Inferring Unseen Components of the Internet Core

Brian Eriksson, Paul Barford, Joel Sommers, Robert Nowak

***Abstract*—**

Despite many efforts over the past decade, the ability to generate topological maps of the Internet at the router-level accurately and in a timely fashion remains elusive. Mapping campaigns commonly involve `traceroute`-like probing that are usually non-adaptive and incomplete, thus revealing only a portion of the underlying topology. In this paper we demonstrate that standard probing methods yield datasets that implicitly contain information about much more than just the directly observed links and routers. Each probe yields information that places constraints on the underlying topology, and by integrating a large number of such constraints it is possible to accurately infer the existence of unseen components of the Internet (i.e., links and routers not directly revealed by the probing). Moreover, we show that this information can be used to adaptively re-focus the probing in order to more quickly discover the topology. These findings suggest radically new and more efficient approaches to Internet mapping. Our work focuses on the discovery of the core of the Internet. We define "Internet core" as the set of routers that is roughly bounded by ingress/egress routers from stub autonomous systems. We describe a novel data analysis methodology designed to accurately infer *(i)* the number of unseen core routers, *(ii)* the unseen hop-count distances between observed routers, and *(iii)* unseen links between observed routers. We use a large experimental dataset to validate the proposed methods. For our data set, we show that our methods can predict the number of unseen routers to within a 13% error level, estimate 60% of the unseen distances between observed routers to within a one-hop error, and robustly detect over 35% of the unseen links between observed routers. Furthermore, we use the information extracted by our inference methodology to drive an adaptive active-probing scheme. The adaptive probing method allows us to generate maps on our data set using 50% fewer probes than standard non-adaptive approaches.

***Index Terms*—Internet Topology, Matrix Completion, Inference, Internet Measurements**

I. Introduction

The performance and stability of Internet applications and services depends intrinsically on the individual networks over which their traffic flows. Among the significant challenges in ensuring acceptable performance and stability for applications and services are the lack of widely deployed end-to-end QoS mechanisms, dynamic events such as flash crowds, failures and outages that disrupt service, and malicious outbreaks and attacks that threaten infrastructure and end hosts. Given the scope and magnitude of these challenges, accurate and timely information about the characteristics and state of network infrastructure is of paramount importance.

Topological connectivity is one of the most basic characteristics of network infrastructure. Connectivity can be specified in different ways, but one of the most important is *router-level connectivity*: the graph structure of routers and the links between them. While router-level connectivity within an individual ISP may be well known and relatively simple for the provider to monitor, understanding connectivity beyond a single domain is problematic.

Mapping the Internet's router-level topology has been the subject of a large number of studies over the past decade (*e.g.,* [1], [2], [3], [4], [5], [6]). The standard method for gathering data on Internet topology is to use `traceroute`-like tools to send TTL limited probes between a set of sources and destinations. These tools return the IP addresses of router interfaces encountered in the source-to-destination (forward) direction of a path. The challenges in recovering the topology from these measurements include (1) interface disambiguation (*i.e.,* associating one or more IP addresses with a single router) [2], [5], (2) perspective (*i.e.,* understanding how much of the Internet is captured by a set of measurements) [3], and (3) measurement infrastructure management (*e.g.,* widely-deployed, active probe-based infrastructures, the load they impose on the Internet and the data they return) [7]. The end result is that despite concerted efforts, we are still unable to generate accurate maps of the Internet's router-level topology in a timely fashion.

In this paper, we address a subset of the general problem of router-level Internet topology mapping. Our objective is to infer the existence of components that have not been observed (unseen routers, unseen route lengths, and unseen links) in a partial probing of the Internet core. We appeal to prior work on Internet mapping to define what is meant by *the Internet core*. Specifically, the core is composed of the set of routers that are greater than one hop away from end hosts and is roughly bounded by the borders of stub autonomous systems [3]. We expand on this definition in Section III. We believe that this somewhat imprecise definition of *core* is sufficient since we would like to identify as large a core component as possible. We argue that unseen core inference is important because the large volume of traffic that traverses the core of the Internet (although this has been challenged recently in [8]).

We assume an infrastructure from which active probe-based measurements can be made and that the infrastructure has a relatively broad deployment. Using this infrastructure as the starting point for gathering data, our inference methodology has three components. The first component addresses the unseen core router problem. Namely, given an increase in

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

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

J. Sommers is with the Department of Computer Science, Colgate University.

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

This work was supported in part by the National Science Foundation (NSF) grants CNS-0831427, 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.

the network probing infrastructure, how many extra core routers will we find? Our solution to this problem is related to solutions for the so-called ‘unseen species problem’ but cast in a networking context. The second component of our methodology addresses the problem of inferring unseen links between observed routers. Instead of using measurements with router interface IP addresses, we exploit a matrix completion algorithm that is based on a novel exploitation of data returned by `traceroute` probes. This approach enables accurate and efficient estimation of connectivity without the need for interface disambiguation which has proven to be difficult and to have a significant impact on resulting maps [5]. The final component consists of a targeted, matrix completion-based probing methodology we call *TargetComplete* that merges our contributions of inferring unseen core routers and unseen core links to efficiently reveal areas of the core containing the most uncertainty. This efficiency is of critical importance in large-scale Internet topology studies where issues of timeliness will results in out-of-date information using exhaustive probing [2].

While inferring the existence of unseen components may at first blush seem impossible, bear in mind that the Internet topology is highly structured. Each active probe yields information that places constraints on the underlying topology. Our algorithms effectively integrate a large number of such constraints to produce inferences about unseen components.

The accuracy of our proposed inference methodology is validated as follows. We evaluate the capabilities of our method by using existing high quality topology data sets from the iPlane project [9], supplemented with a large set of additional `traceroute` measurements gathered from PlanetLab nodes [10]. The interface-disambiguated iPlane measurements act as a ground truth on the observed portion of the Internet for our topology estimation methods. From these data sets we identified nearly 115,000 core nodes, constituting the test set for our evaluation.

The methodologies described in this paper provide important insights on the topological characteristics of our empirical data. On this data set, experiments using our unseen core router methodology estimate the number of additional routers found given increased probing within a 13% error level. Using our unseen link methodology, we find 35% of the unseen links in the topology with limited false alarms. And finally, using our *TargetComplete* targeted probing methodology, we generate topology maps using less than 50% of the probes required using a non-adaptive technique. These improvements point to a new ability for timely large-scale Internet map generation through the use of our efficient, targeted methods.

The remainder of this paper is organized as follows. We discuss prior studies that are related to and inform our work in Section II. In Section III we describe our data gathering methods and infrastructure, and the datasets used in our evaluations. We provide an overview of our estimation methodologies in Section IV. In Section V, we describe our methodology for estimating the number of unseen core routers, and in Section VI we describe our methodology for estimating unseen core links. An application of this estimation of the unseen areas of the network is the TargetComplete probing methodology described in Section VII. We summarize, conclude and describe future directions for our work in Section VIII.

II. Related Work

There have been many studies that have focused on developing methods for mapping Internet topology. While topology can be considered on different levels including application-specific connectivity or autonomous system connectivity, our focus in this work is on router-level connectivity. A great deal of prior work in this area has focused on using active `traceroute`-like probes as a basis (*e.g.,* [1], [2], [3], [5]). In each case, these studies highlight several challenges associated with this kind of approach, including the need for widely distributed nodes from which probes can be sent (*i.e.,* to address the need for a broad perspective) and the difficult problem of interface disambiguation. A number of large topology mapping efforts that attempt to address the problem of limited perspective have been active for years including the well known Skitter [11] and Dimes [12] efforts. While the problem of interface disambiguation has been known since Paxson's work in the mid-1990's [13], the recent study by Sherwood *et al.* demonstrates how problematic this issue can be when using standard disambiguation techniques [5]. Another study that is related to ours is by Magoni and Hoerdt [14]. In that paper, the authors describe a `traceroute`-based approach and encounter the same difficulties with perspective and router interfaces. While we are informed by prior work on Internet topology mapping based on active probes, our study differs in our use of lightweight probes that measure hop counts in order to infer connectivity between nodes.

Our consideration of “Internet core” is informed by prior topology mapping studies including [1], [3], [15]. While these papers provide various definitions of “core”, we believe that a strict definition is of less importance and ultimately arbitrary. The goal of our study is not to find specific boundaries, but to find as much of the central component of the Internet as possible. To that end, our definition is similar to what is given in [16] — roughly that the core is bounded by routers that are greater than one IP hop beyond end hosts or border routers of stub autonomous systems.

Our work is also informed by Eriksson *et al.* in [6], [17], [18]. In those studies, the authors propose methods for establishing Internet maps based on passive observations of hop counts in packets. While the idea of using inference methods to estimate incomplete hop counts is similar, our work differs from theirs in objective (unseen core inference), data (the use of active probes), and methods (unseen router estimation, matrix completion, unseen link estimation). The work in [18] examines the problem of estimating pairwise hop counts using incomplete measurements to a set of landmarks. In contrast, our work will demonstrate a methodology for estimating pairwise hops using only massively incomplete pairwise distance observations between the objects.

Our recent work in [19] describes DomainImpute, a new method for inferring unseen properties of Internet topology. Specifically, the DomainImpute algorithm demonstrates that by intelligent segregation of interdomain/intradomain links, hop distances between routers can be more accurately estimated compared with off-the-shelf Matrix Completion techniques. The DomainImpute algorithm is a specific extension of the unseen link estimation work described in this paper, in contrast to the larger Internet core topology inference and measurement framework described here.

A standard problem in statistics is the "unseen species problem", where given an incomplete observation, we try to estimate how much was missed. Classic results in [20] estimated the number of unseen species of moths in an environment given a limited observation, and the work in [21] estimates the total number of words Shakespeare knew given his collective works. Recently, methodologies in both [22] and [23] have examined the problem of unseen species estimation in the context of networking. Both of these methodologies are directed towards finding the *total* number of routers/links in a network given limited observations. While estimating the total number of unseen routers is a compelling problem, validating the results is an impossible task without information pertaining to the entire network (infeasible when considering the Internet). Here we focus on the problem of estimating how many additional routers would be found given a fractional increase in the probing infrastructure. This would be of interest to anyone trying to determine whether or not to continue probing a network to discover additional nodes.

We show in Section VI that the link lengths between core routers can be constructed into a very large hop count matrix. Due to the limited number of probes sent throughout the network, this matrix is very sparsely populated. Recent work in [24] has shown that matrices of size $N \times N$ and of linear rank r can be *exactly* reconstructed with only k known elements, where $k \sim O(rN \log N)$. Due to the very large size of the matrix, our work uses an efficient matrix factorization method from [25]. We use these prior results in an attempt to infer the unobserved path lengths between arbitrary core routers. By expanding upon these techniques, we develop a novel methodology for estimating unseen link locations in the network, an issue previously unexplored in Internet literature.

While our TargetComplete probing methodology in Section VII looks at a similar goal as prior work on the Double-Tree algorithm [26] and Vantage Point Spreading algorithm [27], both prior algorithms use specially crafted probes to limit the number of probes needed to discover the topology. In the network tomography literature, recent work in [28], [29] has also examined how to decrease the number of probes required to reconstruct logical tree topologies using delay-based methods. In contrast to this prior work, our TargetComplete methodology focuses on the reduction of the number of source-destination pairs used to probe the network using standard off-the-shelf TTL limited probing techniques (e.g., `traceroute`), not on the crafting of special probes to minimize measurement load. In addition to generating efficient topology maps, we offer the results of our TargetComplete technique as validation that the unseen core discovery techniques of Section V and Section VI are correctly revealing areas of particular uncertainty in the network.

III. Experimental Data

In this section we describe the core router data used in our study and how it was collected. The goal of our data collection effort was to establish a data set from which our methods could be evaluated. Specifically, we required a representative corpus of core Internet routers with disambiguated interfaces (to the extent possible using available techniques) that could act as ground truth for our work.

A. Core Router Definition

In this work we adopt a pragmatic definition of which routers constitute the Internet's core. Consider the result of performing a `traceroute` probe from an end host in a stub network with prefix A to a host in a stub network with prefix B. In the ordered list of routers obtained from tracing the route, the first router considered part of the core is the one with the last occurrence of an address in prefix A. The last router considered part of the core is the first one responding with prefix B. The attempt is to avoid considering any router as part of the core that is fully within a network with the same prefix as an end host.

This approach is conservative in the sense that it is likely to omit some core routers from consideration, (*e.g.*, routers that connect to actual core nodes within a single AS). However, as stated above, our intent is not to focus on exactly identifying the core boundary, but rather to accurately capture the gross characteristics of the core, and to develop a core topology inference framework. In the future, the methods developed in this paper may find application in boundary identification from our set of candidate core nodes.

B. Core Router Identification

In order to identify as many core router IPv4 addresses as possible, we leveraged high-quality data provided by an on-going measurement project and collected additional data using the Planetlab [30] infrastructure. The existing data was provided by the iPlane project [9], which performs a `traceroute` probe from all available Planetlab hosting sites to a set of target prefixes obtained through the Routeviews project [31]. We used four weeks of iPlane data collected over the period of 12 December 2008 to 8 January 2009.

In addition to the iPlane data, we collected `traceroute` data between a full mesh of Planetlab hosting sites. At the time of our measurement collection there were over 900 *hosts* that are part of Planetlab, but there were only about 375 distinct *sites*. Of these sites, only a subset are available at any given time due to host maintenance or other issues. To perform each `traceroute`, we used the Paris `traceroute` tool [32]. Informed by the study by Luckie *et al.* [33], we invoked the tool once using UDP-based probes and a second time using ICMP-based probes for each destination in order to discover as many core routers as possible through active probing. We set options in the Paris `traceroute` tool so that an individual measurement between hosts took longer, but produced a low rate of probe traffic. We collected the full mesh of Planetlab `traceroute` probes three times (roughly evenly distributed) over the same period that we obtained the iPlane data set. Due to Planetlab site and host transience, we were able to use approximately 216 Planetlab sites for each of the three rounds of full mesh probing. We gathered three complete measurement data sets beginning on December 11, 2008, December 22, 2008 and January 6, 2009.

Using these two data sets, we were able to discover 125,146 unique core router IPv4 addresses. The total number of hops (links) observed between all of these nodes was 519,273. A standard problem in `traceroute`-based topological studies is the issue of IP interface disambiguation, which is also referred to in the literature as *alias resolution*. That is, Internet routers are typically assigned multiple IP addresses (*e.g.*, each interface on a router may have a unique IP address assigned to it). Identifying which addresses correspond to the same physical router is the role of alias resolution. To identify the core routers (*i.e.*, de-alias) our data set, we used a router alias database published by the iPlane project, which builds on previously published alias resolution methodologies, including those used by the Rocketfuel project [2]. After alias resolution, we identified 114,815 core routers. Indeed our main reason for using the iPlane data (as opposed to other widely available topology datasets) was that an IP alias database is also published.

We make no explicit assertions about the extent to which this data set represents the actual core of the Internet since there is no authoritative source for that information. While the limitations of using Planetlab for investigations of Internet topology are known, we argue that both the scope and content of the data are appropriate for our analysis since the goal of this work is primarily methodological in nature. Furthermore, it is important to note that since the objective of our study is accurate identification of the router-level connectivity (*i.e.*, the topology) of the Internet's core, the fact that our data are gathered over multiple days is of less importance than it would be if our goal was to identify *changes* in the topology. That is a subject for future work.

IV. Inferring Unseen Components of the Core

Our methodology for inferring the unseen components of the core of the Internet is divided into three components. Practically speaking, these components are predicated on having an initial set of source/destination hop count data from which core topology estimates can be made and a measurement infrastructure from which additional probes can be sent. The components of our discovery methodology are:

1) *Estimate the population size of unseen routers.* Use the initial data set to predict the number of additional core routers that would be discovered using additional probes.
2) *Estimate the unseen connectivity between observed core routers.* Use the massively incomplete set of observed hop counts between core routers to estimate unseen route lengths and infer unseen links between observed core routers.
3) *Specify where additional probes will improve estimates.* By fusing the methodologies of step #1 and #2, we develop the TargetComplete probing methodology to determine which host in the measurement infrastructure should send further probes (to a specified destination host) in order to discover unseen routers and unseen links.

We describe each component in the following sections. Toward our goal of being able to discover the core accurately and to maintain core maps over time, we envision these components being run on an on-going basis. It is important to note that in terms of practical deployment and use, the computational complexity of all of our algorithms is such that the unseen core estimates can be made with a probe load on the order of tens of thousands in total, which for a large distributed infrastructure is minimal.

V. Estimating the Number of Unseen Core Routers

Consider sending a series of `traceroute` probes through the network and observing a collection of core routers[1]. How complete is this set of core routers? Are there significantly more core routers in the network that have not been observed by the `traceroute` probing set? Determining this missing set size is analogous to the problem of predicting the number of unseen species in an environment given some sample set of observations, or estimating the size of Shakespeare's vocabulary based on the number of unique words appearing in his known works [20], [21]. Using the set of `traceroute` probes between Planetlab nodes and other points in the network, we can use properties of the occurrence of routers to predict how many more routers will be found in the network given increased probing. The prediction idea we employ is based on a more sophisticated version of the following simple idea. Suppose we randomly split the `traceroute` dataset into two halves. A certain number of routers are discovered by the routes in one half of the dataset, and a certain number of additional "new" routers are discovered in the other half. This gives us a rough idea of how many new routers might be discovered, were we to double the original size of the `traceroute` measurement campaign.

Now let us consider the problem of predicting the number of additional routers found in the core of the Internet as the result of increased measurements, based on the number we discover through the initial `traceroute` campaign. Consider `traceroute` probing the Internet from a set of sources to a set of destinations. Let n_i denote the number of core routers that appear in exactly i routes in this `traceroute` dataset. While the methodologies in [22], [23] have both examined the problem of unseen species estimation in the context of networking, their results are directed towards finding the *total* number of routers/links in a network given limited observations. For the purposes of this work we are interested in leveraging the methodology from [20] and [21], where the number of unseen routers will be estimated for a *fractional increase* in the number of destinations probed. We consider this a more practical problem than the previously framed unseen networking research, as it is important to have knowledge of what it is possible to discover using a feasible amount of additional probing of the network.

[1] We will refer to disambiguated interfaces as core routers

Missing Species Estimator - *Given the values $n_1, n_2, ...n_k$ where n_i is the number of routers that occur in exactly i routes in the* `traceroute` *dataset, the number of additional routers found by increasing the destination points by some fraction $t \in [0, 1]$ can be estimated as,*

$$\widehat{r}(t) := \sum_{i=1}^{k} (-t)^i \, n_i \ , \tag{1}$$

where the value t represents the percent increase in the number of probes, with $t = 1$ being a doubling of the probing infrastructure.

This estimator proposed in [21] relies on extrapolating the information from the observed data to a fractional increase of the number of observations. The rationale of the estimator hinges on two key assumptions. The first underlying premise is that the number of times a given router is observed increases roughly linearly as a function of the number of `traceroute` measurements, modulo a bit of randomness in this growth rate depending on the specific set of `traceroute` measurements employed. To test the validity of the first assumption, we observe the behavior of the growth of the number of times a router is encountered as a function of the number of probes. This observation is compared to a linear function fit to the observations. The agreement with a linear trend can be gauged by the R^2 *coefficient of determination* metric [34], which measures the linear relationship between observed average values and the best linear fit of these observed values. By definition, $R^2 = 1$ if there is perfect correlation between the observed values and the best linear fit, and $R^2 = 0$ if the two sets of sequences are uncorrelated. The average R^2 across all observed routers in our iPlane/Planetlab probe dataset was found to be $= 0.9986$ (with standard deviation $= 0.0013$), this indicates that the average router observations are almost perfectly correlated with the best linear fit.

The second assumption is that all `traceroute` measurements, past and future, are independent and identically distributed. This is reasonable in our situation because the sources and destinations in our measurement campaign are widely distributed end hosts in the Internet, and therefore the `traceroute` dataset is a fairly random sampling of paths through the Internet core.

A. Experimental Performance

From the Planetlab/iPlane probing infrastructure described in Section III, we observe 114,815 core routers. From the Missing Species Estimator, we can predict that from knowledge of core router occurrence characteristics, we will discover an additional 46,032 core routers given a doubling (*i.e.,* $t = 1$) of the `traceroute` probing infrastructure. Next, we would like to assess the accuracy of this estimate. We can test the accuracy by taking ten random realizations of probing only half of our dataset (taken by maintaining the same number of `traceroute` sources, but probing to only a randomly chosen half partition of the `traceroute` destinations). Across these ten experiments, these reduced probing sets resulted in finding on average only 91,018 core routers (in contrast to 114,815 core routers found in the full probing set). By the simple formula in Equation 1 and the characteristics of the routers found using these reduced probing sets, we would predict to find an average additional 38,582 core routers given the full probing set, for an average total of 129,600 core routers. This is only a 13% deviation from the actual total observed number of routers found using the complete set of destinations.

VI. Estimating Unseen Connectivity

Given that we can now estimate how many core routers were *not observed*, we now look to examine what can be said about the unseen topology associated with the core routers that *have been observed.* Specifically, given an observed set of core routers, we focus on accurately estimating the hop length between every pair of observed core routers. Our estimation method is based on the leveraging of `traceroute` measurements with disambiguated interfaces. The focus of the analysis below is similar to unseen core router estimation, namely to estimate the unseen connectivity between core nodes given `traceroute` measurements.

A natural way to represent the router-level topology of the Internet is by using a *hop count matrix* $\mathbf{H}^{(I)}$. For the entire IPv4 Internet, $\mathbf{H}^{(I)}$ is a $2^{32} \times 2^{32}$ matrix with each element $\mathbf{H}^{(I)}_{i,j}$ representing the number of routers between IP address i and IP address j. If the matrix $\mathbf{H}^{(I)}$ is known, then the router-level topology of the Internet in all places is completely resolved. In this paper, we will focus on reconstructing a portion of this full hop count matrix, denoted as the $N \times N$ matrix $\mathbf{H}$, given N observed core routers from the core measurement data. To fill in even this portion of the hop count matrix completely would require an infeasible N^2 probing of the Internet. Instead, we reconstruct this matrix by using the set of core measurements from Section III. We examine how the `traceroute` measurements (*i.e.,* containing labels of intermediate nodes) can be used to improve our connectivity estimates.

To construct the hop count matrix from `traceroute` measurements, consider a single `traceroute` probe sent between two nodes (p_1, p_2) returns the path,

$$p_1 \rightarrow r_1 \rightarrow r_2 \rightarrow r_3 \rightarrow r_4 \rightarrow p_2 \tag{2}$$

From this single path, many hop elements of $\mathbf{H}$ can be observed (assuming interface disambiguation), with r_1 being one hop away from r_2, two hops away from r_3, etc. Intuitively, this has a multiplicative effect on our ability to populate the hop count matrix $\mathbf{H}$ versus a single ping-style hop count measurement. In this fashion, we can use the large set of `traceroute` measurements to extract the hop count matrix.

It is obvious from Table I that many of the matrix elements have no information (indicated here as a "-"). Using only `traceroute` probes to fill in the core-router-to-core-router hop elements will result in a hop count matrix that could be highly incomplete depending on the perspective afforded by the Traceroute campaign. Given this assumed incomplete hop count observation matrix, our first objective is to impute (or "fill in") the missing observations to resolve the router-level topology of the observed core routers.

TABLE I
AN EXAMPLE HOP COUNT MATRIX USING OBSERVED HOP ELEMENTS FROM THE SINGLE `traceroute` PATH $p_1 \rightarrow r_1 \rightarrow r_2 \rightarrow r_3 \rightarrow r_4 \rightarrow p_2$ (WHERE "-" REPRESENTS AN UNKNOWN ELEMENT).

	p_1	p_2	r_1	r_2	r_3	r_4
p_1	0	5	1	2	3	4
p_2	-	0	-	-	-	-
r_1	-	4	0	1	2	3
r_2	-	3	-	0	1	2
r_3	-	2	-	-	0	1
r_4	-	1	-	-	-	0

A. *Matrix Completion Algorithm*

Given the observed elements of the hop count matrix, $\mathbf{H}$, we will now estimate the unseen hop elements. We appeal to recent work on *Matrix Completion* methodologies in [24], [25] that frames the estimation of incomplete matrices as the problem of inferring the matrix $\widehat{\mathbf{H}}$ with the lowest rank (the number of nonzero eigenvalues) that agrees with the observed elements.

$$\begin{aligned} &\text{minimize} \quad \text{rank}\left(\widehat{\mathbf{H}}\right) \\ &\text{subject to} \quad \widehat{H}_{i,j} = H_{i,j} \quad \text{s.t. } H_{i,j} \text{ was observed} \end{aligned}$$

Motivation for using Matrix Completion to impute hop counts is two-fold. First, given k observations from the $N \times N$ matrix $\mathbf{H}$ (for N number of end hosts), it was found in [24] that using the Matrix Completion algorithm, we can *exactly* reconstruct the matrix given the number of observations satisfy $O\left(rN \log N\right)$ (where r is the rank of the matrix). Therefore, even for massively incomplete matrices, given low rank structure we may be able to accurately reconstruct the unseen elements. Second, prior work in [18] shows that a hop count matrix can be accurately represented by a low-rank approximation, indicating Matrix Completion should perform well on these matrices.

B. *Experimental Performance of Matrix Completion*

A hop matrix is constructed using 10,276 core routers found by probing between 216 active Planetlab nodes and 375 Planetlab node destinations using the methodology described in Section III. This dataset is massively incomplete, with only 1.94% of the hop elements observed. To assess an accurate estimation error rate, 100,000 observed core router to core router hop elements (chosen completely at random) were held out of the dataset and used to validate the performance of the Matrix Completion procedure. The error metric used to assess the estimation accuracy is the Root Mean Squared Error (RMSE) defined as:

$$RMSE(\hat{H}) \quad = \quad \sqrt{\frac{1}{|\mathbf{y}|} \sum_{\{i,j\} \in \mathbf{y}} \left(H_{i,j} - \hat{H}_{i,j}\right)^2}$$

(Where $\mathbf{y}$ denotes the holdout set of coordinates and $|\mathbf{y}|$ is the size of the holdout set). If our estimator has an RMSE of 1, then we can estimate the hop distance (on average) within a single hop of the true hop distance. In addition to the Matrix Completion algorithm results, we look to compare the hop estimation results against a baseline hop estimation methodology.

1) Mean Hop Estimation: To provide a benchmark for comparison, we consider the following simple approach to impute the missing hop counts. One can estimate each missing hop count using the mean of the hop counts that have been observed.

$$\widehat{H}_{i,j} = \frac{1}{\left|\mathcal{P}\left(H\right)\right|} \sum_{x} \sum_{y} \mathcal{P}\left(H_{x,y}\right)$$

where $\mathcal{P}\left(H_{x,y}\right)$ is equal to $H_{x,y}$ if the hop length between routers x and y was observed and 0 if no information was observed, and $\left|\mathcal{P}\left(H\right)\right|$ denotes the total number of observed hop counts.

2) Missing Hop Count Estimation Results: For estimating the held-out hop elements in the 10,276×10,276 core router hop count matrix, we find that the mean methodology estimates hop counts with RMSE of 5.96, while the new Matrix Completion approach has a RMSE of 2.03. Therefore, the new Matrix Completion algorithm estimates the missing elements with accuracy almost 4 hops better than the mean hop estimation procedure on average. The empirical cumulative distribution of the errors can be seen in Figure 1, this shows the probability that the imputation deviation (absolute value of the difference between the true hop value and the estimated hop value) is less than or equal to the deviation value on the x-axis for both imputation methodologies. In Table II, the deviation of the estimated held-out hop element values from the true hop element values are shown with respect to both the Matrix Completion method and the mean imputation method. As evident in the table, while the hold out data for the Matrix Completion method is slightly biased by the 4.7% of the hold out data that is estimated with deviation of greater than 4 hops from the observed value, this is in comparison to the mean imputation method that has over 43% of the hold out data estimated with deviation over 4 hops. Meanwhile, over 60% of the unobserved hop elements can be estimated within 1 hop with the Matrix Completion method. This level of accuracy in estimation directly motivates classifying unseen links in the topology using this new Matrix Completion methodology.

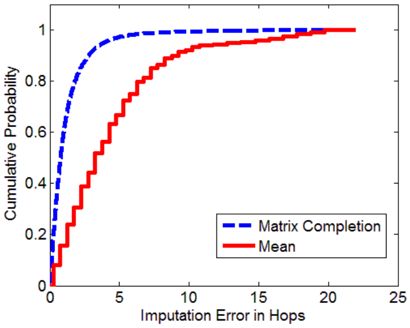


Fig. 1. Empirical Cumulative Probability for the imputation error using both Matrix Completion and Mean imputation.

C. *Inferring Unseen Core Links*

Using our hop-based Matrix Completion approach, we now have an estimated hop count matrix $\widehat{\mathbf{H}}$, containing the

TABLE II
DIVISION OF MATRIX COMPLETION ERRORS FOR HOLDOUT DATA

Error Range	Matrix Completion Percent	Mean Imputation Percent
Less than one hop off	60.2%	15.8%
Less than two hops off	82.7%	30.7%
Less than three hops off	91.3%	44.5%
Less than four hops off	95.3%	56.4%
More than four hops off	4.7%	43.6%

predicted hop counts between all the core routers found by probing. We now infer the existence of unseen links between routers given our estimated matrix. We wish to identify *links*, source-destination pairs $\{i, j\}$ where $H_{i,j} = 1$ but $H_{i,j}$ was not observed.

Given an estimated hop count matrix $\widehat{\mathbf{H}}$, one simple methodology for estimating unseen links would be to threshold the estimated hop count values. This methodology would classify all estimated hop values below a certain threshold (λ) as a link. This results in the creation of a hop count thresholding adjacency matrix $\mathbf{A}^{(hop)}$,

$$A^{(hop)}_{j,k} = \begin{cases} 1 & : \text{if } \widehat{h}_{j,k} < \lambda \\ 0 & : \text{otherwise} \end{cases} \tag{3}$$

Where the chosen value of λ gives an explicit trade-off between the number of links missed and the number of false links erroneously declared as existing (*i.e.,* the false alarm rate).

A deficiency of this hop thresholding methodology is that there is no consideration for the variance of our hop estimate. For areas of the network with high uncertainty, this method could possibly erroneously classify links. Instead, we consider the statistical inference methodology of *bootstrapping* [35] to systematically decide which pairs of observed routers have connected links in the topology by considering the variance of our hop count estimate.

The bootstrap thresholding methodology will be executed as follows. Consider repeatedly subsampling the observed hop count data, where each subsample $\mathbf{H}^{(i)}_{B}$ (for $i = \{1, 2, ..., M\}$), contains 95% of the observed hop counts chosen at random (with the other 5% of the observed hop counts held out from consideration). By performing the Matrix Completion algorithm on each subsampling matrix, we obtain M estimates of the full hop count matrix $\widehat{\mathbf{H}}^{(i)}_{B}$. Using this repeated subsampling when $M = 40$, for each unobserved hop count we will have the set of 40 estimates, $\{\widehat{h}^{(1)}_{j,k}, \widehat{h}^{(2)}_{j,k}, ..., \widehat{h}^{(40)}_{j,k}\}$. To estimate the stability of these estimates, we find the empirical *bootstrap confidence limits* $\left[\widehat{h}^{L}_{j,k}, \widehat{h}^{U}_{j,k}\right]$ by sorting the set of estimates and taking the second smallest hop count value ($\widehat{h}^{L}_{j,k}$) and the second largest estimated hop count value ($\widehat{h}^{U}_{j,k}$). These are the empirical 95% bootstrap confidence limits on our estimation of this hop count value, where given our observed data we are 95% confident that the true hop value, $h_{j,k}$ lies between $\left[\widehat{h}^{L}_{j,k}, \widehat{h}^{U}_{j,k}\right]$.

These confidence limits help to inform which elements are links. Consider the empirical bootstrap confidence limit $\left[\widehat{h}^{L}_{j,k}, \widehat{h}^{U}_{j,k}\right]$. If the confidence upper bound, $\widehat{h}^{U}_{j,k}$, was close to one, this would imply that we are confident that the true value of $h_{j,k}$ is one due to the confidence region containing no other possible hop count value. Therefore, we would very likely infer that there is a link in the topology between routers j and k. On the other hand, if the confidence upper bound was much larger than one, this would imply that we are not confident that the true value $h_{j,k}$ is one, and therefore no link likely exists between core routers j and k. This intuition gives rise to a thresholding methodology to find the bootstrap thresholding adjacency matrix $\mathbf{A}^{(boot)}$ where an unseen adjacency (core link) is implied to exist if the confidence upper bound, $\widehat{h}^{U}_{j,k}$ is below some value λ, and an adjacency is assumed not to exist if the confidence upper bound is greater than λ:

$$A^{(boot)}_{j,k} = \begin{cases} 1 & : \text{if } \widehat{h}^{U}_{j,k} < \lambda \\ 0 & : \text{otherwise} \end{cases} \tag{4}$$

The complete bootstrap thresholding methodology is described in Algorithm 1.

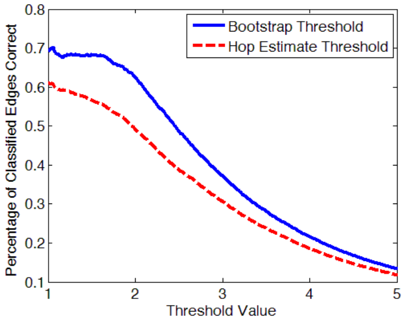


Fig. 2. Percentage of total links correctly classified plotted against threshold of confidence upper bound (λ) for both bootstrap upper bound estimate and hop count estimate.

D. *Experimental Performance of Unseen Link Inference*

Using the probing set of 10,726 core routers found between Planetlab nodes, we tested the performance of our unseen link classification methodology. After holding out 500,000 randomly-chosen hop count values containing 6,116 links (where we have observed $h_{i,j} = 1$) and 493,884 non-links (where we have observed $h_{i,j} > 1$). This hop count estimation was repeated 40 times and the Unseen Core Link Estimation methodology was tested. This behavior can be seen in Figure 2, which shows the percentage of correctly classified links (out of all the links classified) against the threshold value for both the bootstrap thresholding methodology and the hop count thresholding methodology. The results show that the bootstrap thresholding methodology is more accurate at classifying true unseen links than the more simple hop count thresholding methodology, with almost 10% more edges classified correctly for values of the threshold $\lambda < 2$. For the bootstrap thresholding methodology, the results in Figure 2 indicate that performance of link classification is good for $\lambda \leq 2$, with a majority (roughly 70%) of identified links being true links with performance degrading as the threshold value $\lambda > 2$. The threshold $\lambda \leq 2$ can be interrupted as follows: given the observed data we are confident that there are no intermediate hops between pairs of core routers classified as directly linked. For $\lambda > 2$, meaning that our 95% confidence bounds contains possible hop values greater than one, the performance of the classifier degrades significantly as λ is increased. Note the sharp knee in the curve at $\lambda = 2$, indicating that immediately past this point the fraction of correctly identified links drops off precipitously. Therefore, the logical choice for the threshold using the bootstrap methodology is $\lambda = 2$.

Algorithm 1 - Unseen Link Estimation Algorithm - Bootstrap Thresholding

```
1) Obtain 40 subsampled versions of the observed hop count matrix $\mathbf{H}_B^{(i)}$ ($i = \{1, 2, ..., 40\}$). Where each matrix contains a randomly chosen 95% of the total observed elements.
2) Perform Matrix Completion of each of the subsampled hop count matrices. Obtaining the estimated matrices, $\widehat{\mathbf{H}}_B^{(i)}$ ($i = \{1, 2, ..., 40\}$)
3) For every unobserved hop count element $h_{j,k}$
    a) Using the vector of bootstrap estimates, $\{\widehat{h}_{j,k}^{(1)}, \widehat{h}_{j,k}^{(2)}, ..., \widehat{h}_{j,k}^{(40)}\}$, sort the estimated values from smallest to largest, $\{\widehat{h}_{j,k}^{\mathcal{I}(1)}, \widehat{h}_{j,k}^{\mathcal{I}(2)}, ..., \widehat{h}_{j,k}^{\mathcal{I}(40)}\}$. Such that $\widehat{h}_{j,k}^{\mathcal{I}(i)} \leq \widehat{h}_{j,k}^{\mathcal{I}(i+1)}$
    b) Eliminate the smallest and largest values ($\widehat{h}_{j,k}^{\mathcal{I}(1)}, \widehat{h}_{j,k}^{\mathcal{I}(40)}$) to obtain the 95% empirical bootstrap confidence bounds.
    c) The remaining largest value is now the bootstrap confidence upper bound, $\widehat{h}_{j,k}^{U} = \widehat{h}_{j,k}^{\mathcal{I}(39)}$.
    d) If $\widehat{h}_{j,k}^{U} < \lambda$, then establish an link between core routers $j, k$. Otherwise, no link exists.
```

The results for this Unseen Link Classification Algorithm on our 10,726 core router dataset can be seen in Table III for multiple values of the threshold λ on the two thresholding algorithms. As seen in the table, using the desired threshold (*i.e.,* $\lambda = 2$), the bootstrap methodology discovers 2,196 of the true links in the topology, with only 37.4% of the classified links being false alarm links[2]. This is in comparison to the hop count thresholding method (with $\lambda = 2$) which finds a greater number of true links (3,388 links classified) but more than half of the classified links using this methodology (50.7%) are false links. This significantly higher accuracy motivates the application of the bootstrap thresholding methodology over the simple hop estimate thresholding methodology.

VII. TargetComplete - Adaptively Targeted Core Probing

Given the unseen core router and unseen core link estimation procedures describe above, we now focus on a combination of these two algorithms in order to illuminate unseen areas of the Internet core. Our approach is to use our previous techniques for identifying unseen areas of the network to target for further probing. Specifically, consider the problem of trying to discover the core of the Internet using a series of traceroute probes between a set of possible traceroute end host sources, $\mathcal{S} = \{s_1, s_2, ..., s_N\}$, and a set of possible traceroute end host destinations, $\mathcal{D} = \{d_1, d_2, ..., d_M\}$. The path found by each traceroute probe between source s_i and destination d_j can be denoted by the subgraph $\mathcal{G}_{i,j}$. Therefore, the complete router topology that is visible from this full set of sources and destinations can be represented as,

$$\mathcal{G} = \bigcup_{i \in \mathcal{S}} \bigcup_{j \in \mathcal{D}} \mathcal{G}_{i,j}, \qquad (5)$$

Where $\mathcal{G}$ is the union of all the traceroute paths between end hosts in the source set to the end hosts in the destination set.

The key problem with this idea is that using all possible sources and all possible destinations requires performing all $N \cdot M$ possible traceroute probes, which in cases where N or M are very large may be infeasible. Also, there are likely redundant probes sent out where particular $\mathcal{G}_{i,j}$ subgraphs contribute little to no new information about the full graph $\mathcal{G}$ due to a majority of their vertices and edges already having been observed by a previous path. This realization motivates the following modification of the problem: given a set of possible traceroute end host sources and a set of possible traceroute end host destinations, our goal is to use the measurement data to determine the *source-destination probes* that best further exposes the core topology. In our methodology, exposing the core topology is defined as finding a significant percentage of unique core routers and a significant percentage of unique core router edges with respect to the core topology found using the entire source set $\mathcal{S}$ and the entire destination set $\mathcal{D}$.

A. Random Selection Probing

If we make no assumptions or inferences about the utility of probing each source-destination pair, then the only methodology available is to probe by arbitrarily selecting source-destination pairs. We will consider this method as the baseline performance to compare the performance against our targeted methodology

B. TargetComplete Probing Algorithm

Given the methodologies on unseen router estimation in Section V and the techniques on unseen edge estimation in Section VI, we can combine estimates to inform the user as to which source-destination pair will reveal the most information about the core.

1) Unseen Species Selection: Consider the case where we have already probed some subset of destinations from each traceroute source in the network. Therefore, for each source $\mathcal{S}_i$, we have a source subgraph $\mathcal{G}^*_{\mathcal{S}_i}$, consisting of the set of routers and links found probing from source $\mathcal{S}_i$ to the selected destinations. We can state the current combined probing subgraph,

$$\mathcal{G}^* = \bigcup_{i=1,2,...,k} \mathcal{G}^*_{\mathcal{S}_i}$$

[2] With the false negative links being the number of undetected links on this network.

TABLE III
PERFORMANCE OF UNSEEN LINK CLASSIFICATION ALGORITHM WITH VARIOUS THRESHOLD VALUES USING λ THRESHOLDING ON BOTH THE BOOTSTRAP THRESHOLDING AND THE HOP COUNT THRESHOLDING METHODOLOGIES.

	Hop Count Thresholding			Bootstrap Thresholding		
	$\lambda = 1$	$\lambda = 2$	$\lambda = 3$	$\lambda = 1$	$\lambda = 2$	$\lambda = 3$
Number of Classified True Links	1422	3388	4769	526	2196	3959
Number of Classified False Links	911	3484	10744	235	1315	6660
Percentage of True Links out of all Classified Links	60.9%	49.3%	30.7%	69.1%	62.6%	37.3%
Percentage of False Links out of all Classified Links	39.1%	50.7%	69.3%	30.9%	37.4%	62.7%

Which source-destination probe pair will return the largest number of unique core routers with respect to the combined probing subgraph $\mathcal{G}^*$? Given the limited information about the core topology in the combined subgraph $\mathcal{G}^*$, we do not have enough data to make an informed decision to solve this problem. A problem that can be solved is the slightly modified problem of *which source will return the largest number of unique core routers with respect to the specific source subgraph $\mathcal{G}^*_{\mathcal{S}_i}$?* Reframing the problem in this manner indicates that it is very similar to the unseen species problem of Section V. Given a source subgraph, $\mathcal{G}^*_{\mathcal{S}_i}$, we estimate the number of unique unseen routers that would be observed given a fractional increase in the number of destinations (analogous to the probing of another destination from this source) in this source subgraph. We argue that the source subgraph that is estimated to have the largest increase in the number of observed unique core routers with respect to its source subgraph should be considered the best choice for increasing the number of unique core routers with respect to the combined probing subgraph $\mathcal{G}^*$.

2) Matrix Completion Selection Algorithm: The unseen species method selects the best source to probe from, but what destination should be chosen? Another possible probing strategy arises from the unseen core connectivity estimation techniques in Section VI. Consider the $N \times M$ probe hop count matrix $\mathbf{H}^P$ representing the hop distances between the `traceroute` end host sources $\mathcal{S}$ and the `traceroute` end host destinations $\mathcal{D}$. From the `traceroute` probes, we receive the number of hops between the sources and destinations, thus filling in the probe hop count matrix (with the unprobed pairs having unknown hop count values). Instead of estimating the size of the changes on the subgraph (as the previous Unseen Species method), we send probes based on determining which unobserved source-destination pair has the most uncertainty in the probe hop count matrix with respect to the observed hop counts. To determine the uncertainty of the hop counts in each of the missing source-destination pairs, we consider performing K-fold cross-validation (CV) [35] on the observed hop count values, similar to our unseen link estimation methodology.

Using K-fold CV, the Matrix Completion algorithm is performed on the incomplete probe hop count matrix $\mathbf{H}^P$, yielding K estimates for each unknown source-destination pair hop count element ($\{h^{(1)}_{i,j}, h^{(2)}_{i,j}, ..., h^{(K)}_{i,j}\}$). These collections of estimates are used to obtain the variance of each unobserved hop count estimation value. The intuition is as follows, if the variance of the hop count estimation is low, it implies that given the current observed hop counts, we have a good idea of the topological distance between the selected source-destination pair. Conversely, if the variance of the hop count estimation is high, it implies that we do not know very much about the topological distance between the specific source and destination, making it a candidate for probing.

3) TargetComplete Probing Algorithm: We can combine the two methods, Matrix Completion Selection and Unseen Species Selection, to form the *TargetComplete* method that offers a "best of both worlds" solution that chooses both the best source and best destination for an additional `traceroute` probe. To perform TargetComplete, we simply use the Unseen Species selection method to find the best source to probe from (given the prior set of source subgraphs $\mathcal{G}_{\mathcal{S}_i}$), and then use the Matrix Completion selection methodology to find the destination for that chosen source that has the highest uncertainty (*i.e.,* average variance) given K-fold cross-validation and the Matrix Completion algorithm. An outline of the approach is shown in Algorithm 2.

Algorithm 2 - TargetComplete Probing Algorithm

Initialize:

- From every source, randomly probe some number of destinations.
- Fill in the observed source-destination pair elements in the probe hop matrix, $\mathbf{H}$.

Main Body

1) Using Equation 1 find $\widehat{i}$, the source that the unseen species estimator predicts will find the most unseen routers given an increase in probing.
2) Using K-fold Cross Validation and Matrix Completion, find the K cross validation estimates for each destination j for source $\widehat{i}$, $\{h^{(1)}_{\widehat{i},j}, h^{(2)}_{\widehat{i},j}, ..., h^{(K)}_{\widehat{i},j}\}$
3) Find the destination for source $\widehat{i}$ with the highest cross validation variance.
$$\widehat{j} = \underset{j}{\text{argmax}} \left(\text{var} \left(\{h^{(1)}_{\widehat{i},j}, h^{(2)}_{\widehat{i},j}, ..., h^{(K)}_{\widehat{i},j}\} \right) \right) \quad (6)$$
4) Probe the chosen source-destination pair $\left(\widehat{i}, \widehat{j}\right)$. Adding the observed hop count value, $h_{\widehat{i},\widehat{j}}$ to hop count matrix $\mathbf{H}$.
5) If there are still more source-destination pairs to probe, go to 1.

C. Targeted Probing Experiments

Using the 216 active Planetlab nodes as sources, we sent `traceroute` probes to a destination set of 360 Planetlab nodes. Those measurements identified a set of 10,276 core routers with 34,859 links found between them. To initialize the probing algorithm, we performed five `traceroute` probes to randomly selected destinations from each Planetlab node in the source set. Using the two probing methodologies pertaining to selecting source-destination pairs to probe (TargetComplete, random selection), we examine the performance of the probing methodologies on discovering the unseen core topology using further probing. In this analysis, we compare performance by considering the number of unique core routers and unique core edges found by each probing methodology.

Figure 3 shows the results with respect to the number of `traceroute` probes needed to find both previous unseen core routers and unseen core links given the two probing methodologies, showing considerably more core infrastructure being discovered by our TargetComplete procedure compared with the random probing approach. Table IV and Table V show the number of probes needed to discover a specified number of unique routers and links, respectively. The tables show that for both routers and links, the TargetComplete probing methodology uses less than half (50%) of the number of source-destination pair probes compared with a random methodology to obtain the same number of observed core routers/links. This suggests that the TargetComplete methodology correctly selects areas of the network about which the structure is uncertain, showing the power of both the unseen core router and matrix completion methodologies on revealing unseen areas of the network.

TABLE IV
UNSEEN ROUTER OBSERVATIONS - REQUIRED NUMBER OF `traceroute` PROBES NEEDED TO OBSERVE A SPECIFIED NUMBER OF UNIQUE ROUTERS.

	Random Probing	TargetComplete Probing	
# Routers Found	# Probes Required	# Probes Required	Percentage of Random Probes
250	545	185	33.95%
500	1,632	537	32.90%
750	2,511	913	36.36%
1,000	3,356	1,636	48.75%

TABLE V
UNSEEN LINK OBSERVATIONS - REQUIRED NUMBER OF `traceroute` PROBES NEEDED TO OBSERVE A SPECIFIED NUMBER OF UNIQUE LINKS.

	Random Probing	TargetComplete Probing	
# Links Found	# Probes Required	# Probes Required	Percentage of Random Probes
500	391	183	46.80%
1,000	1,465	413	28.19%
1,500	2,027	745	36.75%
2,000	2,712	1,197	44.14%

VIII. CONCLUSIONS AND FUTURE WORK

Generating timely and accurate maps of the Internet has been a compelling objective for some time, but remains beyond our grasp. The scope, diversity and dynamics of the infrastructure along with the fact that service providers often actively thwart measurement by standard methods all complicate the issue. In this paper we address a subset of the Internet mapping problem by restricting our focus to discovering unseen portions of the core. We argue that unseen core inference is important since the core carries a large amount of application and service traffic and that it is tractable since it excludes the hundreds of millions of end hosts and their associated links. We address the unseen core problem by developing a novel unseen discovery inference methodology. Our methodology consists of estimating unseen core nodes, inferring unseen links in the network, and TargetComplete, a targeted probing methodology that efficiently reveals unseen areas of the Internet.

We demonstrate the capabilities of our methodology using `traceroute` datasets collected in PlanetLab and by the iPlane project [9]. We show that our unseen core node technique estimates the number of additional core nodes found given increased probing with only a 13% deviation from the actual observed number. We also show that a matrix completion algorithm is able to estimate over 60% of the core links within one hop of actual and roughly 82% of the core links within two hops of their actual value. We further develop an unseen core link classification algorithm, which finds over 35% of the true unseen core links with limited false alarm links. Finally, we validated the performance of both unseen router and unseen link estimation methodologies by merging the techniques in the TargetComplete algorithm. In comparison with a baseline random probing procedure, TargetComplete discovers more topology components (*i.e.,* routers and links) while requiring less than half the number of `traceroute` probes.

Our novel, efficient TargetComplete technique points to new capabilities for timely large-scale Internet map generation. In addition to expanded network discovery through efficient, targeted probes, the individual components of our methodology estimate how much additional unseen topology still exists in the network, allowing for an intelligent stopping criteria for probing campaigns. In future work, we plan to investigate the problem of understanding how the core maps evolve by conducting measurements over longer time frames. We will also look to perform Internet core boundary identification on other Internet data sets using our established candidate core nodes from this study.

REFERENCES

[1] R. Govindan and H. Tangmunarunkit, “Heuristics for Internet Map Discovery,” in *Proceedings of IEEE INFOCOM*, Tel Aviv, Israel, March 2000.

[2] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson, “Measuring ISP topologies with Rocketfuel,” *IEEE/ACM Transactions on Networking*, vol. 12, no. 1, 2004.

[3] P. Barford, A. Bestavros, J. Byers, and M. Crovella, “On the Marginal Utility of Network Topology Measurements,” in *Proceedings of ACM IMW*, San Francisco, CA, October 2001.

[4] D. Alderson, L. Li, W. Willinger, and J. Doyle, “Understanding Internet Topology: Principles, Models and Validation,” *IEEE/ACM Transactions on Networking*, vol. 13, no. 6, December 2005.

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

[6] B. Eriksson, P. Barford, and R. Nowak, “Network Discovery from Passive Measurements,” in *Proceedings of ACM SIGCOMM*, Seattle, WA, August 2008.

[7] V. Paxson, “Strategies for Sound Measurement,” in *Proceedings of ACM IMC*, Taormina, Italy, October 2004.

[8] C. Labovitz, D. McPherson, and S. Iekel-Johnson, “2009 Internet Observatory Report,” in *NANOG 47*, October 2009.

[9] H. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani, “iPlane: An Information Plane for Distributed Services,” in *Proceedings of USENIX OSDI*, Seattle, WA, November 2006.

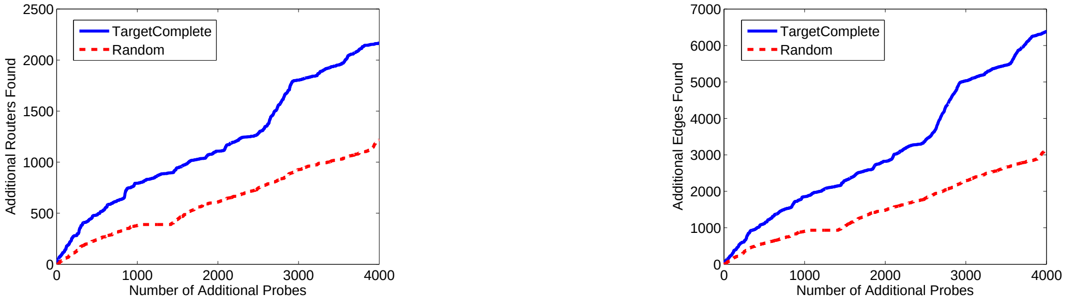


Fig. 3. (Left) - Number of additional unique core routers found using the two probing techniques, (Right) - Number of additional unique core links found using the two probing techniques

[10] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman, "PlanetLab: An Overlay Testbed for Broad-Coverage Services," *SIGCOMM CCR*, vol. 33, no. 3, pp. 3–12, 2003.

[11] CAIDA, "The Skitter Project," http://www.caida.org/, 2009.

[12] Y. Shavitt and E. Shir, "DIMES: Let the Internet Measure Itself," *ACM SIGCOMM CCR*, vol. 35, October 2005.

[13] V. Paxson, "Measurement and Analysis of End-to-end Internet Dynamics," Ph.D. dissertation, University of California - Berkeley, 1997.

[14] D. Magoni and M. Hoerdt, "Internet Core Topology Mapping and Analysis," *Elsevier Computer Communications*, vol. 28, October 2004.

[15] A. Broido and K.Claffy, "Internet Topology: Connectivity of IP Graphs," in *Proceedings of SPIE International Symposium on Convergence of IT and Communication*, Denver, CO, August 2001.

[16] J. Pansiot and D. Grad, "On Routes and Multicast Trees in the Internet."

[17] B. Eriksson, P. Barford, R. Nowak, and M. Crovella, "Learning Network Structure from Passive Measurements," in *Proceedings of ACM IMC*, San Diego, CA, October 2007.

[18] B. Eriksson, P. Barford, and R. Nowak, "Estimating Hop Distance between Arbitrary Host Pairs," in *Proceedings of IEEE Infocom Conference*, Rio de Janeiro, Brazil, April 2009.

[19] B. Eriksson, P. Barford, J. Sommers, and R. Nowak, "Ghost in the machine: Inferring unseen components in the internet core," in *Proceedings of IEEE INFOCOM Mini-Conference*, Shanghai, CN, April 2011.

[20] I. Good and G. Toulmin, "The Number of New Species, and the Increase in Population Coverage, When a Sample is Increased," *Biometrika*, vol. 43, pp. 45–63, 1956.

[21] B. Efron and R. Thisted, "Estimating the Number of Unseen Species: How Many Words did Shakespeare Know?" *Biometrika*, vol. 63, pp. 435–447, 1976.

[22] F. Viger, A. Barrat, L. Dall'Asta, C.-H. Zhang, and E. Kolaczyk, "What is the Real Size of a Sampled Network? The Case of the Internet." in *Physical Review E*, vol. 75, 2007.

[23] M. Roughan, J. Tuke, and O. Maennel, "Bigfoot, Sasquatch, the Yeti and Other Missing Links," in *Proceedings of ACM IMC*, Vouliagmeni, Greece, October 2008.

[24] E. J. Candes and B. Recht, "Exact Matrix Completion via Convex Optimization," in *Foundations of Computational Mathematics*, vol. 9, 2009, pp. 717–772.

[25] B. Recht, M. Fazel, and P. Parrilo, "Guaranteed Minimum Rank Solutions to Linear Matrix Equations via Nuclear Norm Minimization," in *SIAM Review*, vol. 52, no. 3, 2010, pp. 471–501.

[26] 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.

[27] R. Beverly, A. Berger, and G. Xie, "Primitives for active internet topology mapping: Toward high-frequency characterization," in *Proceedings of ACM IMC*, Melbourne, Australia, November 2010.

[28] J. Ni, H. Xie, S. Tatikonda, and Y. R. Yang, "Efficient and Dynamic Routing Topology Inference from End-to-End Measurements," in *IEEE/ACM Transactions on Networking*, vol. 18, no. 1, February 2010, pp. 123–135.

[29] B. Eriksson, G. Dasarathy, P. Barford, and R. Nowak, "Toward the practical use of network tomography for internet topology discovery," in *Proceedings of IEEE INFOCOM*, San Diego, CA, March 2010.

[30] A. Bavier, M. Bowman, B. Chun, D. Culler, S. Karlin, S. Muir, L. Peterson, T. Roscoe, T. Spalink, and M. Wawrzoniak, "Operating System Support for Planetary-Scale Network Services," in *USENIX NSDI*, March 2004.

[31] "Route Views Project," http://www.routeviews.org/.

[32] B. Augustin, X. Cuvellier, B. Orgogozo, F. Viger, T. Friedman, M. Latapy, C. Magnien, and R. Teixeira, "Avoiding Traceroute Anomalies with Paris Traceroute," in *Proceedings of ACM IMC*, Rio de Janeiro, Brazil, October 2006.

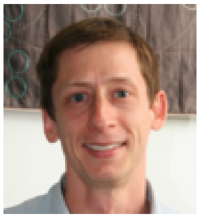
[33] M. Luckie, Y. Hyun, and B. Huffaker, "Traceroute Probe Method and Forward IP Path Inference," in *Proceedings of ACM IMC*, Vouliagmeni, Greece, October 2008.

[34] L. Wasserman, "All of Nonparametric Statistics (Springer Texts in Statistics)." Springer, May 2007.

[35] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer, 2001.

Brian Eriksson Brian Eriksson is a postdoctoral research associate at Boston University in the department of computer science. He received his B.S. degree (with distinction) from the University of Wisconsin in computer engineering in 2004; and his M.S. and Ph.D. in electrical engineering from the University of Wisconsin in 2006 and 2010, respectively. His research interests include the analysis of networked systems, Internet measurements, signal processing, and machine learning.

Paul Barford Paul Barford is an associate professor of computer science at the University of Wisconsin - Madison and Chief Scientist at Qualys, Inc. His research interests are in measurement and analysis of Internet traffic and topological structure; Internet security including anomaly and intrusion detection; power-aware networking focused on reducing energy demand and carbon footprint of Internet systems. He received his BS in electrical engineering from the University of Illinois at Urbana Champaign and his PhD in computer science from Boston University.



Joel Sommers Joel Sommers received the B.S. degree in mathematics and a second B.S. degree in computer science from Atlantic Union College, South Lancaster, MA, in 1995; the M.S. degree in computer science from Worcester Polytechnic Institute, Worcester. MA, in 1997; and the Ph.D. degree from the University of Wisconsin at Madison in 2007. He is currently an Assistant Professor of computer science with Colgate University, Hamilton, NY. His research interests are in measurement and analysis of network traffic and networked systems.

Robert Nowak Robert Nowak received the B.S. (with highest distinction), M.S., and Ph.D. degrees in electrical engineering from the University of Wisconsin- Madison in 1990, 1992, and 1995, respectively. He is the McFarland- Bascom Professor of Engineering at the University of Wisconsin-Madison. His research interests include signal processing, machine learning, imaging and network science, and applications in communications, bioimaging, and systems biology.