Network Positioning from the Edge

An empirical study of the effectiveness of network positioning in P2P systems

David R. Choffnes, Mario A. Sánchez and Fabián E. Bustamante
EECS, Northwestern University
{drchoffnes,msanchez,fabianb}@eecs.northwestern.edu

***Abstract*—Network positioning systems provide an important service to large-scale P2P systems, potentially enabling clients to achieve higher performance, reduce cross-ISP traffic and improve the robustness of the system to failures. Because traces representative of this environment are generally unavailable, and there is no platform suited for experimentation at the appropriate scale, network positioning systems have been commonly implemented and evaluated in simulation and on research testbeds. The performance of network positioning remains an open question for large deployments across hosts located at the edges of the network.**

This paper evaluates how four key classes of network positioning systems fare when deployed at scale and measured in P2P systems where they are used. Using 2 billion network measurements gathered from more than 43,000 IP addresses probing over 8 million other IPs worldwide, we show that network positioning exhibits noticeably worse performance than previously reported in studies conducted on research testbeds. To explain this result, we identify several key properties of this environment that call into question fundamental assumptions driving network positioning research.

I. Introduction

Network positioning systems have been proposed as a scalable way to determine the relative location of hosts in the network, measured in terms of latency or available bandwidth [1]. Most work in this area has focused on efficiently predicting latencies between hosts [2]–[4], and has been implemented on research testbeds to enable features such as closest server selection in CDNs and low-latency neighbor selection in DHTs.

Network positioning information can also benefit the growing number of large-scale P2P systems (e.g., streaming video, VoIP and file sharing [5]–[8]) that run on hosts located at the edges of the network (e.g., on desktops or appliances behind NAT boxes on residential links). Such information potentially enables clients to achieve higher performance, reduce cross-ISP traffic and improve the robustness of the system to failures.

Because traces representative of this environment are generally unavailable, and there is no platform suited for experimentation at the appropriate scale, the performance of network positioning remains an open question for such large deployments across hosts located at the edges of the network.

This paper evaluates how four key classes of network positioning systems fare when deployed and measured at the scale of real, popular P2P systems. For this study, we gathered a large, representative dataset based on information reported by hosts participating in the Vuze BitTorrent system [9] through an extension to this client, currently installed by hundreds of thousands of peers,

The Vuze BitTorrent client provides operational deployments of Vivaldi [2], Vivaldi version 2 [10] and CRP [11], in addition to a rich interface for accessing peers' positioning information. With Vivaldi running on approximately one million hosts online at any moment, this represents the largest deployment of any network positioning service. We sample Vivaldi network coordinates and CRP network positions, and perform network measurements to evaluate their accuracy. We additionally use the latency measurements between hosts to understand Meridian [4] and GNP [3] performance in this environment. Finally, we collect traceroute measurements between BitTorrent peers for diagnosing network positioning performance.

This paper makes the following contributions. First, we find that the accuracy of the network coordinate systems is significantly worse when used at the edge of the network than when evaluated from the perspective of a research testbed. We use empirical data and simulations to show that these systems exhibit large errors both in terms of milliseconds and relative differences between predicted and actual latencies. For example, we find that GNP exhibits a median error of 65 ms while the same for Vivaldi is at least 150 ms. Furthermore, the average relative errors for coordinate-based and direct-measurement positioning systems is at least twice as large as previously reported. While previous work [12] predicts large absolute and relative errors in coordinate systems, this work is the first to show the extent of these errors empirically.

Second, we show that this inaccuracy leads to significant loss in performance in the case of low-latency distributed hash tables (DHTs), which use network coordinates to guide neighbor selection. We find that DHTs using inaccurate network positioning information can select neighbors that are between one and two orders of magnitude worse than optimal in terms of latency.

Third, we explore the root causes of errors in network positioning in the P2P environment at an Internet scale. We find that these errors are in part explained by the inherent dimensionality of the worldwide latency space viewed from hosts at the edge, which is much larger than reported based on studies that use testbed deployments. We find direct evidence of this effect by studying geographic properties of paths between hosts, which are not easily modeled by surfaces of simple geometric shapes. Further, we find that the rates of triangle-inequality violations (TIVs) is much higher (up to 400%) and more widespread (covering over 99.5% of the source/destination pairs) than previously reported. We then use traceroute measurements to demonstrate how first- and

Class	Reference system
Landmark-based embedding	GNP [3]
Landmark-free embedding	Vivaldi [2] and Vivaldi v2 [10]
Direct measurement	Meridian [4]
Measurement reuse	CRP [11]

TABLE I
CLASSES OF NETWORK POSITIONING USED IN THIS STUDY.

last-mile links contribute significant variability (25% to 100% more variance than "core" links) to latencies measured in our environment.

Last, we suggest that future network positioning research focus on using measured network topology information to provide accurate relative distance estimation, based on our observations regarding challenges in predicting latency in this environment. To facilitate new research in network positioning, we will make our anonymized dataset publicly available. This data consists of approximately 2 billion latency samples, 30 million traceroute measurements and hundreds of millions of network position samples gathered during a two-week period.

The remainder of the paper is organized as follows. In the next section, we describe the four classes of network positioning approaches that we evaluate in this study. Section III provides details on our dataset and how we use it to evaluate positioning performance. We analyze the accuracy of network positioning and its impact on performance in Section IV, then explore sources of their errors in Section V. Based on this analysis, in Section VI we discuss future directions for research on network positioning systems in the P2P environment. We conclude in Section VII.

II. BACKGROUND

There is a rich body of work that addresses the design and implementation of network positioning systems [1]–[3], [11], [13]–[18]. While a number of recently proposed services aimed at providing more general metrics [19]–[22], network positioning systems remain the most scalable and widely used information plane services [21], particularly in the P2P domain. In this section, we describe four classes of network positioning systems that we cover in this study (Table I).

Landmark-based Embedding. Landmark-based systems estimate network distances to participating hosts by embedding their network locations in a multi-dimensional Euclidean space based on the hosts' distances to a given set of *landmarks*. The Global Network Positioning (GNP) system [3] provides efficient distance estimation by designating a relatively small set of landmarks that constantly measure latencies to each other and use this information to determine their positions in an N-dimensional Euclidean space. Each participating host then measures its latency to each landmark, combining this information with each landmark's position to estimate its own position. Based on simulations of a network topology containing over 1,000 nodes and an evaluation in PlanetLab the authors found that GNP had a median relative error of 0.08 and a 90th percentile relative error of 0.52. To reduce the computational cost of the GNP embedding algorithm, ICS [14] and Virtual Landmarks [16] replace it with a computationally cheaper approximation based on principal component analysis with a potential cost of lower accuracy. To avoid the load imbalance and lack of failure resilience created by a set of fixed landmarks, PIC [13] uses landmarks only for bootstrapping, while Lighthouse [15] avoids them altogether by combining multiple local coordinate systems into a global coordinate system through a transition matrix.

Landmark-free Embedding. Landmark-free systems, in contrast, fully decentralize the computation of network locations encoded in a low-dimensional coordinate space [2], [23]. Among these systems, the Vivaldi network positioning system [2] is the most widely deployed. It embeds network locations in an N-dimensional Euclidean space; each node maintains its own positions, measures RTT latencies to other hosts and periodically exchanges position information with them. A node recomputes its position by simulating a force from a spring corresponding to the error between the coordinate-based Euclidean distance and measured latency. The authors evaluate the accuracy of the approach using PlanetLab nodes and King-based [24] network distances between 1740 DNS servers and found that its error is competitive with GNP [3]. In a follow-up study, Ledlie et al. [10] showed that the accuracy of this system was much lower "in the wild", as measured from PlanetLab nodes participating in the Vuze Vivaldi implementation. They proposed and implemented several features that improve performance in this environment, finding that accuracy improves by 43%.

Direct Measurement. Despite the success of these systems, recent studies have called into question the usefulness of network coordinates [25]. For example, Wong et al. [4] note that embedding errors from network coordinates always leads to suboptimal peer selection and instead propose Meridian, a structured approach to direct measurement. To ensure scalability, Meridian organizes nodes into an overlay consisting of "rings" of nodes that locate nearby peers in $\log(N)$ steps, where N is the number of nodes participating in the system. Using a simulation-based evaluation with King-based latencies between 2500 DNS servers and a deployment in PlanetLab with 166 nodes, the authors show that the accuracy of Meridian is significantly better than that of approaches using virtual coordinates.

Measurement Reuse. Direct measurement provides high accuracy, but even structured approaches to latency measurement can incur significant overhead for large systems. The CDN-based Relative Positioning (CRP) approach [11] is based on the observation that relative network positioning is sufficient for many applications [26] and proposes a low-cost technique to provide this service by reusing measurements performed by content distribution networks. CRP takes advantage of the fact that CDNs are already measuring the Internet and make this information available in the form of DNS redirections. By comparing nodes' redirection behavior, CRP can determine whether nodes are relatively near one another. Specifically, CRP uses the notion of a *ratio map* to encode the percent of time that a node has been redirected toward a particular set of replica servers and compares these maps using the cosine similarity metric. The authors use PlanetLab nodes to show that CRP provides accuracy comparable to Meridian.

Unique targets	
Hosts	19,765
Unique IPs	43,674
Host behind middleboxes	$\approx$ 86%
Coverage	
Prefixes	7,975
ASes	1,625
Countries	129

TABLE II
SUMMARY OF VANTAGE POINTS DURING THE 15-DAY PERIOD IN OUR STUDY.

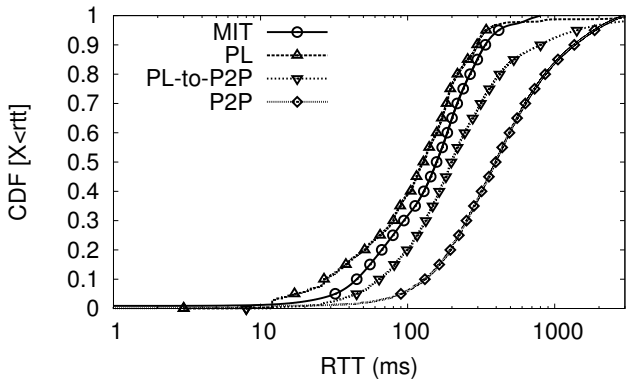


Fig. 1. CDFs of latencies from different measurement platforms (semilog scale). Our measurement study exclusively between peers in Vuze (labeled P2P) exhibits double the median latency "in the wild" (labeled PL-to-P2P).

III. DATASET AND METHODOLOGY

Our study focuses on measurements between P2P end systems primarily located at the edges of the network, while all previous evaluations of network positioning were based on data gathered between and from PlanetLab nodes. We present results based on measurements collected from more than 40,000 IPs broadly distributed worldwide, with between 6,500 and 7,100 IPs online per day.

Table II summarizes key characteristics of the vantage points used in this study. For comparison, note that the number of vantage points online during the 15-day period of our study is five times greater than *all* of the vantage points participating in DIMES [22] since 2004. Our users are located in more than an order of magnitude more BGP prefixes than those available from PlanetLab [27]. Finally, note that because the peers in our study are often located behind middleboxes at the edges of the network, they allow us to measure portions of the Internet not visible when using traditional measurement techniques [28]. The following paragraphs describe our dataset and how we use it to evaluate existing network positioning approaches.

A. Dataset

The dataset used in this study was gathered from users running the Ono plugin [6] for the Vuze BitTorrent client. To compare each technique's distance estimate with ground truth, our software performs latency measurements between connected peers. Our software also issues traceroute probes to connected hosts for discovering topological information. The data used in this study was collected during the period of June 10 to June 25, 2008 and will be made publicly available in an anonymized format.

A distinctive aspect of our measurement approach is that it records measurements at the scale and in the environment where network coordinates are intended to be used. These latencies (P2P), shown in Fig. 1, are generally much larger than those from MIT King [2] and PlanetLab (PL). In fact, the median latency in our dataset is twice as large as reported by the study from Ledlie et al. [10], which used PlanetLab nodes to probe Vuze P2P users (PL-to-P2P).

During the observation period our collaborative measurement infrastructure collected over 100 million samples from peers per day. The dataset used in this paper consists of more than 1.41 billion Vivaldi samples, 60 million CRP ratios, 2 billion total latency samples and 33 million traceroute measurements. The Vivaldi samples were recorded from 43,674 source IP addresses; the CRP ratios are derived both from a host's local ratios and from those gathered by issuing remote DNS lookups, covering more than 3.3 million distinct IP addresses.

Finally, we collect traceroute measurements to a random set of peers connected to each measurement host. We use the host's built-in `traceroute` command with the default settings, and at most one measurement is performed at a time. Specifically, we collect the ordered set of IP hops and their latencies as reported by the output of the command. During the measurement period, we collected more than 30 million path measurements starting at more than 70,000 first-hop router IP addresses.

B. Latency Matrix

The ping measurements that we collect can be used directly for evaluating the live performance of CRP and Vivaldi. To broaden the scope of our study we construct a matrix of latencies, enabling us to simulate performance for the GNP and Meridian systems in this context.

Despite a relatively large deployment of our measurement software, it is often the case that there is a latency measurement from host A to B but *not* from host B to A. This occurs because latency measurements are performed independently by participating hosts, and destinations are selected essentially at random. As a result, our latency measurements form a large, sparse matrix where each row/column represents an individual IP address. To put the size of this matrix in context, there are more than 376 billion (43,674 x 8,611,489) possible entries.

To be useful for analysis and simulation, we must extract a dense submatrix. To create a manageable-sized matrix, we choose to generate one where the elements are routable BGP prefixes. We base this decision on the assumption that BGP prefixes represent blocks of IP addresses in the same network. Thus, such a matrix approximates latencies between pairs of *networks* instead of pairs of *IP addresses*.

Based on this approach, we bucket our measurements into source and destination routable BGP prefixes (according to [29]), using the minimum observed RTT for each matrix element. We use the *minimum* to approximate the intrinsic

network latency and filter out dynamic changes that make distance estimation more challenging. This simplifying assumption alleviates the need for techniques such as latency smoothing [30], and intuitively should lead to favorable results for network positioning systems.

Because this approach still yields a sparse 6380x72343 matrix, we use the square submatrix and iteratively remove rows and columns that contain the largest number of empty elements until a sufficiently dense submatrix remains. There is a sharp drop-off in the number of elements in an n%-full matrix as n approaches 100, so we use $n = 95$. We found that different approaches to filling the remaining empty matrix elements (e.g., using the median, average or minimum value for each row or column) do not significantly affect the results when the matrix is nearly full. In this study, we fill the empty elements with the median value for a given row to preserve that particular statistical property. We use this process to generate a 479x479 matrix. The rows represent ISPs in North America, Europe, Asia (including the Middle East), South America and Oceania.

IV. Performance from End Systems

In this section, we evaluate the accuracy of network positioning systems in a P2P environment and their impact on the performance of an example application that uses them.

For evaluating GNP performance, we use the authors' simulation implementation. The results are based on three runs of the simulation, each using a randomly chosen set of 15 landmarks, 464 targets and an 8-dimensional coordinate space. Since the original code for Meridian is not publicly available, we wrote our own discrete-event simulator for it, with guidance from the authors. We will make our Meridian simulation code publicly available. Our simulation settings are proportional to those in the original evaluation, with 379 randomly selected Meridian nodes, 100 target nodes, 16 nodes per ring and 9 rings per node. Our results are based on four simulation runs, each of which performs 25,000 latency queries.

A. Accuracy

We begin our analysis by evaluating the accuracy of GNP and of the Vuze Vivaldi implementations in terms of errors in predicted latency. Meridian and CRP are omitted here because they do not provide quantitative latency predictions. Figure 2 presents the cumulative distribution function (CDF) of errors on a semilog scale, where each point represents the absolute value of the *average* error from one measurement host. We find that GNP has lower measurement error (median is 59.8 ms) than the original Vivaldi implementation (labeled V1, median error is $\approx$ 150 ms), partially due to GNP's use of fixed, dedicated landmarks. Somewhat surprisingly, Ledlie et al.'s Vivaldi implementation (labeled V2) has slightly larger errors in latency (median error is $\approx$ 165 ms) than GNP and V1; however, we show in the next paragraph that its relative error is in fact smaller.

Relative error, the difference between the expected and measured latency divided by the measured latency, is a better

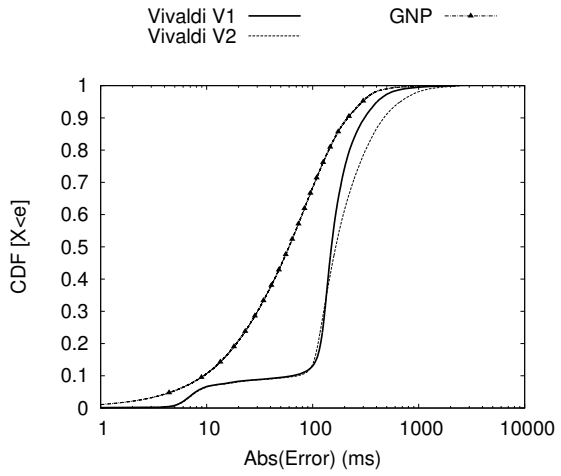


Fig. 2. Absolute value of errors between estimated and measured latencies, in milliseconds. The median error for GNP is about 60 ms whereas the same for Vivaldi V1 and V2 are 150 and 165 ms, respectively.

measure of accuracy for network positioning systems. To compute relative errors, we first calculate the absolute value of the relative error between Vivaldi's estimated latency and the ping latency for each sample, then find the average of these errors for each client running our software.

In Fig. 3, we plot a CDF of these values; each point represents the average relative error for a particular client. For Vivaldi V1, the median relative error for each node is approximately 74%, whereas the same for V2 is 55%. Both errors are significantly higher than the 26% median relative error reported in studies based on PlanetLab nodes [10].

Interestingly, the median error for Vivaldi V2 is approximately the same as for GNP, indicating that decentralized coordinates do not significantly hurt relative performance.

Finally, because Meridian and CRP do not predict distances, Fig. 3 plots the relative error for the closest peers found by all the network positioning systems studied. Meridian finds the closest peer correctly approximately 20% of the time while CRP can locate the closest peer more than 70% of the time.

B. Latencies in P2P environments

It is possible for client P2P traffic to interfere with the latency measurements. For instance, queuing delays introduced by natural P2P traffic could significantly increase delays in latency measurements and alter the perceived structure of the Internet latency space. To evaluate the impact of this traffic, we compare the set of latencies measured when our clients are actively downloading with those collected when they are *idle* (i.e., having upload/download rates 4 KB/s or less).

Figure 4 shows the CDF of latencies for these two sets, where each point represents the *average* of latencies from one source to all of its destinations. To ensure enough diversity in the latency measurements, we include only hosts that perform at least 50 measurements. While the idle latencies are not surprisingly smaller than those in the complete dataset, the difference in median latencies is less than 10%. Of course, since we can control only for idle activity at the source measurement host, not the remote hosts being measured,

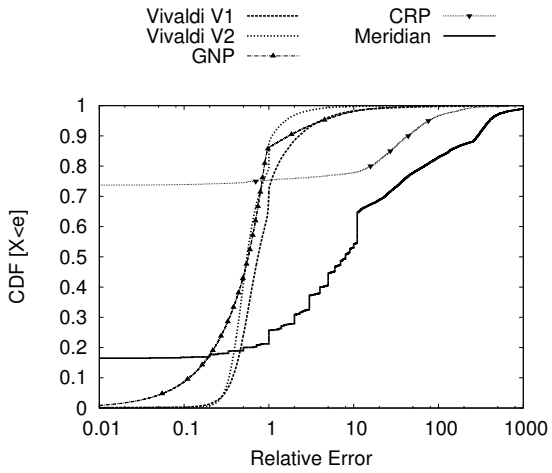


Fig. 3. Absolute value of *relative* errors between estimated and measured latencies. Vivaldi V1 and V2 exhibit median errors that are triple or double previously reported; however, these errors are similar to those in GNP. For Meridian and CRP, which do not predict distances, we use the error for the closest host found by each approach.

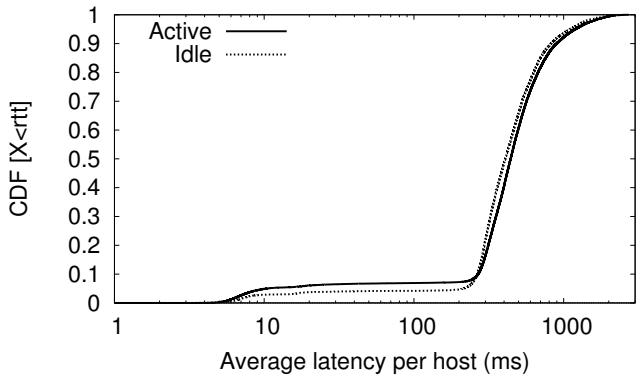


Fig. 4. CDF of average latencies for vantage points when idle and actively downloading. The distributions are nearly identical, suggesting that downloading behavior has a minimal impact on observed latencies.

we estimate a worst-case 20% difference in median latency. Because the distributions of latencies between idle and active hosts are similar and their differences are not relatively large, we do not expect these differences to significantly impact relative errors in latency prediction nor the structure of the latency space.

C. Impact on Applications

As pointed out by Lua et al. [12], relative error in latencies alone do not necessarily predict the quality of network positioning as experienced by the user. We now focus on whether the errors we reported in the previous paragraphs do indeed negatively affect application performance (e.g., for CDN replica server selection, nearby peer selection in P2P file sharing, etc.).

This section focuses on the case of distributed hash tables (DHTs), which can use nearby hosts to reduce the time to perform read and write operations. In this case, a positioning system need only guarantee that nodes closer to the local host have smaller estimated distances than those farther away.

One way to measure this is the relative application-level penalty (RALP) metric initially proposed by Pietzuch et al. [31]. This metric measures the latency penalty incurred by applications using network positioning to select the closest N peers, compared to optimal selection.

To calculate RALP for a host, we first create a set of measured latencies, G, between this host and a set of other hosts, ordered according to "ground-truth" ping measurements. We then create a corresponding set of measured latencies, P, ordered by the hosts' proximity according to the positioning systems. For Meridian and CRP, which do not predict distances, we order the closest peers they found based on their measured latencies.

We then find the average RALP for each measurement node using the following equation:

$$1/n \cdot \sum_{i=1}^{n} (p_i - g_i)/g_i$$

where n is the number of nodes being measured and i is the index in the ordered sets.

Figure 5 shows a CDF of the average RALP values for each measurement node when comparing the Meridian-selected node, the best 10 CRP-selected nodes and the 10 nodes ordered by estimated distance for the other positioning systems.

Note that the vast majority of RALP values is greater than 1, indicating that errors in the network positioning system lead to significant loss in performance for the DHT that uses it. For example, the median RALP for Vivaldi V2 when assessing the closest 10 nodes is 26.9, meaning that for half the peers in our study, the average latency to Vivaldi-driven peers is about 27 times worse than optimal. By comparison, Ledlie et al. [10] saw median RALP values near 0.4 when measuring from PlanetLab. Also note that the median RALP in our study is much larger than the *average per-peer* relative errors shown in the previous section – this occurs because the set of nearest nodes according to Vivaldi often have significantly larger latency than the "ground-truth" nearest nodes. Finally, Meridian and CRP exhibit similar and comparatively good performance, showing that on average these systems locate close nodes most of the time.

Based on the empirical results from our study, existing network positioning systems not only exhibit large errors in predictions, but those errors significantly impact application performance in large-scale P2P environments. In the next section, we explore why this is the case.

V. Sources of Error

Many authors have pointed out issues that impair accuracy in network positioning systems, including churn, coordinate drift, corruption, latency variance and intrinsic errors. While solutions have been proposed to address the first three problems [10], [32], [33], this section focuses on variance and intrinsic errors in latency prediction, as they represent fundamental challenges for latency-based approaches to network positioning.

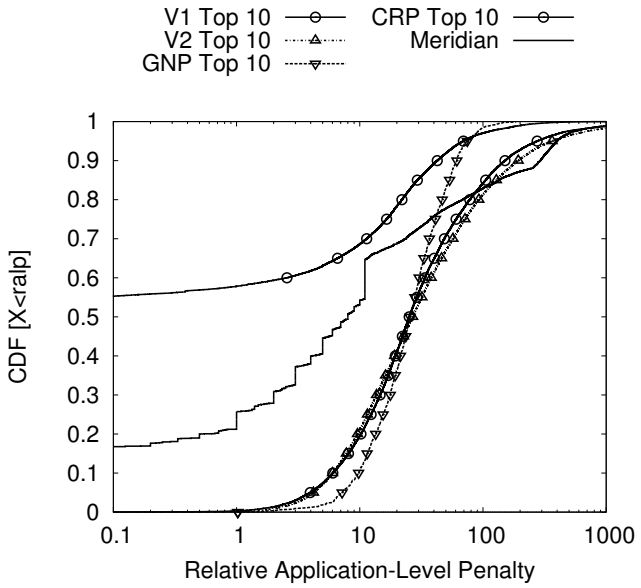


Fig. 5. Relative application-level penalty for using network positioning. The vast majority of values are greater than one and the median values indicate order-of-magnitude loss in performance.

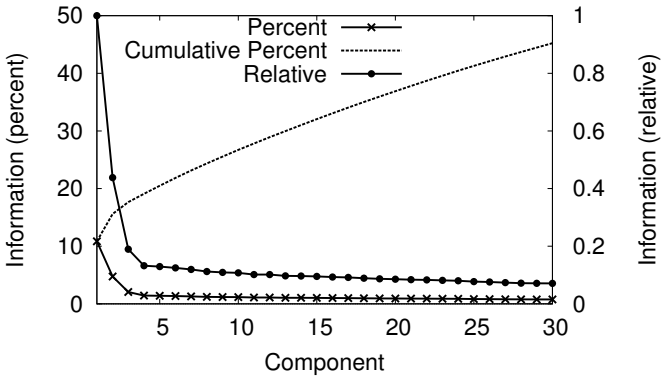


Fig. 6. Plot indicating portion of variance captured by each principal component. The first five components capture only a small portion (20%) of the total variance.

A. Network Embedding

Starting from a matrix of network latencies for a collection of hosts, early work on network positioning has relied on the use of principal component analysis (PCA) to estimate the number of linear combination of elements sufficient to capture most of its variance (e.g., [16]). If the vast majority of the variance is modeled by a few principal components, then a small number of dimensions may be sufficient to use in embedding network distances in an Euclidean space. This analysis has been previously used to select 2, 4 or 7 dimensions [2], [10], [16].

We perform the same PCA analysis on the latency matrix described in Section III. Figure 6 presents a scree plot of the relative variance captured by each of the first 30 components, in descending order of the amount of variance they capture. The figure contains curves for (*i*) the percent of the *total* variance captured by each component (**Percent**, left y-axis), (*ii*) the *relative* variance captured by each component normalized by the value for the first component (**Relative**, right y-axis) and (*iii*) the *cumulative* variance captured by all components with rank less than or equal to x (**Cumulative Percent**, left y-axis).

Traditionally, one uses the first two curves to identify the inherent dimensionality of the space by locating the "knee" in the curve. While the knee appears to occur around the 4th or 5th component, these first few components capture only a small amount (20%) of the variance. Although the values quickly diminish for other components, the curve exhibits a long tail. For instance, 9 components are required to capture 25% of the variance and at least 37 components are required to capture 50% of the variance.

Previous work in PlanetLab has shown much higher variance captured by small numbers of coordinates, which can be explained by the platform's relatively small number of nodes located near the "core" of the Internet. To hint at the effect of evaluating a smaller number of networks, we further reduced our matrix to 274x274 routable prefixes (99% full). After running PCA on this matrix, the amount of variance captured by the first component *nearly doubles* and the variance for the first 5 components increases by 35%. This suggests two effects: analysis on matrices formed by limited vantage points underestimates the complexity of the Internet delay space; however, even with the smaller matrix based on latencies from the "edge" of the network, the majority of the variance is not captured by the first few components. We posit that this additional complexity is one of the primary reasons why network coordinates yield such large errors at scale.

Another argument that has been used to justify low-dimensional coordinate spaces is that the Internet is "flat." In particular, Ledlie et al. [10] provide latency CDFs indicating that the RTTs from Asia to Europe are larger than those from Asia to North America. From this, the authors conclude that paths from Asia to Europe flow through North America suggesting that the "world is flat" and the Internet indeed embeds in a low dimensional Euclidean space.

We reexamined the latency distributions, grouped by source and destination continent, from our dataset. We then use traceroute measurements between P2P users to analyze the actual data flow. Due to space constraints, we focus on the single case mentioned above: paths from Asia to Europe. Other source-destination pairs exhibit similar complexity in the topography.

Fig. 7 plots the CDF of latencies from source nodes in Asia to destination nodes grouped by continents. Compared with the measurements reported by Ledlie et al. [10], while the shapes of the curves differ, we observe a similar effect with latencies from Asia to North America being generally smaller than those from Asia to Europe.

However, while latencies hint at path properties, they reveal no actual information about how a packet travels from source to destination. Figure 8 shows the percentage of hops from Asia to Europe that occur in a particular continent. To generate this plot, we first convert router IP addresses to countries according to information provided by Team Cymru [29], then we translate countries to continents according to the UN

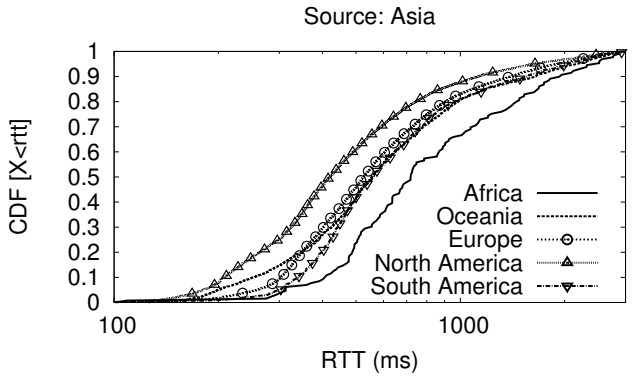


Fig. 7. CDF of latencies from source nodes in Asia to destination nodes, grouped by destination continent. Similar to previous work, latencies from Asia to North America are generally smaller than those from Asia to Europe.

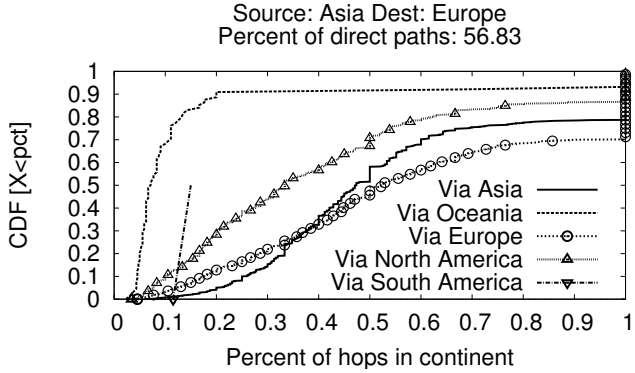


Fig. 8. Plot indicating portion of portion of paths that pass through various continents when the source is in Asia and the destination is in Europe. This clearly shows that the network is not "flat" because data over half of the paths are directly between Asia to Europe, while the remaining ones take diverse paths.

Statistics Division [34]. First note that over half (nearly 57%) of the measured *paths between Asia and Europe did not pass through a third continent*, indicating direct paths between those continents are common (albeit with larger latencies than those to North America) and thus latency is a poor predictor for data forwarding path characteristics. Further, while a significant portion of paths do pass through North America, there are other paths that include Oceania and even South America. These properties make it challenging to capture the structure of the network using a small number of coordinates.

B. Triangle Inequalities

Triangle inequality violations (TIVs) in the Internet delay space occur when the latency between hosts A and B is larger than the sum of the latency from A to C and C to B ($A \neq B \neq C$). This is caused by factors such as network topology and routing policies. Wang et al. [35] demonstrate that TIVs can significantly reduce the accuracy of network positioning systems.

We performed a TIV analysis on our dataset and found that over 13% of the triangles had TIVs (affecting over 99.5% of the source/destination pairs). Figure 9 visualizes the

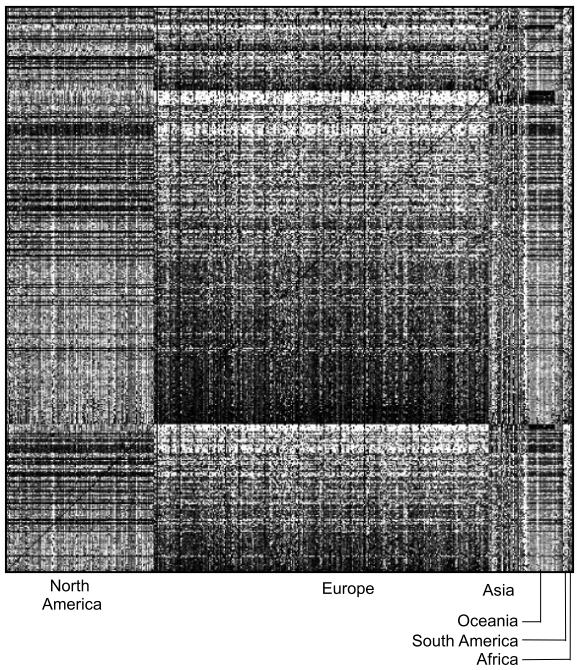


Fig. 9. Map of severity of TIVs in our measured latency space, where rows and columns from the same continent are grouped together. A white point represents the most severe TIV. While there are clear definitions of these continental clusters, there are severe TIVs scattered throughout the matrix, most of which do not exhibit an easily identifiable pattern.

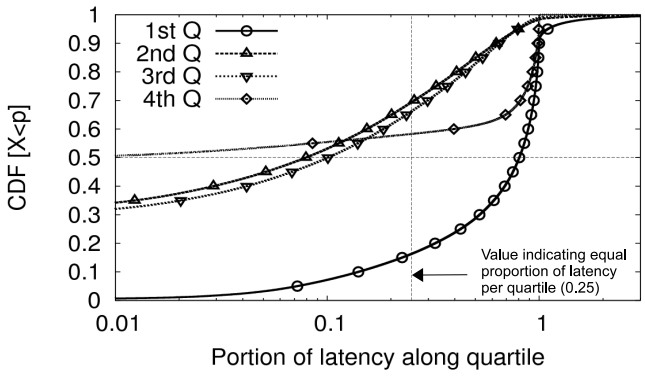
severity of these TIVs, where lighter colored points indicate more severe TIVs and rows/columns belonging to the same continent are grouped together (as done by Wang et al. [35]). The figure shows that some networks experience few TIVs (dark lines), some experience a large number (light lines) and many experience a significant number in non-uniform patterns.

Compared to TIV rates reported in an analysis of datasets from Tang and Crovella [16], TIVs rates in the P2P environment we studied are between 100% and 400% higher, and the number of source/destination pairs experiencing TIVs in our dataset (nearly 100%) is significantly greater than the 83% reported by Ledlie et al. [10]. These patterns for TIVs and their severity hints at the challenges in accounting for TIVs in coordinate systems. While an important first step toward improving network positioning systems is to make them TIV-aware [35], it remains to be seen whether this approach can yield sufficient coverage and performance for client applications.

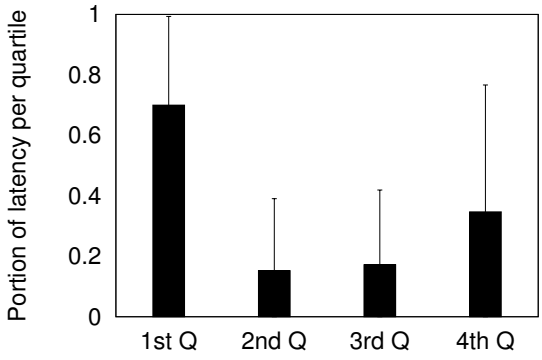
C. First- and Last-Mile Issues

It is well known that last-mile links often have poorer quality than the well provisioned links in transit networks. The problem is particularly acute in typical P2P settings. However, most of today's network positioning systems either ignore or naively account for this effect.

To understand the risks of ignoring this issue in a latency-based network positioning system for a P2P environment, Fig. 10(a) plots CDFs of the portion of end-to-end latency (log scale) along quartiles of the IP-level path between the measured hosts; Fig. 10(b) provides a bar graph showing the



(a) CDF.



(b) Mean and standard deviation.

Fig. 10. Portion of end-to-end latency contained in each quartile of the IP-level path between endpoints. The top graph shows a CDF of these values for each quartiles; the bottom one shows the mean and standard deviations for each quartile. The figures show that the first quartile of the path contains the largest portion of latency most of the time, and the significant number of values greater than 1 indicate large variance in latencies in this portion of the path.

mean and standard deviation for each quartile. We determine the per-hop latencies from our traceroute measurements.

If the latency were evenly distributed among IP hops along a path, the curves would center around $x = 0.25$. In contrast, the first quartile (which is very likely to contain the entire first mile) stands out from the rest, containing disproportionately large fractions of the total end-to-end latency. For instance, when looking at the median values, the 1st quartile alone captures 80% of the end-to-end latency. The middle two quartiles, in contrast, each account for only 8%. Also note that the first quartile (and a significant fraction of the last quartile) has a large number of values close to and larger than 1. This demonstrates the variance in latencies along these first and last miles, where measurements to individual hops along the path can yield latencies that are close to or larger than the total end-to-end latency (as measured by probes to the last hop). In fact, more than 10% of the 1st quartile samples have a ratio greater than 1. While Vivaldi uses "height" to account for (first- and) last-mile links [2], this analysis suggests that a single parameter is insufficient due to the large and variable latencies in a large-scale P2P environment.

Finally, we note that there is asymmetry between the first and last quartiles of the IP-level hops in terms of the amount of latency in each quartile. Intuitively, the first and last miles of paths should exhibit similar characteristics on average. The primary cause for this discrepancy is that most of the hosts in our dataset ($> 80\%$) reside behind middleboxes, which commonly block inbound traceroute probes. Thus, while the probes are allowed traverse any middleboxes near the source (in the outbound direction), they often cannot access the last mile of the path.

VI. Discussion

In the previous sections, we demonstrated the challenges involved in predicting latency between an arbitrary set of heterogeneous end systems in the wide area. First, we showed that coordinate-based systems that model the latency space using few dimensions can lead to poor performance for applications requiring low latency. Even Meridian, which uses direct measurements, can yield large errors because its structure is based on Euclidean assumptions about the structure of the latency space. Not surprisingly, the key reasons for these errors are related to latencies as measured from the edge of the network – namely, large variance over short time scales and TIVs in the delay space. These properties pose an enormous challenge to any positioning system that makes simplistic assumptions (implicit or otherwise) regarding the structure of the underlying paths forwarding data packets.

In light of these results, we believe that the network *topology* is a more appropriate abstraction for building a networking positioning system in this context. The topology not only models the underlying structure for intrinsic network distance, it is also relatively stable over long time periods [36]. While previous work has proposed using structural information for estimating arbitrary latencies [17], [18], we believe that it is sufficient for a positioning system to provide only *relative* proximity information. By relaxing the requirement for latency estimation, we can avoid the intrinsic errors from embeddings and sidestep challenges to latency estimation such as jitter and cross traffic common in P2P systems. For example, we observe that peers can discover their *local topology* (e.g., the set of edge routers in their provider network, or those in their provider's upstream networks) at relatively low cost – those peers seeing similar local topologies are likely to be near one another. We are currently using this approach to develop a fully decentralized, efficient and scalable approach to network positioning based on local topology information gathered from infrequent traceroute measurements.

VII. Conclusion

In this paper we evaluates how key classes of network positioning systems fare when deployed at scale and measured in P2P systems where they are used. Our results, based on the largest measurement dataset for these environment, show that network positioning exhibits noticeably worse performance than previously reported in studies conducted on research testbeds. After exploring the root causes for this inaccuracy, we found that the Internet delay space is inherently difficult to

model and it contains widespread triangle-inequality violations that frustrate distance estimation. Based on this result, we argue for *relative* network positioning that relies on local topology information instead of latencies.

REFERENCES

[1] P. Francis, S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang, "IDMaps: A global Internet host distance estimation service," *IEEE/ACM Transactions on Networking*, vol. 9, no. 5, October 2001.

[2] Dabek, Cox, Kaashoek, and R. Morris, "Vivaldi: A decentralized network coordinate system," in *Proc. of ACM SIGCOMM*, 2004.

[3] T. Ng and H. Zhang, "Predicting Internet network distance with coordinates-based approaches," in *Proc. of IEEE INFOCOM*, 2002.

[4] B. Wong, A. Slivkins, and E. Sirer, "Meridian: A lightweight network location service without virtual coordinates," in *Proc. of ACM SIGCOMM*, 2005.

[5] M. Adler, R. Kumary, K. Rossz, D. Rubenstein, T. Suel, and D. D. Yaok, "Optimal peer selection for P2P downloading and streaming," in *Proc. of IEEE INFOCOM*, 2005.

[6] D. R. Choffnes and F. E. Bustamante, "Taming the torrent: A practical approach to reducing cross-ISP traffic in P2P systems," in *Proc. of ACM SIGCOMM*, 2008.

[7] K. Gummadi, R. Gummadi, S. Gribble, S. Ratnasamy, S. Shenker, and I. Stoica, "The impact of DHT routing geometry on resilience and proximity," in *Proc. of ACM SIGCOMM*, 2003.

[8] M. J. Freedman, E. Freudenthal, and D. Mazières, "Democratizing content publication with coral," in *Proc. of USENIX NSDI*, 2004.

[9] Vuze, Inc., "Vuze," January 2009, http://www.vuze.com.

[10] J. Ledlie, P. Gardner, and M. Seltzer, "Network coordinates in the wild," in *Proc. of USENIX NSDI*, 2007.

[11] A.-J. Su, D. Choffnes, F. E. Bustamante, and A. Kuzmanovic, "Relative network positioning via CDN redirections," in *Proc. of the ICDCS*, 2008.

[12] E. K. Lua, T. Griffin, M. Pias, H. Zheng, and J. Crowcroft, "On the accuracy of embeddings for internet coordinate systems," in *Proc. of IMC*, 2005.

[13] M. Costa, M. Castro, A. Rowstron, and P. Key, "PIC: Practical internet coordinates for distance estimation," in *Proc. of the ICDCS*, 2004.

[14] H. Lim, J. C. Hou, and C.-H. Choi, "Constructing internet coordinate system based on delay measurement," in *Proc. of IMC*, 2003.

[15] M. Pias, J. Crowcroft, S. R. Wilbur, T. Harris, and S. N. Bhatti, "Lighthouses for scalable distributed location," in *Proc. of IPTPS*, 2003.

[16] L. Tang and M. Crovella, "Virtual landmarks for the internet," in *Proc. of IMC*, 2003.

[17] H. V. Madhyastha, T. Anderson, A. Krishnamurthy, N. Spring, and A. Venkataramani, "A structural approach to latency prediction," in *Proc. of IMC*. New York, NY, USA: ACM, 2006, pp. 99–104.

[18] B. Maggs, "Challenges in engineering the world's largest content delivery network," October 2008, http://www.aqualab.cs.northwestern.edu/HotWeb08/program.html.

[19] H. V. Madhyastha, T. Isdal, michael Piatek, C. Dixon, T. Anderson, A. Kirshnamurthy, and A. Venkataramani, "iPlane: an information plane for distributed systems," in *Proc. of the USENIX Operating Systems Design and Implementation (OSDI)*, 2006.

[20] A. Nakao, L. Peterson, and A. Bavier, "A routing underlay for overlay networks," in *Proc. of ACM SIGCOMM*, August 2003.

[21] H. V. Madhyastha, E. Katz-Bassett, T. Anderson, and A. Venkataramani, "iPlane Nano: path prediction for peer-to-peer applications," in *Proc. of USENIX NSDI*, 2009.

[22] Y. Shavitt and E. Shir, "DIMES: let the Internet measure itself," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 5, Oct. 2005.

[23] Y. Shavitt and T. Tankel, "Big-bang simulation for embedding network distances in euclidean space," *IEEE/ACM Transactions on Networking*, vol. 12, no. 6, 2004.

[24] K. P. Gummadi, S. Saroiu, and S. D. Gribble, "King: Estimating latency between arbitrary Internet end hosts," in *Proc. ACM IMW*, November 2002.

[25] R. Zhang, C. Tang, Y. C. Hu, S. Fahmy, and X. Lin, "Impact of the inaccuracy of distance prediction algorithms on Internet applications - an analytical and comparative study," in *Proc. of IEEE INFOCOM*, 2006.

[26] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, "Topologically-aware overlay construction and server selection," in *Proc. of IEEE INFOCOM*, June 2002.

[27] PlanetLab, "Planetlab: An open testbed for developing, deploying, and accessing planetary-scale services," http://www.planet-lab.org/.

[28] M. Casado, T. Garfinkel, W. Cui, V. Paxson, and S. Savage, "Opportunistic measurement: Extracting insight from spurious traffic," in *Proc. of HotNets*, November 2005.

[29] Team Cymru, "The Team Cymru IP to ASN lookup page," http://www.cymru.com/BGP/asnlookup.html.

[30] J. Ledlie, P. Pietzuch, and M. Seltzer, "Stable and accurate network coordinates," in *Proc. of the ICDCS*, 2006.

[31] P. Pietzuch, J. Ledlie, and M. Seltzer, "Supporting network coordinates on PlanetLab," in *Proc. of WORLDS*, 2005.

[32] M. Freedman, K. Lakshminarayanan, and D. Mazires., "OASIS: Anycast for any service," in *Proc. of USENIX NSDI*, May 2006.

[33] M. A. Kaafar, L. Mathy, T. Turletti, and W. Dabbous, "Virtual networks under attack: disrupting internet coordinate systems," in *Proc. of ACM CoNEXT*, 2006.

[34] United Nations Statistics Division, "Composition of macro geographical (continental) regions, geographical sub-regions, and selected economic and other groupings," revised 31 January 2008, http://unstats.un.org/unsd/methods/m49/m49regin.htm.

[35] G. Wang, B. Zhang, and T. S. E. Ng, "Towards network triangle inequality violation aware distributed systems," in *Proc. of IMC*, 2007.

[36] J. Li, M. Guidero, Z. Wu, E. Purpus, and T. Ehrenkranz, "Bgp routing dynamics revisited," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 2, 2007.