In practice, not all end hosts will be observed at every monitor, and therefore H is typically incomplete (*i.e.*, $h_{i,j}$ is missing for certain i, j). We describe a procedure for dealing with this incomplete data problem later in the paper. We also assume that the complete set of hop-count distances (an $m \times m$ matrix denoted by A) between all pairs of monitors is available. These distances may be obtained either passively or actively.

IV. MULTIDIMENSIONAL SCALING (MDS)

Several previous studies have indicated that the salient features (*e.g.*, latencies or hop-count distances) of high-dimensional networks can be captured by low-dimensional embeddings [8], [9], [10], [13], [14]. We pursue this idea by determining a low-dimensional embedding that preserves

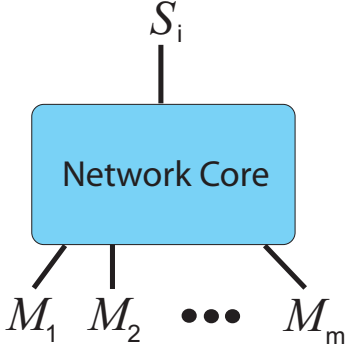


Fig. 1. Basic network model consists of passive monitors M_1, M_2, \dots, M_m that capture packets sent by end host S_i through a densely connected core component.

the distances of the observed hop-counts $h_{i,j}$ as accurately as possible. In turn, the embedding produces estimates (or imputations) of the hop-count distances between sources and monitors that were not observed and between n sources themselves (also not observed).

A. Classical Multidimensional Scaling (MDS)

First consider the case where n^2 active measurements have been performed between all pairs of the n end hosts in the network. This provides the hop-count distances between all pairs of end hosts, creating the distance matrix D (where $D_{i,j}$ is the number of hops between end hosts S_i and S_j). Using the complete observation of pairwise distances, one can apply classical Multidimensional Scaling (MDS) [15] to map the sources to a lower dimensional space ($S_i \rightarrow X_i$). This lower dimensional space will be of dimension $= d$, where $d \ll n$. In this mapped space, the Euclidean distance between points ($\|X_i - X_j\|$) corresponds to the estimated hop-count distances between the two end hosts in the network.

The embedding dimension d is chosen to be as small as possible while still providing an embedding that accurately preserves the hop-count distances. This is accomplished by considering the energy ratio

$$e(d) = \frac{\sum_{i=1}^d \lambda_i}{\sum_{i=1}^m \lambda_i} \quad (1)$$

where λ_i is the i -th eigenvalue of the matrix A . The embedding dimension d is selected as the smallest d such that $e(d) > 90\%$. This can be considered as the smallest embedding dimension, such that 90% of the energy of the original high-dimensional data is retained. Motivation for this threshold of 90% will be shown in Section VII-D.

B. Landmark MDS using Active Measurements

Given a very large number of end hosts, the computational complexity of classical MDS may be too large to practically implement (given the $O(n^3)$ computation time and $O(n^2)$ number of active probes needed). This challenge can be addressed using a technique known as Landmark MDS [16]. Landmark MDS reduces the complexity by measuring the

distances between each end host to only m landmark points, and then embedding each end host using knowledge of the pairwise distances between each one of the landmarks (*i.e.*, the m monitors in our application). This method reduces the computational complexity from $O(n^3)$ to $O(m^2n + m^3)$. In a network setting, this requires making m^2 active measurements to obtain the pairwise distances between the monitors, and then mn active measurements to obtain the distances from each source to each of the landmarks. The GNP method from [8] can be considered as a special case of Landmark MDS with active measurements.

V. LANDMARK MDS USING INCOMPLETE PASSIVE MEASUREMENTS

Active probing to determine the hop-count distances between end hosts and monitors is impractical. The virtue of our proposed framework is that some of these distances can be obtained passively, this eliminates the requirement of mn active measurements between each IP source and each landmark. Note that if we had a complete set of passive measurements from each of the n end hosts to all m number of monitors, it would be possible to simply use Landmark MDS to embed both monitors and end hosts in a lower-dimensional space. The challenge of the passive approach is that many of the distances will not be observed, resulting in an incomplete data problem. Thus, for a given set of monitors, many of hop count distance observations are missing from the observed set. For each end host S_i , we have a (potentially incomplete) hop count vector $h_{i,I^{(i)}}$ where $I^{(i)}$ indicates the indices from the passive monitors that have observed traffic from end host S_i . We assume that this data is Missing-at-Random (where the missing data locations are chosen at random in each vector)².

The challenge becomes how to determine the embedding from incomplete hop-count data. Previous work on Landmark MDS with missing data in [17] simply removes any data point with missing features. In our application, this simple-minded approach would be catastrophic, since it is unlikely that many end hosts will be observed at all m monitors. We next adapt a method previously designed to handle noisy distance observations to handle the missing data problem.

A. Stress Function Construction

Given A , the array of pairwise hop-count distances between passive monitors, H the (potentially incomplete) hop-count distances from the end hosts to the monitors, we can then construct the $(n+m) \times (n+m)$ symmetric distance matrix D , where each element represents the observed pairwise distance between two end hosts. This matrix can be represented as follows:

$$D = \begin{bmatrix} A & H^T \\ \emptyset_1 & \emptyset_2 \end{bmatrix}$$

where \emptyset_1 represents the unknown $n \times m$ reverse hop count matrix (the number of hops from the monitors to the end hosts)

²Clearly, this Missing-at-random assumption is reliant on monitor placement in real topologies

and \emptyset_2 denotes the $n \times n$ array of missing hop-count distances between pairs of the n end hosts.

Next define a mask W to indicate the locations of the observed hop values:

$$W_{i,j} = 1 \quad \text{for } 1 \leq i, j \leq m$$

$$W_{i,j} = \begin{cases} 1 : & \text{if } h_{i,j} \text{ is known} \\ 0 : & \text{if } h_{i,j} \text{ is missing} \end{cases}$$

For $m+1 \leq i \leq m+n$ and $1 \leq j \leq m$. An example of the structure of W is depicted in Figure 2.

$$W = \begin{array}{cc} & \begin{array}{cc} m & n \end{array} \\ \begin{array}{c} \left[\begin{array}{cc|cc} 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right] & \begin{array}{c} m \\ n \end{array} \end{array}$$

Fig. 2. Example mask array W , (with $n = 4$ and $m = 2$). Note that not all hop-counts from end hosts to monitors are observed, and none of the hop-counts *between* end hosts are observed.

The embedding process amounts to assigning a d -dimensional vector to each monitor and source. Each vector is generically written as X_i , for the i -th monitor or source. The collection of all vectors, or embedded points, is represented by the $(m+n) \times d$ matrix X . The embedding is designed by minimizing the *stress function*:

$$\text{stress}(X) = \sum_{i=1}^{n+m} \sum_{j=1}^{n+m} W_{i,j} (\|X_i - X_j\| - D_{ij})^2 \quad (2)$$

with respect to the embedded points X . Intuitively, minimizing the stress function will minimize the squared error between the estimated distance between end hosts and the observed hop count distances. Note that the stress function places no cost on unobserved distances.

B. Stress Optimization Method

The stress function is minimized through an iterative procedure. Let $X^{(t)}$ denote the embedded points at iteration t . Using the majorization method from the MDS literature [15], we can bound the stress function from Equation 2 using a convex function and then minimize the bounding function. The procedure guarantees the stress is reduced at each iteration; *i.e.*, $\text{stress}(X^{(t+1)}) \leq \text{stress}(X^{(t)})$. The bounding function can be solved iteratively by:

$$X_{i,a}^{(t+1)} = z_i \sum_{j \neq i} W_{i,j} \left(X_{j,a}^{(t)} + \frac{D_{ij} (X_{i,a}^{(t)} - X_{j,a}^{(t)})}{\|X_i^{(t)} - X_j^{(t)}\|} \right) \quad (3)$$

For $z_i = \frac{1}{\sum_{j \neq i} W_{i,j}}$, $i = 1, 2, \dots, n+m$ and $a = 1, 2, \dots, d$ (where d is found using Equation 1 on the m -by- m monitor

distance array A). A proof of the monotonically decreasing stress of this method is stated in [18]. The procedure is repeated until the embedding coordinates converge to where $\|X^{(t)} - X^{(t+1)}\| < \epsilon$, with $\epsilon > 0$. The step-by-step methodology for solving this problem is described in Algorithm 1.

Algorithm 1 - MDS Algorithm with Incomplete Passive Measurements

Initialize:

- Using the matrix of pairwise distances between passive monitors (A), use Equation 1 to find the embedding dimension d .
- Randomly create d -dimensional placement vector $X_i^{(1)}$ for all $i = 1, 2, \dots, n+m$.
- Set $t = 1$, $\epsilon > 0$.

Main Body

- 1) Adjust the monitor embedding

$$X_{i,a}^{(t+1)} = \frac{1}{m-1} \sum_{i=1, j \neq i}^m \left(X_{j,a}^{(t)} + \frac{A_{ij} (X_{i,a}^{(t)} - X_{j,a}^{(t)})}{\|X_i^{(t)} - X_j^{(t)}\|} \right)$$

For $i = \{1, 2, \dots, m\}$

- 2) Adjust all the end host node embedding with respect to the monitor embedding. We can reduce the formula in Equation 3 to:

$$X_{i,a}^{(t+1)} = \frac{1}{|\mathcal{I}^{(i)}|} \sum_{j \in \mathcal{I}^{(i)}} \left(X_{j,a}^{(t)} + \frac{h_{ij} (X_{i,a}^{(t)} - X_{j,a}^{(t)})}{\|X_i^{(t)} - X_j^{(t)}\|} \right)$$

For $i = \{m+1, m+2, \dots, n+m\}$, where $\mathcal{I}^{(i)}$ is the set of monitor indices observed for end host S_i .

- 3) **if** $\|X^{(t)} - X^{(t+1)}\| < \epsilon$ **then**
 Go to **Step 4**.
 else
 Set $t = t + 1$ and go to **Step 1**.
 end if
 - 4) Estimate the pairwise hop distances using the embedding: $\hat{d}_{i,j} = \|X_i - X_j\|$
-

VI. EXPLOITING BGP DATA

Consider the setup in Figure 3, where two end hosts have the same observed hop count to a single monitor. Given only this single element of information, the embedding algorithm may place these two end hosts close (resulting in a small estimated pairwise hop distance), even though they exist in separate autonomous systems (ASes) and may be potentially far apart in the real Internet topology. By exploiting BGP routing data, we can extract the specific AS that each end host exists in, and use that information to improve our pairwise hop estimation technique.

TABLE I
COMPARISON OF FOUR DIFFERENT TECHNIQUES FOR DISCOVERING PAIRWISE DISTANCES BETWEEN END HOSTS (n IS THE NUMBER OF END HOSTS, m ARE THE NUMBER OF MONITORS ($n \gg m$), AND k IS THE NUMBER OF ITERATIONS PERFORMED ON THE OPTIMIZATION PROCEDURE FROM SECTION V-B).

Method	Number of Active Measurements	Computational Complexity
Skitter	$O(n^2)$	$O(1)$
Landmark MDS using Active Measurements	$O(m^2 + nm)$	$O(m^2n + m^3)$
Landmark MDS using Passive Measurements	$O(m^2)$	$O(kmdn + m^3)$

Intuitively, given no other distance knowledge, one would prefer to embed end hosts close together if they exist in the same autonomous system (AS), while embedding the end hosts farther apart if they are in different ASes. From either examining BGP looking glass servers or our own selection of passive measurement monitors, we can gather characteristics of paths from end hosts in different ASes, including the mean distance of these paths (μ_B). We can weakly assume that end hosts in the same AS should have path lengths closer to zero, while end hosts in different autonomous systems should have path lengths closer to this average value, μ_B . This motivates creating a BGP distance matrix containing our weak assumptions on what the end host distances should be from the BGP data,

$$D_{i,j}^B = \begin{cases} \mu_B & : \text{if } S_i \text{ and } S_j \text{ are in different ASes} \\ 0 & : \text{if } S_i \text{ and } S_j \text{ are in the same AS} \end{cases}$$

For $1 \leq i, j \leq n + m$.

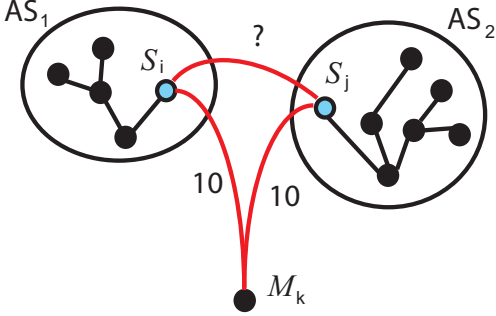


Fig. 3. Two end hosts with the same hop distance to a single monitor.

Instead of relying only on the incomplete hop count distances between the end hosts and the monitors, we can also use weak assumptions based on the AS distance information to embed the end hosts. Intuitively, one would not weight the weak assumptions of the AS distances the same as the observed hop count distances from passive measurements. It follows that the AS distance confidence for each pair should be inversely proportional to the variance for all paths. From a small set of probes, we can find the sample variance of the length of paths in the same AS, σ_{in}^2 , and the sample variance of the length of paths in different AS, σ_{out}^2 . This leads to the definition of a new mask array W^B ,

$$W_{i,j}^B = \begin{cases} \frac{1}{\sigma_{in}^2} \lambda & S_i \text{ and } S_j \text{ exist in the same AS} \\ \frac{1}{\sigma_{out}^2} \lambda & S_i \text{ and } S_j \text{ exist in different ASes} \end{cases}$$

For $1 \leq i, j \leq n + m$, and where the weight value, λ , can be adaptively found using a bisection method to find the value that embeds the monitors with the lowest error (with respect to the observed pairwise hop distances to other monitors).

The stress function for this AS exploiting technique can then be defined as:

$$\text{stress}^B(X) = \sum_{i=1}^{n+m} \sum_{j=1}^{n+m} W_{i,j} (\|X_i - X_j\| - D_{ij})^2 + \sum_{i=1}^{n+m} \sum_{j=1}^{n+m} W_{i,j}^B (\|X_i - X_j\| - D_{ij}^B)^2$$

The methodology for minimizing this stress function is described in Algorithm 2.

VII. RESULTS

A. Passive Measurement Datasets

We use two different data sets to evaluate the algorithms described in the prior sections. The first are a set of topologies generated by Orbis [19]. 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.

The second data set that we use in this paper is a router-level connectivity map of the Internet based on data collected by Skitter [4]. Measurements in Skitter are based on traceroute-like active probes sent from a set of 24 monitors to a set of nearly 1M target hosts distributed throughout the Internet. We use the openly available router-level map create from data collected between April 21 and May 8, 2003. This map consists of 192,224 unique nodes and 609,066 undirected links. It is important to note that the goal of the Skitter target host list is to have one responding node in each /24 prefix. Thus, the characteristics of the Skitter graph with respect to destination subnets is different from Orbis generated topologies, which reflect collections of nodes in subnets.

Our analysis assumes that the only data that will be used to infer network structure is the end host address (used only to uniquely identify a host) and TTL extracted from the header of

Algorithm 2 - MDS Algorithm with Incomplete Passive Measurements *and* BGP Information

Initialize:

- 1) Using the matrix of pairwise distances between passive monitors (A), use Equation 1 to find the embedding dimension d .
- 2) Using bisection search, find λ that minimizes the error embedding the monitors in d dimensions.
- 3) Randomly create d -dimensional placement vector $X_i^{(1)}$ for all $i = 1, 2, \dots, n + m$.
- 4) Set $t = 1, \epsilon > 0$

Main Body:

- 1) Adjust the monitor embedding

$$X_{i,a}^{(t+1)} = \frac{1}{m-1} \sum_{i=1, j \neq i}^m \left(X_{j,a}^{(t)} + \frac{A_{ij} (X_{i,a}^{(t)} - X_{j,a}^{(t)})}{\|X_i^{(t)} - X_j^{(t)}\|} \right)$$

For $i = \{1, 2, \dots, m\}$.

- 2) Adjust all the end host node embedding with respect to the monitor embedding and the AS information

$$X_{i,a}^{(t+1)} = \frac{1}{|\mathcal{I}^{(i)}| + \sum_{j=1}^{n+m} W_{i,j}^B} (s_i + t_i)$$

For $i = \{m+1, m+2, \dots, m+n\}$, where $\mathcal{I}^{(i)}$ is the set of monitor indices observed for end host \mathcal{S}_i . Where:

$$s_i = \sum_{j \in \mathcal{I}^{(i)}} \left(X_{j,a}^{(t)} + \frac{h_{ij} (X_{i,a}^{(t)} - X_{j,a}^{(t)})}{\|X_i^{(t)} - X_j^{(t)}\|} \right)$$

$$t_i = \sum_{j=1, j \neq i}^n W_{i,j}^B \left(X_{j,a}^{(t)} + \frac{D_{i,j}^B (X_{i,a}^{(t)} - X_{j,a}^{(t)})}{\|X_i^{(t)} - X_j^{(t)}\|} \right)$$

- 3) **if** $\|X^{(t)} - X^{(t-1)}\| < \epsilon$ **then**
 Go to **Step 4**.
else
 Set $t = t + 1$ and go to **Step 1**.
end if
 - 4) Estimate the pairwise hop distances using the embedding: $\hat{d}_{i,j} = \|X_i - X_j\|$
-

each packet. In the case of the Orbis and Skitter data sets, we synthesize these values. In real world applications, honeynet data can be analyzed using the clever technique described in [20] to infer the number of hops between the honeypot monitor and the host. This inference is made based on the fact that (i) there are only a few initial TTL values used in popular operating systems (e.g., 64 for most UNIX variants, 128 for most Microsoft variants and 255 for several others), and (ii) typical hop counts for end-to-end paths are far less than the differences between the standard TTL values. Thus, hop count is inferred by rounding the TTL up to the next

highest initial TTL value and then subtracting the initial TTL.

B. Mean Hop Distance Estimation

To compare the performance of our embedding algorithms, we must find a methodology to contrast against. Due to the reliance of our purposed method on incomplete passive measurements, the previous network embedding algorithms in [14], [13], [8] are not comparable. We instead choose the most logical, a mean estimation approach, where the estimated pairwise hop distance for any pair of end hosts in the topology are estimated to be the mean of the pairwise hop distance found by active measurements between the monitor nodes (represented by the matrix A).

$$\hat{d}_{\text{mean}} = \frac{1}{m^2} \sum_i^m \sum_j^m A_{i,j}$$

Note that the estimate of \hat{d}_{mean} is the same for all choice of end hosts i, j .

C. Embedding Experiments

For each dataset, we obtain a low-dimensional embedding using the MDS procedure described above. Given the low-dimensional embedding of each end host, we estimate (or impute) the hop-count distance between pairs of end hosts. We begin by running the iterative methods from Section V and Section VI on a series of synthetic Internet topologies generated using the Orbis tool [19]. For the purposes of this paper, we assume that the passive measurement monitors are randomly placed in the topology, with our experiment monitors randomly chosen from the set of leaf nodes in the topology, with the passive measurements simulated as the length of the shortest path found in the topology between the end hosts and the monitors. In all experiments, the BGP/AS information was synthetically created by randomly choosing a subset of 15 end hosts and classifying each end host's AS as the index of the pairwise closest end host in the random subset.

The error metric used to assess the estimation accuracy is the Root Mean Squared Error (RMSE) defined as:

$$\begin{aligned} RMSE(\hat{D}) &= \sqrt{\sum_{i,j} (D_{i,j} - \hat{D}_{i,j})^2} \\ &= \sqrt{\sum_{i=m+1}^{n+m} \sum_{j=m+1}^{n+m} (D_{i,j} - \|X_i - X_j\|)^2} \end{aligned}$$

If our estimator has an RMSE of 1, then we can estimate the true hop distance (on average) within a single hop. It also follows for the mean estimation approach (from Section VII-B), that the RMSE results are the sample standard deviation of the ground truth hop count values.

In Figure 4, the size of the topology is held constant, while performance of the algorithm is compared as the number of monitors varies between graphs (from 8 measurement monitors to 32 measurement monitors). The error rate (RMSE) is plotted against the amount of "missingness" in the passive

measurement data, where if there are 16 monitors, then X observed features refers to observing X out of 16 elements for each hop count vector with the observed elements chosen at random.

Figure 5 shows the performance of the algorithm in estimating pairwise distances between end hosts are seen with respect to varying topology sizes, where using the Orbis topology generating toolkit the topologies are rescaled in terms of the total number of nodes in the graph (from 1000 to 2000 nodes).

In both Figures 4 and 5, the performance of the algorithm is seen with and without the autonomous system information. The results show that when there is very little information from the passive measurements (low total number of monitors, high levels of incompleteness, etc.), the BGP exploiting algorithm performs considerably better than the embedding algorithm that ignores the BGP data. It is only when there is a large amount of available passive measurement data, as shown in Figure 4-(right), that the BGP exploiting algorithm is moderately outperformed by the algorithm that ignores the BGP data. Meanwhile, both MDS algorithms outperform the naive mean estimation approach given moderate levels of missingness in the hop count vectors.

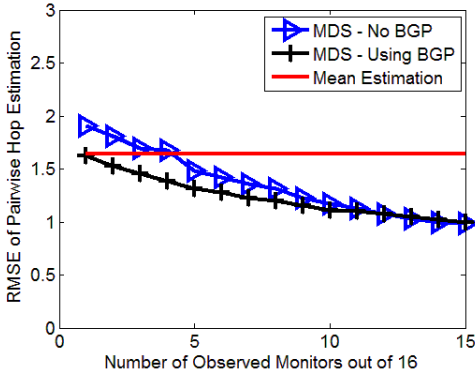


Fig. 5. Simulation results for error rates of pairwise hop estimation for synthetic topology versus amount of available data ($m=16$, $n=2000$)

D. Dimensionality Selection

Using the dimensionality selection technique described in Section IV-A, we now examine how much a reduced dimension embedding space effects our estimation technique. In Figure 6, we show the performance of the MDS algorithm (with no BGP/AS information) as the dimension of the embedding space increases. One can see that after the determined embedding dimension $d = 5$ (found using Equation 1), any additional embedding dimension has very little reduction on both the RMSE and the confidence bounds of the pairwise distance estimation.

E. Additional Monitors

Given the improvement on the pairwise hop estimation by adding monitors (as seen in Figure 5), one important question could be what effect does each additional monitor added

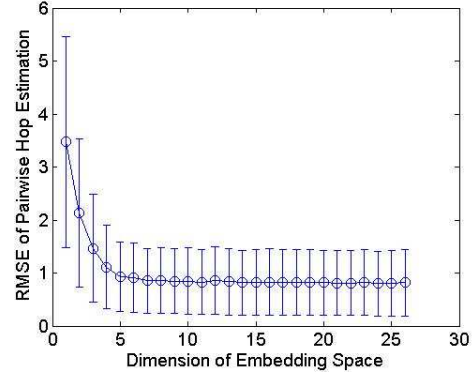


Fig. 6. The effect of embedding dimension to estimating the pairwise distance values for the synthetic topology, $n = 1000$, w/ $m = 32$ and calculated dimension $d = 5$, confidence bars indicating ± 1 standard deviation

to the experiment have on the estimation error rate? Using the synthetic topology, in Figure 7 we see the effects that each additional monitor has on the estimation RMSE and the confidence bounds. As shown in the figure, after the placement of roughly 10 monitors, each additional node has relatively low impact on the resulting error rates and confidence bounds.

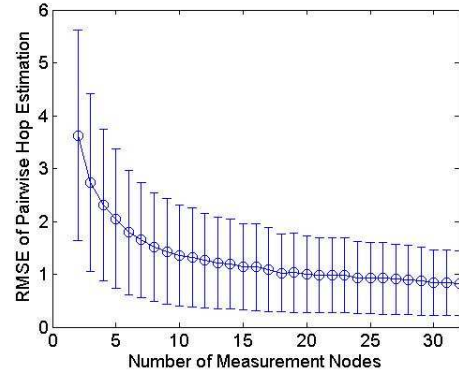


Fig. 7. The effect of adding additional monitors to estimating the pairwise distance values for the synthetic topology, observing complete hop count data, $n = 3000$, confidence bars indicating ± 1 standard deviation

F. Skitter

We also consider the performance of the estimation techniques using the Skitter dataset [4]. Using the same methodology as the Orbis topology experiments, a random subset of $n = 1000$ leaf nodes were selected as end hosts and randomly selected other leaf nodes were selected as passive measurement monitors. The pairwise distances were estimated for different levels of missing data in Figure 8 using both embedding methodologies. Again, the MDS algorithm exploiting the BGP/AS information outperforms the non-BGP/AS algorithm for experiments with low levels of observed data.

G. Routing Asymmetry Results

The resulting topology estimates from the Multidimensional Scaling technique reveal a *symmetric* topology, where the

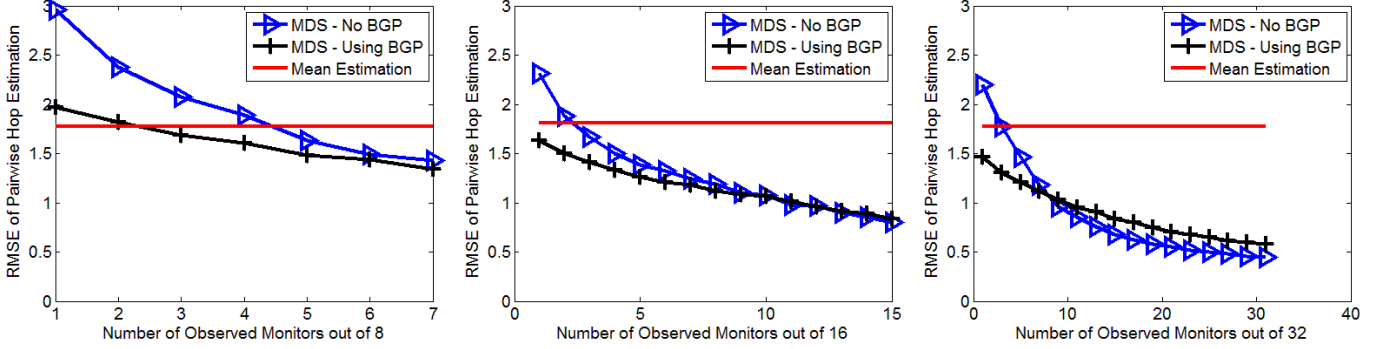


Fig. 4. Simulation results for error rates of pairwise hop estimation for synthetic topology versus amount of available data ($n=1000$). (left) $m=8$, (center) $m=16$, (right) $m=32$

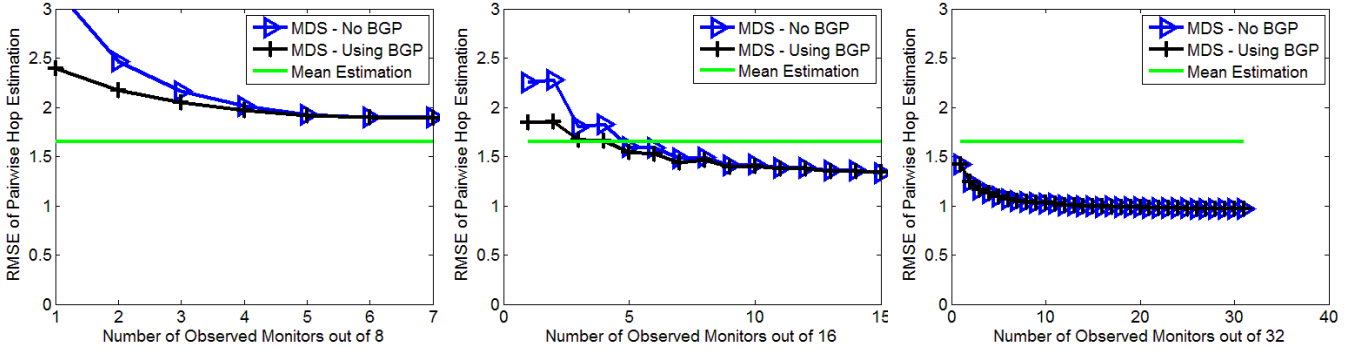


Fig. 8. RMSE of pairwise hop estimation simulation results for the Skitter topology ($n = 1000$). (Left) $M = 8$, (Center) $M = 16$, (Right) $M = 32$

forward and reverse paths between end hosts are of equal lengths. However, prior empirical research on Internet routing has shown that asymmetry on forward and reverse paths between hosts is not uncommon (*e.g.*, [21], [22], although we were unable to find prior empirical work that broadly quantifies the difference between the forward and reverse path lengths). Given the possibility of path length asymmetry, we examine the impact on our estimators when the reverse paths are off by one hop, two hops, and three hops (with equal probability of positive or negative length offset).

The effects of this distribution of path length asymmetry on our estimation methodology can be seen in Figure 9. Although the error rate increases as the amount of asymmetry in the forward and reverse hop counts increases, the MDS methodology (for both the BGP/AS and non-BGP/AS method) still outperforms the naive mean estimation methodology for all but the highest levels of missingness.

VIII. FUTURE WORK

The next step for our work is to attempt to construct a larger topology using these pairwise estimated distances. For the case of two end hosts and a single monitor, as seen in Figure 10, resolving the logical topology using both the passive measurements and the estimated pairwise hop distance is a trivial task. But, as the number of end hosts grows and the number of monitors increases, fusing the numerous

subtopologies and adding the necessary nodes to the topology becomes a non-trivial problem. One potential future direction is fusing our work on resolving shared path lengths between paths in [6] with the work in this paper. To improve upon

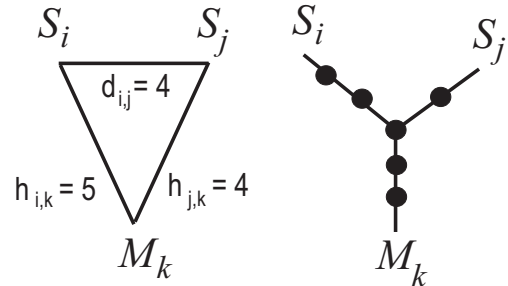


Fig. 10. Discovering the logical topology using distances

the performance of the algorithm, another potential future direction is to study how one can use directed active probes to fill-in our incomplete passive measurements. Also to be considered is the effect of intelligent placement of the passive measurement monitors as opposed to the random placement considered in this paper.

IX. CONCLUSIONS

Our goal is to establish comprehensive maps of Internet topology quickly, accurately and with minimal impact on the

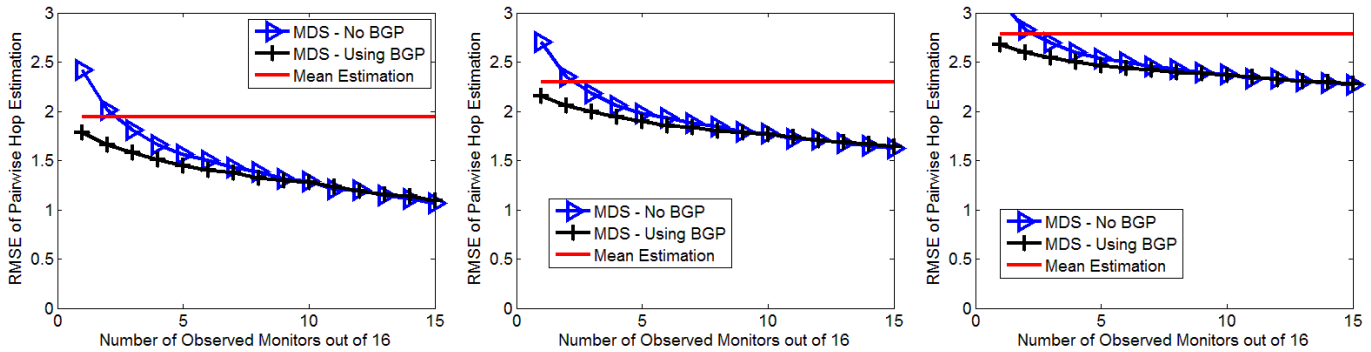


Fig. 9. Simulation results for asymmetric reverse paths for synthetic topology ($n = 1000, m = 16$) versus amount of available data. (left) Reverse paths off by 1 hop, (center) Reverse paths off by 2 hops, (right) Reverse paths off by 3 hops

network. In this paper, we take an important step toward that goal by presenting a method for establishing pairwise hop distances between nodes in the Internet. Our approach begins by deploying a landmark measurement infrastructure in the Internet, which uses active probes to establish accurate hop distances between all of its nodes. The measurement infrastructure is also enabled to passively monitor links and collect end host source addresses and TTLs from packets. We describe a multi-dimensional scaling algorithm that we apply to the active and passive measurements that generates estimates of hop distances between all observed nodes (end host addresses and landmarks). Our algorithm is designed to generate estimates even when all end host addresses have not been observed in all landmarks. The embedding algorithm is then enhanced to incorporate autonomous system information for the end hosts, which results in improved estimation performance. We evaluate the capabilities of our method using topologies generated by Orbis and from Skitter data. The results show that our method is able to generate highly accurate pairwise hop count estimates even with a small landmark infrastructure and a relatively large amount of missing data. Our ongoing efforts are focused on refinement of the methodology, additional empirical evaluations, and devising methods for using pairwise hop distances to establish and enhance Internet topology maps.

REFERENCES

- [1] R. Govindan and H. Tangmunarunkit, "Heuristics for Internet Map Discovery," in *Proceedings of IEEE INFOCOM '00*, Tel Aviv, Israel, March 2000.
- [2] N. Spring, R. Mahajan, and D. Wetherall, "Measuring ISP Topologies with Rocketfuel," in *Proceedings of ACM SIGCOMM '02*, Pittsburgh, PA, August 2002.
- [3] P. Barford, A. Bestavros, J. Byers, and M. Crovella, "On the marginal utility of network topology measurements," in *Proceedings of ACM Internet Measurement Workshop (IMW '01)*, San Francisco, CA, October 2001.
- [4] CAIDA, "The Skitter Project," <http://www.caida.org/tools/measurement/skitter/>, 2007.
- [5] B. Eriksson, P. Barford, R. Nowak, and M. Crovella, "Learning network topology from passive measurements," in *Proceedings of ACM Internet Measurements Conference (IMC) '07*, San Diego, CA, October 2007.
- [6] B. Eriksson, P. Barford, and R. Nowak, "Network Discovery from Passive Measurements," in *Proceedings of ACM SIGCOMM '08*, Seattle, Washington, August 2008.
- [7] H. Zheng, E. Lua, M. Pias, and T. Griffin, "Internet Routing Policies and Round Trip Times," in *Proceedings of The Passive and Active Measurement Workshop*, Boston, MA, April 2005.
- [8] E. Ng and H. Zhang, "Predicting Internet Network Distance with Coordinate-based Approaches," in *Proceedings of IEEE INFOCOM '02*, New York, NY, April 2002.
- [9] F. Dabek, R. Cox, F. Kaashoek, and R. Morris, "Vivaldi: A Decentralized Network Coordinate System," in *Proceedings of ACM SIGCOMM '04*, Portland, OR, August 2004.
- [10] L. Tang and M. Crovella, "Virtual Landmarks for the Internet," in *Proceedings of ACM Internet Measurement Conference (IMC '03)*, Miami, FL, October 2003.
- [11] P. Key, L. Massoulié, and D. Tomozei, "Non-Metric Coordinates for Predicting Network Proximity," *INFOCOM 2008. The 27th Conference on Computer Communications, IEEE*, April 2008.
- [12] J. Ledlie, P. Gardner, and M. Seltzer, "Network Coordinates in the Wild," in *Proceedings of USSENIX Network Systems Design and Implementation (NSDI '07)*, San Jose, CA, April 2007.
- [13] Y. Shavitt and T. Tanel, "Hyperbolic Embedding of Internet Graphs for Distance Estimation and Overlay Construction," *IEEE/ACM Transactions on Networking*, To Appear.
- [14] M. Costa, M. Castro, A. Rowstron, and P. Key, "PIC: Practical Internet coordinates for distance estimation," in *International Conference on Distributed Systems*, March 2004.
- [15] T. Cox and M. Cox, "Multidimensional Scaling." Chapman and Hall, 1994.
- [16] V. de Silva and J. Tenebaum, "Global versus local methods in nonlinear dimensionality reduction," in *Proceedings of NIPS (Neural Information Processing Systems) 2003*, 2003, pp. 721–728.
- [17] J. C. Platt, "Fastmap, metricmap, and landmark MDS are all Nystrom algorithms," in *Technical Report, Microsoft Research*, 2004.
- [18] E. Gansner, Y. Koren, and S. North, "Graph drawing by stress majorization," in *Proceedings Symposium on Graph Drawing GD'04*, September 2004.
- [19] 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.
- [20] C. Jin, H. Wang, and K. Shin, "Hop-Count Filtering: An Effective Defense Against Spoofed Traffic," in *Proceedings of IEEE INFOCOM '03*, San Francisco, CA, April 2003.
- [21] Y. He, M. Faloutsos, S. Krishnamurthy, and B. Huffaker, "On Routing Asymmetry in the Internet," *Global Telecommunications Conference, 2005. GLOBECOM '05. IEEE*, vol. 2, Nov.-2 Dec. 2005.
- [22] V. Paxson, "End-to-end Routing Behavior in the Internet," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 5, pp. 41–56, 2006.