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 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 [6]. Network positioning information has been used in a growing number of large-scale P2P systems [1], [3], [8], [9] that run on hosts located at the edges of the network (e.g., on desktops or appliances behind NAT boxes on residential links). Because traces representative of this environment are generally unavailable, and there is no platform suited for experimentation at the appropriate scale, the corresponding performance of network positioning remains an open question.

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 [22] through an extension to this client, currently installed by hundreds of thousands of peers.

The Vuze BitTorrent client provides operational deployments of Vivaldi [5], Vivaldi version 2 (Pyxida) [11] and CRP [19], in addition to a rich interface for accessing peers' positioning information. 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 [24] and GNP [13] 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. 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. Third, we explore the root causes of errors in network positioning in the P2P environment at an Internet scale, based on latency and topology measurements.

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 positions 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. Sec. 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 Sec. IV, then explore sources of their errors in Sec. V.

II. Background

There is a rich body of work that addresses the design and implementation of network positioning systems [5], [12], [13], [19], [20]. In this section, we describe four classes of network positioning systems that we cover in this study.

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 set of landmarks. The Global Network Positioning (GNP) system [13] provides efficient implementation of this approach. *Landmark-free systems*, in contrast, fully decentralize the computation of network locations encoded in a low-dimensional coordinate space [5], [18]. Among these systems, the Vivaldi network positioning system [5] is the most widely deployed.

Despite the success of these systems, recent studies have called into question the usefulness of network coordinates [25]. For example, Wong et al. [24] note that embedding errors from network coordinates always leads to suboptimal peer selection and instead propose Meridian, a structured approach to *direct measurement*.

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

TABLE I
SUMMARY OF VANTAGE POINTS DURING THE 15-DAY PERIOD.

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 [19] is based on the observation that relative network positioning is sufficient for many applications [16] and proposes a low-cost technique to provide this service by *reusing measurements* performed by content distribution networks.

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 I 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 [17] since 2004. Our users are located in more than an order of magnitude more BGP prefixes than those available from PlanetLab [15]. 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 [2]. The following paragraphs describe our dataset and how we use it to evaluate existing network positioning approaches. For a more detailed description, see the associated technical report [4].

A. Dataset

The dataset used in this study was gathered from users running the Ono plugin [3] 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 [5] and PlanetLab (PL). In fact, the median latency in our dataset is twice as large as reported by the study from Ledlie et al. [11], which used PlanetLab nodes to probe Vuze P2P users (PL-to-P2P).

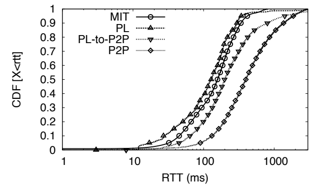

Fig. 1. CDFs of latencies from different measurement platforms (semilog scale). Our measurement study exclusively between peers in Vuze (labeled P2P) exhibits double the median latency "in the wild" (labeled PL-to-P2P).

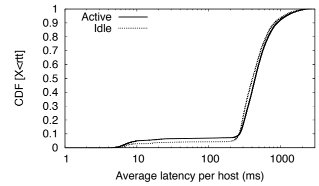

Fig. 2. CDF of average latencies for vantage points when idle and actively downloading. The distributions are nearly identical, suggesting that downloading behavior has a minimal impact on observed latencies.

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 IPs; 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 IPs.

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. During the measurement period, we collected more than 30 million path measurements starting at more than 70,000 first-hop router IPs.

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.

To evaluate the limits of network positioning performance in terms of intrinsic latencies between routable address blocks, we construct a latency matrix of source and destination *routable BGP prefixes* (according to [21]), using the minimum observed RTT for each matrix element. 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 $n = 95$, resulting in 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. Our Meridian 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 3 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 ≈ 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 ≈ 165 ms) than GNP and V1; however, we show in the next paragraph that its relative error is in fact smaller.

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. 4, 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 [11].

Finally, because Meridian and CRP do not predict distances, Fig. 4 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 2 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% and the curve shapes are similar. As such, we believe the impact of P2P traffic is not significant.

C. Impact on Applications

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.

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. [14]. 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 $1/n \cdot \sum_{i=1}^{n}(p_i - g_i)/g_i$, 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 for coordinate systems is greater than 1, indicating that errors in the 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. [11] 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.

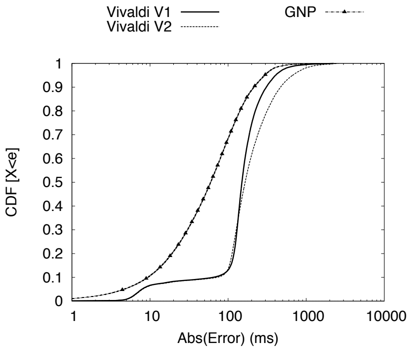

Fig. 3. Absolute value of errors between estimated and measured latencies, in milliseconds. The median error for GNP is about 60 ms whereas the same for Vivaldi V1 and V2 are 150 and 165 ms, respectively.

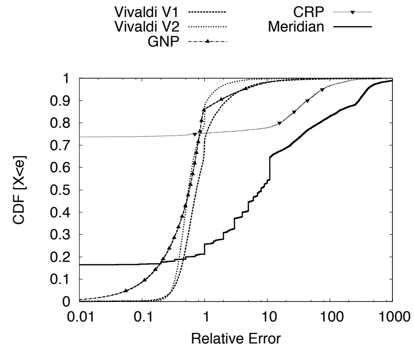

Fig. 4. Absolute value of *relative* errors between estimated and measured latencies. Vivaldi V1 and V2 exhibit median errors that are triple or double previously reported; however, these errors are similar to those in GNP.

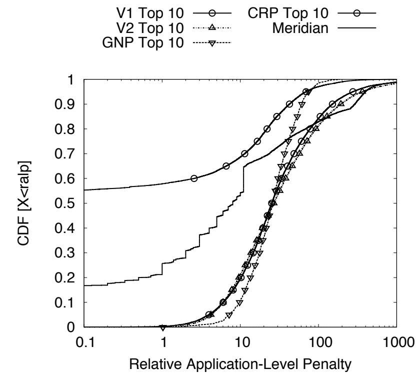

Fig. 5. Relative application-level penalty for using network positioning. The vast majority of values are greater than one and the median values indicate order-of-magnitude loss in performance.

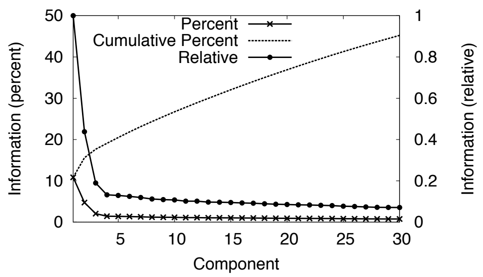

Fig. 6. Plot indicating portion of variance captured by each principal component. The first five components capture only a small portion (20%) of the total variance.

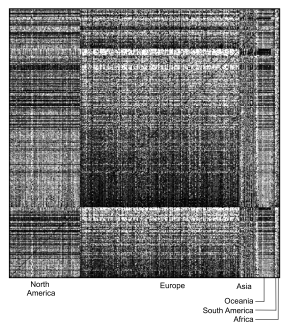

Fig. 7. Map of severity of TIVs in our measured latency space, where rows and columns from the same continent are grouped together. A white point represents the most severe TIV.

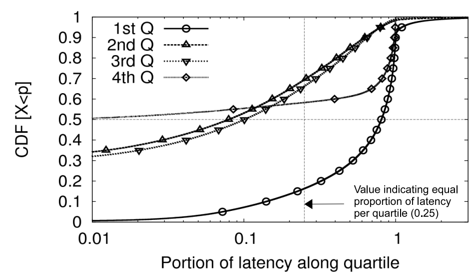

Fig. 8. CDF of portion of end-to-end latency contained in each quartile of the IP-level path between endpoints.

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 [7], [10], [11], this section focuses on variance and intrinsic errors in latency prediction, as they represent fundamental challenges for latency-based approaches to network positioning.

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 combinations of elements sufficient to capture most of its variance (e.g., [20]). 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 [5], [11], [20].

We perform the same PCA analysis on the latency matrix described in Sec. III. Fig. 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.

B. Triangle Inequalities

Triangle inequality violations (TIVs) in a 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. [23] 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 7 visualizes the 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. [23]). 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 [20], 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. [11]. These patterns for TIVs and their severity hints at the challenges in accounting for TIVs in coordinate systems.

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. 8 plots CDFs of the portion of end-to-end latency (log scale) along quartiles of the IP-level path between the measured hosts, using 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 [5], this analysis suggests that a single parameter is insufficient due to the large and variable latencies in a large-scale P2P environment.

References

[1] 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.

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

[3] 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.

[4] D. R. Choffnes, M. A. Sanchez, and F. E. Bustamante, "Network positioning from the edge," Tech. Rep. NWU-EECS-09-19, 2009.

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

[6] 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.

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

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

[9] 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.

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

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

[12] 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.

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

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

[15] PlanetLab, "http://www.planet-lab.org/."

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

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

[18] 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.

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

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

[21] Team Cymru, "http://www.cymru.com/bgp/asnlookup.html."

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

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

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

[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.