On the Stability of Network Distance Estimation

Yan Chen, Khian Hao Lim, and Randy H. Katz
University of California, Berkeley

Chris Overton,
Keynote Systems, Inc.

Abstract

Estimating end-to-end Internet distance can benefit many applications and services, such as efficient overlay construction, overlay routing and location, and peer-to-peer systems. While there have been many proposals on Internet distance estimation, how to design and build a scalable, accurate and dynamic monitoring system is still an open issue. To this end, we propose an overlay distance monitoring system, *Internet Iso-bar*, which yields good estimation accuracy with much less overhead for online monitoring than the existing systems. Internet Iso-bar clusters hosts based on the similarity of their perceived network distance, and chooses the center of each cluster as a monitor site. The distance between two hosts is estimated using inter- or intra-cluster distances. Evaluation using real Internet measurements shows that Internet Iso-bar achieves high estimation accuracy and stability with much smaller measurement overhead than Global Network Positioning (GNP) [1]. Furthermore, by adjusting the number of clusters, we can smoothly trade off the measurement and management cost for better estimation accuracy.

1 Introduction

With the rapid growth of the Internet, there emerges a new class of large-scale globally-distributed network services and applications, such as overlay routing and location systems, application-level multicast, and peer-to-peer file sharing. As these systems have flexibility in choosing their communication paths and targets, they can benefit significantly from dynamic network distance prediction (e.g., latency and bandwidth).

Existing network distance estimation systems can be grouped into two categories: *static estimation* [1, 2] and *dynamic monitoring* [3, 4, 5]. The former schemes, such as Global Network Positioning (GNP) [1], can achieve high accuracy for estimating the normal distance between pair of hosts, but have scalability problems for continuously updating the estimates.

On the other hand, although the congestion outbursts within seconds are hard to detect and bypass, the delay in Internet inter-domain path failovers average over three minutes [6]. Dynamic distance monitoring systems aim to detect these failures timely, and have applications circumvent them. However, existing schemes lack scalability [5], or accuracy [3, 4], or face problems on deployment [3].

In this paper, we focus on designing a scalable distance monitoring system, while achieving good accuracy. In particular, we focus on the following problem: Given N end hosts that may belong to different administrative domains, how to select a subset of them as monitoring sites and build an overlay distance estimation/monitoring service without knowing network topology and with very small measurement overhead. Besides accurate, efficient, and scalable, a network distance monitoring system should also be incrementally deployable, and easy to use.

To this end, we propose a cluster-based network distance estimation scheme: *Internet Iso-bar*, which clusters the end hosts based on the similarity in their distance to a small set of sites. The cluster centers are selected as monitoring sites for active and continuous probing. The distance between any pair of hosts is estimated using inter- or intra-cluster distances.

We compare the accuracy and stability of our scheme with GNP using real Internet measurements from NLANR [7] and Keynote Inc. [8]. We evaluate the estimation stability on various time scales: daily, weekly, monthly, etc. Our results show that Internet Iso-bar has good prediction accuracy and stability, with much less monitoring overhead than GNP. To the best of our knowledge, Internet Iso-bar is one of the first scalable overlay distance monitoring systems, and our work is the first attempt to evaluate the *stability* of various distance estimation schemes using real Internet measurements.

The rest of the paper is organized as follows. We survey the goals and previous work in Section 2. The clustering and distance estimation techniques of Internet Iso-bar are presented in Section 3. We describe our simulation methodology in Section 4, and evaluate the accuracy and stability of different schemes in Section 5. Finally, we conclude in Section 6.

2 Overview and Related Work

2.1 Internet Iso-bar Goals

Scalability There are millions of users for peer-to-peer systems, such as Napster and Gnutella. So we cannot have any centralized point in the system to continuously measure distance to all hosts.

Small overhead and timely information The monitoring system should provide real-time report for network distance between any pair of hosts, which requires small communication and estimation computation cost.

Good distance estimation accuracy and stability While preserving scalability and timely distance estimation, it is desirable to have as accurate estimation as possible. Meanwhile, the system should perform stably as time evolves.

Incrementally deployable Our goal is to install the monitors on end hosts without interfering with the existing core IP network. Furthermore, with more monitors deployed, the estimation accuracy should be incrementally improved.

Ease of use As a globally distributed system, the monitoring service should be straightforward to install, operate and maintain, and requires no specific parameter tuning.

2.2 Alternative Architectures and Related Work

There are numerous works on Internet end-to-end distance estimation/monitoring systems, both in research prototypes and commercially deployed networks.

Content Distribution Network (Akamai's Network Operations Command Center (NOCC)) For each client address prefix (subnets), Akamai uses traceroute from all CDN servers to find a few core routers (close to the clients) that are always on the path to the client clusters. They constantly monitor the distance from each Internet Data Center (IDC), which hosts the CDN edge servers, to these routers to decide the relative distance to the clients [9] [1]. Although working in real operation, this approach has a potential scalability problem. There are more than 8800 Internet Service Providers (ISPs). Suppose every ISP has an IDC for hosting CDN servers and the clients are grouped by autonomous systems (AS's), we need 114M traceroute measurements for building the distance map among 13,000 existing AS's. A similar amount of measurements is needed to maintain the distance maps. Though they divide the maps into regions, and only measure the distance between the IDCs and core routers of AS's in each region [10], the amount of measurements is only reduced by the factor of the number of regions (assume the ISPs and AS's are evenly distributed in the regions). While helpful when the number of regions is big, it will lose agility because the CDN servers in one region cannot serve the clients in other regions.

IDMaps has special HOPS servers (called *Tracers*) that measure the distances among themselves and to other end hosts [3]. The distance between two end hosts A and B is estimated as the sum of 3 distances: distance between A and its nearest Tracer T_1, distance between B and its nearest Tracer T_2 and the shortest path distance between T_1 and T_2. To achieve scalability, they group the clients by Address Prefix (AP) which is a consecutive address range of IP addresses within which all hosts are equidistant (with some tolerance) to the rest of Internet. Grouping like that leads to hundreds of thousands of APs, which are further clustered based on the network proximity. Then, distances between every pair of APs are needed for such a clustering. As a heuristic, Tracers are placed on transit AS's. However, this requires the cooperation of network providers. IDMaps also faces the problem of prediction accuracy. First, the estimation is based on the triangulation inequality which does not hold unless the Tracers are always placed on or very close to the shortest path between the clients. Secondly, the proximity-based clustering is not as accurate as the similarity-based clustering used by Internet Iso-bar, as we will show in Section 5.

Network Distance Maps To tackle the measurement scalability problem, Theilmann and Rothermel have proposed *network distance maps* [4]. Assuming the existence of measurement servers (mServers), a hierarchical tree of mServers can be constructed so that each mServer only measures the distance to its siblings. Each host is then assigned to its closest mServer. The distance between host A and B is estimated by the distance between each of their ancestor mServers. This hierarchical clustering of monitors is complementary to the monitor site selection scheme of Internet Iso-bar. Thus we omit it for comparison in Table 1.

Global Network Positioning (GNP) is based on absolute coordinates computed from modeling the Internet as a D-dimensional geometric space [1]. Every end host maintains its own coordinates, and network distances to other hosts are predicted by evaluating a *distance function* (e.g., Euclidean distance) over their coordinates. Evaluation based on real Internet measurement data shows it is much more accurate than IDMaps. However, the landmark sites are potential bottlenecks because every host has to measure its distance to the landmarks to compute and update its coordinates. Moreover, it is hard to achieve real-time distance estimation – both source and destination have to obtain measurements to $D+1$ landmarks (D is the number of dimensions used in estimation), recompute the coordinates, and exchange the coordinates to get the estimation. Thus it

[1] As a proprietary technique, there is no white paper available regarding this subject, to the best of our knowledge.

Properties	Akamai NOCC	IDMaps	GNP	RON	Internet Iso-bar
Scalability	Traceroute from each edge server farm to all client subnets	Proximity-based clustering of APs, $O(C^2 + AP)$ measurements	NK measurements (each landmark takes $O(N)$ of them)	N^2 measurements	$C^2 + N$ measurements
Estimation technique and accuracy	Use distance between edge server farms to nearby router of clients	Based on triangulation inequality and proximity-based clustering, inaccurate	Based on high-dimension coordinates, accurate	Exact measurements	Similarity-based clustering, accurate
Timely monitoring	Yes	Yes	No	Yes	Yes
Monitors/landmarks deployment	CDN edge servers	Transit AS's	End hosts	End hosts	End hosts

Table 1: Comparison of various Internet distance estimation systems, assuming there are N end hosts, AP address prefixes, K landmarks and C clusters

is not suitable as an online monitoring system. Moreover, GNP can only model symmetric distance, and its optimization algorithms are expensive to run.

Resilient Overlay Network (RON) is an architecture that allows distributed Internet applications to detect and recover from path outages and periods of degraded performance within several seconds [5]. They keep monitoring the quality of Internet paths between *every* pair of RON nodes. Although the scheme achieves high accuracy, it fails to scale.

Table 1 compares Internet Iso-bar with other schemes.

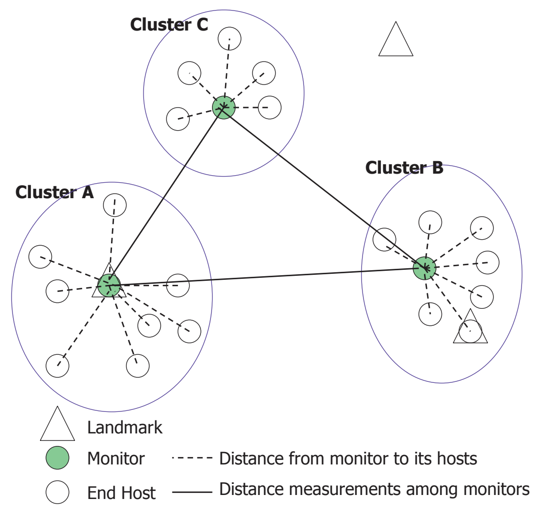

Figure 1: Internet Iso-bar architecture for a peer-to-peer system

3 Internet Iso-bar

The key idea of *Internet Iso-bar* (referred as *Iso-bar* thereafter) is as follows. We group N hosts into K clusters based on several distance metrics (Section 3.1 and Section 3.2), and choose cluster centers as the monitors (Section 3.2). Each monitor continuously measures the distance to all other monitors as well as to the hosts belonging to its cluster. The distance between any pair of hosts is estimated using intra- and inter-cluster distance (Section 3.3). Figure 1 shows the Internet Iso-bar architecture for a peer-to-peer system. Note that the hosts in one cluster may belong to multiple AS's since there are more than 12,000 AS's.

3.1 Distance Metrics

First we define the distance metrics for clustering. We explore two orthogonal metrics: one based on network distance, and the other based on geographical distance.

3.1.1 Clustering based on network distance correlation: There are many ways to define Internet distance. For example, we can use hop count, or latency, or loss rate, or jitter, or bandwidth, etc. In this paper, we define the Internet distance to be the round trip time (RTT). Given asymmetric routing, RTT may not reflect the absolute network distance between two hosts. But in most cases, it can still serve as a good approximation of the end-to-end distance that we aim to estimate. The RTT was measured by ICMP packets in the NLANR traces. The Keynote traces recorded the time for the initial two packets (i.e., the time from sending a SYN packet till receiving a SYN ACK response packet) during TCP three-way handshaking, which is used as the RTT measure.

Using network proximity: Let $net\_dist(i,j)$ denotes the measured network distance between host i and host j, and $cor\_dist(i,j)$ denotes the correlation distance between them. We can directly use $net\_dist(i,j)$ for clustering (i.e., $cor\_dist(i,j) = net\_dist(i,j)$). This is adopted by IDMaps [3] and Network Distance Maps [4]. However, although it requires all pair-wise distance measurements, this scheme is not very accurate, as shown in Section 5. Next, we introduce the network similarity based clustering, which can significantly reduce the amount of measurements.

Using network similarity (Iso-bar): As in GNP, each host i measures distance to landmarks and forms a *network distance vector* (referred as $netV_i$). However unlike GNP, $netV_i$ is used only for clustering and not for distance estimation.

$netV_i$ is a m-dimensional vector where m is the number of landmarks. The landmarks can be all or a subset of the end hosts plus any outside servers that accept measurements.

Let w_k denotes the k-th landmark, and $w_k(netV_i)$ denotes the k-th element of the vector $netV_i$, which stands for the distance between the host i and the k-th landmark. One way is to define the distance between two hosts using the Euclidean distance between their network distance vectors as follows:

$$cor\_dist(i,j) = |netV_i - netV_j|$$

$$= \sqrt{\sum_{k=1}^{m}(w_k(netV_i) - w_k(netV_j))^2}$$

In addition, cosine vector similarity has been widely used to measure the similarity between two vectors. An alternative distance metric is to use the complement of vector similarity.

$$cor\_dist(i,j) = 1 - \frac{netV_i \cdot netV_j}{|netV_i||netV_j|} =$$

$$1 - \frac{\sum_{k=1}^{m} w_k(netV_i) \cdot w_k(netV_j)}{\sqrt{\sum_{k=1}^{m}[w_k(netV_i)]^2 \cdot \sum_{k=1}^{m}[w_k(netV_j)]^2}}$$

3.1.2 Clustering based on geographical distance correlation: Given the longitude and latitude of every host, the correlation distance can be defined as:

$$cor\_dist(i,j) = \sqrt{(long_i - long_j)^2 + (lat_i - lat_j)^2}$$

3.2 Generic Clustering Methods

Given a distance metric, we can apply generic clustering algorithms to group hosts. We define the radius of cluster i as the maximum distance between the monitor (center node) and any host in cluster i. Our clustering seeks to minimize the maximum radius of all clusters, or to minimize the sum of distances between every host and its monitor. The former aims to optimize the worst case prediction, while the latter tries to optimize the average prediction.

3.2.1 Clustering to optimize the worst case: The goal is to place a given number of monitors so that the maximum radius is minimized (denoted as *limit_num_minRmax* clustering). This problem is known as the minimum K-center problem [3]. We use the algorithm in [11] to achieve an approximation within a factor of 2 in $O(N^3)$ time, where N is the number of end hosts.

In addition, we explore the dual problem of limiting the radius of each cluster and minimizing the number of clusters (denoted as *limit_Rmax* clustering). This problem can be converted to the minimum set cover [12] (i.e., Given a graph G with N hosts and a distance bound d, find a smallest subset of N' such that the distance between any host h and its "center" host c_h is bounded by d.). More formally, find the minimum K, such that there is a set $N' \subset N$ with $|N'| = K$ and $\forall\ h \in N$, $\exists\ c_h \in N'$ such that distance(h, c_h) $\leq d$. This is an NP-hard problem. A greedy approximation algorithm is recommended over other methods because it gives comparable results at a fraction of the time ($O(N^2)$ complexity). The approximation ratio is ln$|N|$ [12].

3.2.2 Clustering to optimize the average case: We formulate the clustering problem that minimizes the sum of intra-cluster distance as follows. Given N points, we select K of them to be centers, and assign each point j to the closest center. If point j is assigned to center i, we incur a cost c_{ij}, where c_{ij} is the correlation distance between point i and j. The goal is to select K centers so as to minimize the sum of assignment costs (denoted as *limit_num_minDistSum* clustering). This is an NP-hard minimum K-median problem. We use a 4-approximation algorithm with running time of $O(N^3)$ [13].

3.3 Predicted Distance Computation

For a peer-to-peer system, we predict the distance between any pair of hosts host i and j as follows. If they belong to the same cluster as monitor m, $predicted\_dist(i,j) = (dist(i,m) + dist(j,m))/\ 2$. Otherwise, assuming that the monitor for i is m_i and for j is m_j, then $predicted\_dist(i,j) = dist(m_i, m_j)$.

We need many more inter-cluster estimates than intra-cluster estimates. For example, given N hosts evenly divided into K clusters, and considering all pairs of hosts for distance estimation, the ratio of inter-cluster estimates to intra-cluster estimates will be approximately $K - 1$.

For a client-server system, the distance between any client c and server s is estimated as $predicted\_dist(c,s) = dist(m,s)$, where m is the monitor for c.

We can estimate the distance to a target host using either past measurements or by sending a fresh set of measure-

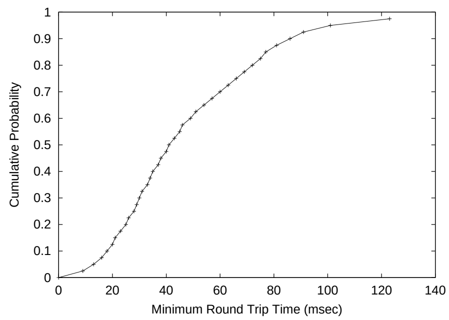

Figure 2: CDF of 06/25/01 daily minimum RTT between each pair of NLANR AMP hosts

ments. Due to the significant delay incurred in the latter approach, we'd like to predict the distance using past information. There are various ways one can use past observations. We consider the following two schemes:

Exponentially Weighted Moving Average (EWMA) $dist_t = \alpha \times measure_{t-1} + (1-\alpha) \times dist_{t-1}$. $dist_t$ and $measure_t$ stand for reported distance and measured distance at time t, respectively. We tried three values for α: 0.1, 0.5 and 0.9.

Geometric mean in a sliding window We also use the geometric mean as prediction, because the Internet distance measurement obey a heavy-tailed distribution (reported by [14] and also confirmed by us). Three window sizes are tested: 5, 15 and 25.

Among all the six combinations, EWMA with α value 0.5 performs the best. So it is adopted for distance prediction in the rest of the paper.

4 Evaluation Methodology

4.1 Internet Measurement Data

We evaluate the performance of different schemes for estimating latency using two sets of real Internet measurement data: one from the National Laboratory of Applied Network Research (NLANR) Active Measurement Project (AMP) data [7], and the other from Keynote Web Site Perspective measurement data [8]. Due to space constraints, we only show a subset of results using NLANR data in this paper; see [15] for more results.

4.1.1 NLANR AMP Data: There are 119 sites participating in the NLANR AMP project, each of which uses a dedicated machine to measure the RTT between itself and the other participating sites by sending ICMP packets. Measurements are made every minute, resulting in a total of 1440 measurements for each day. We collected one month of raw data (6/25/01 to 7/24/01), and one day of more recent data (12/6/01). After filtering out sites with incomplete measurements, we have 106 hosts (i.e., $N = 106$ in our experiments). Figure 2 shows the cumulative distribution function (CDF) of minimum RTT between every pair of hosts.

To have more accurate latency estimation through ping measurement, we use a sliding window of 10 samples, and choose the minium RTT value in the sliding window as the latency. Based on one day's trace, we compute the latency geometric mean of all sliding windows between every pair of sites i and j as $net\_dist(i,j)$, and use them for clustering or coordinate calculation in GNP. We refer to this day as the *birth date*. Then we use the raw measurements of a future day to evaluate the accuracy of estimation results from each technique. The future day is referred as the *estimation date*.

4.1.2 Keynote Web Site Perspective Data: Keynote measurement agents [8] measure Web site performance on a worldwide infrastructure of public measurement computers in 102 statistically selected locations in over 50 cities in the world. While almost all NLANR AMP sites have dedicated T1 connections to the Abilene backbone network [16], the Keynote measurement agents use various connectivity (dial-up, DSL, cable modem, T1, etc.) to link to nearly all major ISPs in the world. The targets, termed as *Keynote Business 40 index*, are mainly the front pages of 40 most popular Web servers used by US businesses. The 102 agents measure the TCP connection time to each of the 40 web sites every 15 minutes, i.e., a client-server type of system. We collect one month of raw data from 11/13/2001 to 12/13/2001, and filter out the occasional error measurements due to such causes as DNS lookup and TCP timeouts.

4.2 Estimation Accuracy Metric

We use the following relative prediction error defined in [17] as a measure for accuracy:

$$relative\ error = Avg[|log_2(\frac{predicted\ distance}{measured\ distance})|]$$

where Avg refers to the average value computed over each of the measurement events (i.e., 1440 in the NLANR daily traces). For NLANR, both the predicted distance and measured distance are based on the minimum RTT of 10-minute sliding window. The relative error reflects how much the estimation deviates from the target value.

Given different estimation techniques, the predicted distance could be either *static* or *dynamic* (see Section 4.3).

GNP is not scalable enough for online update of the coordinates. Therefore we use each site's coordinates obtained from the birth date for distance estimation. For any pair of hosts i and j, the same estimated value is used consistently as predicted distance when we calculate the daily relative error.

For Iso-bar, we assume the monitors actively measuring the distance among themselves, as well as to the other hosts in its cluster. Then those measurements are used to predict the distance between clients as in Section 3.3.

4.3 Analysis of Estimation Accuracy

4.3.1 Static analysis: For static analysis, we use the measurement data of the same day both for offline setup (clustering or coordinates computation) and for online estimation. That is, the birth date and the estimation date are the same. This will help give a sense of the absolute accuracy of each estimation technique without temporal variation.

4.3.2 Stability analysis: It is even more interesting to examine how well estimations derived from a single day's data perform over multiple time intervals. We evaluate the stability of various distance estimation schemes over a six-month period. The birth date is 06/25/01, and the estimation dates are 6/25/01, 6/26/01, 6/28/01, 7/01/01, 7/8/01, 7/15/01, 7/24/01, and 12/06/01, respectively.

5 Evaluation Results

In this section, we first describe different Internet distance estimation techniques to be used for comparison, and then we compare the generic clustering techniques to simplify the evaluation of distance estimation systems. Next, we examine the sensitivity of Iso-bar with varying number of landmarks. Finally, we present the accuracy and stability of various estimation schemes.

5.1 Internet Distance Estimation Techniques Evaluated

We compare five distance estimation schemes:

- Omniscient approach
- GNP
- Clustering with network distance vector (Iso-bar)
  - Using Euclidean distance (*Net_sim*)
  - Using vector similarity (*Net_vsim*)
- Clustering with geographical proximity
- Clustering with network proximity

The omniscient approach makes distance estimation based on complete knowledge about every pair-wise distance on the birth date. It estimates the distance between any pair of hosts as the geometric-mean of the actual measured distance on the birth date. For a fair comparison, we use 15 landmarks (in 7 dimensions as configured in [1]) for GNP and 15 clusters for clustering approaches.

5.2 Evaluation of Generic Clustering Techniques

First, we compare the two clustering methods for optimizing the worst case: *Limit_number_minRmax* clustering and *limit_Rmax* clustering. We save the comparison of *limit_number_minDistSum* clustering to Section 5.4. Figure 3 shows the accuracy and stability using the network similarity based clustering (*Net_sim* and *Net_vsim*). *Limit_number_minRmax* clustering performs slightly better than *limit_Rmax* clustering. We observe similar results when using proximity-based correlation distance for clustering. So unless specifically denoted, we only use *limit_number_minRmax* clustering thereafter.

When comparing *Net_sim* and *Net_vsim*, clustering based on *Net_sim* slightly outperforms *Net_vsim*, thus only *Net_sim* is applied for the stability analysis.

5.3 Iso-bar Sensitivity to Different Number of Landmarks

In this section, we try to answer the following question: Can we achieve good clustering performance based on measurements to only a small random subset of landmarks? Our approach is to compare the performance of Iso-bar (*limit_num_minDistSum*) when the number of randomly-chosen landmarks M varies from 6 to 106. Figure 4 shows that they do not differ much in terms of prediction accuracy. This suggests that we only need coarse level distance information, and a small number of landmarks is sufficient for clustering. For the remaining experiments, we use 106 landmarks and expect similar results for the other numbers of landmarks.

5.4 Stability Results of Prediction Accuracy

Figure 5 shows the CDF of the relative prediction errors for both static and stability analysis. The stability results for daily, weekly and monthly interval are very close to the static one.

Figure 6 shows the amount of relative errors that are below the 80th and 90th percentiles in the CDF of relative errors, and how the relative errors change over time. In this metric, a lower value represents a high level of accuracy. Most methods are relatively stable except omniscient scheme.

We make the following observations. First, estimation based on the omniscient approach gives the highest accuracy on all the estimation dates. We believe this represents an underlying stability in the current Internet. Also note that as in any static estimation scheme, the omniscient approach can not report any transient network congestion. Furthermore, it requires the full $N \times N$ network distance matrix, and thus not scalable. It is worth pointing out that the omniscient approach does not achieve perfect accuracy when it is used to estimate distance on the actual birth date. This is because the distance estimate between any pair of hosts is the geo-

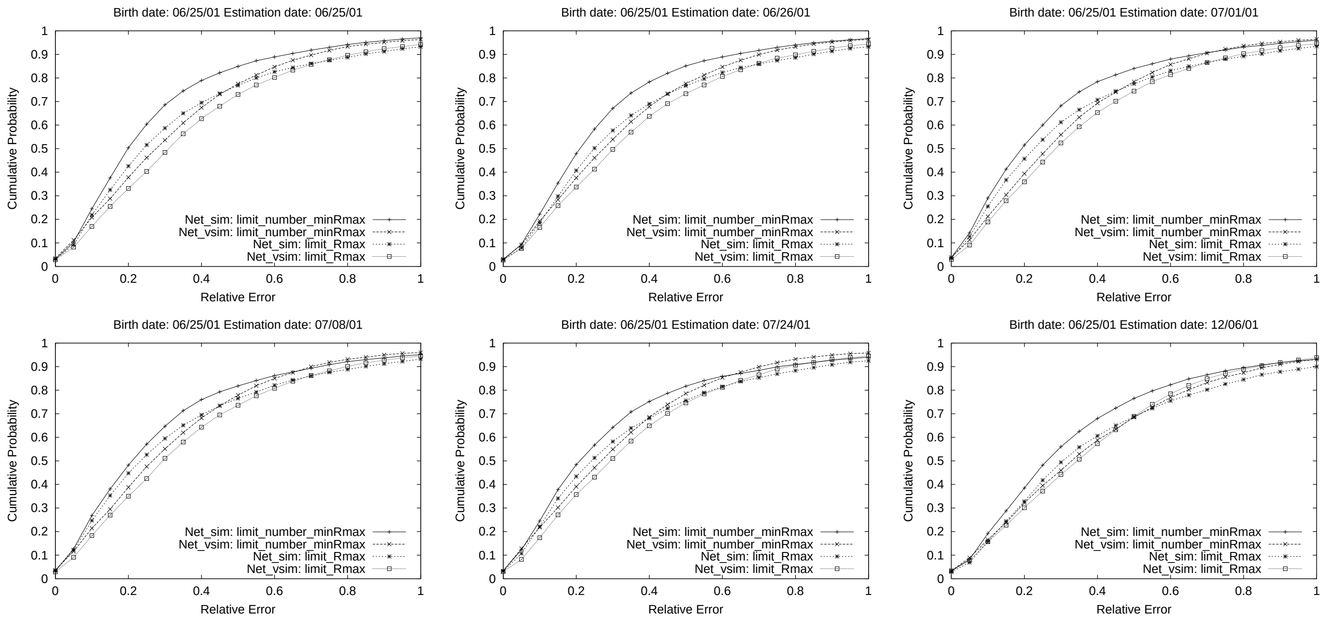

Figure 3: Evaluation of generic clustering methods with static analysis (top left) and stability analysis of five different time intervals with NLANR data

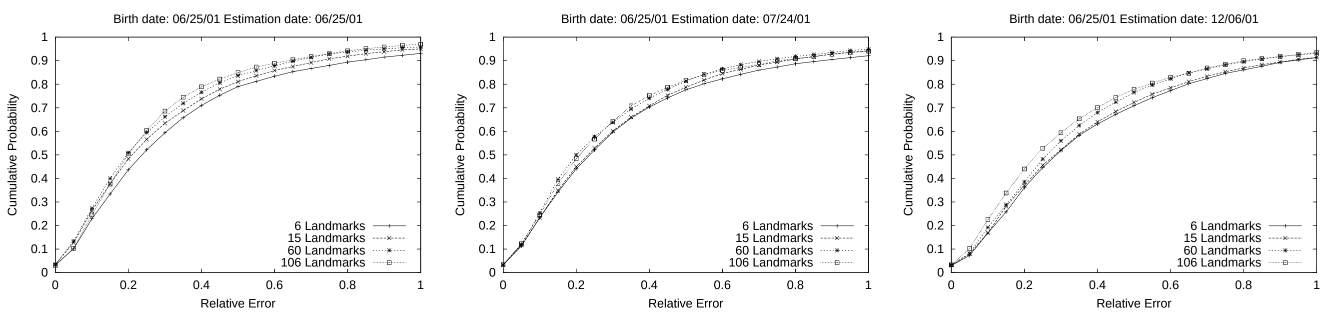

Figure 4: Sensitivity of Internet Iso-bar to various number of landmarks: static analysis (left) and stability analysis (middle, right) with NLANR data

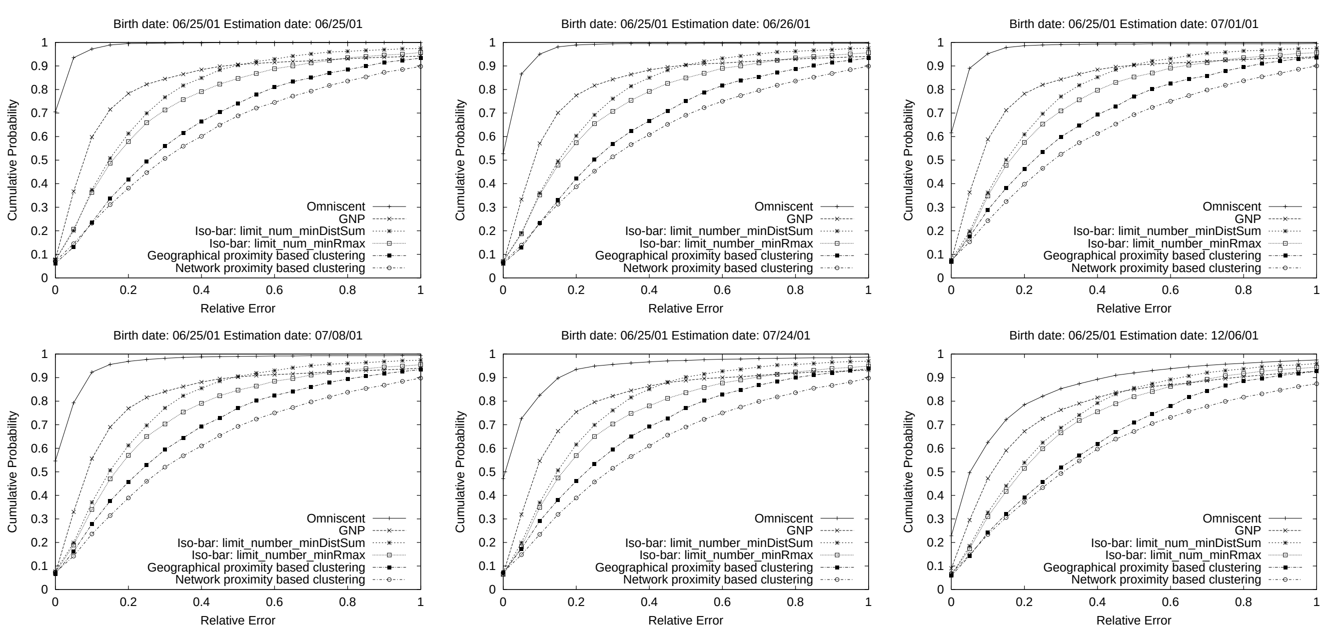

Figure 5: Cumulative Distribution Function (CDF) of relative prediction errors for both static and stability analysis (five different time intervals) by applying various distance estimation services on NLANR data

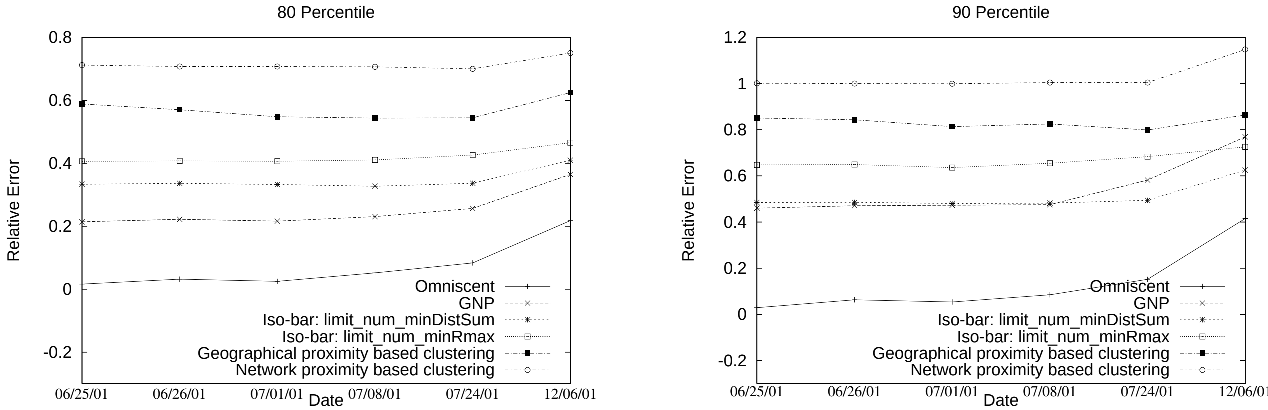

Figure 6: 80 percentile (left) and 90 percentile (right) of the relative errors for various distance estimation services under six different time intervals, evaluated with NLANR data

metric mean of minimum RTT in 10-minute sliding window samples obtained during the whole day. The error reflects the amount of fluctuation in RTT within a day.

Second, GNP and Iso-bar (*limit_num_minDistSum*) have similar performance, second only to the omniscient scheme. GNP performs a little better for the 80th percentile. This implies that GNP provides a very accurate estimate when the underlying data set is stable. However, the accuracy difference between GNP and Iso-bar is almost negligible. Take the 80th percentile results of 06/25/01 for instance, the ratio between predicted distance and real distance for GNP is $2^{0.21} = 1.16$, while for Iso-bar is $2^{0.33} = 1.26$. Given that 90% geometric mean of pair-wise RTT is less than 91 ms for NLANR (06/25/01), the prediction difference between GNP and Iso-bar is less than 10 ms. In addition, GNP requires all hosts to constantly measure distance to the landmarks for online monitoring, which is impractical. Thus it cannot report timely congestion/loss, but that information is more interesting to many clients.

Third, the network distance similarity based clustering yields the highest accuracy among all the clustering based approaches. It performs much better than the network proximity based clustering, which is used in IDMaps [3] and Network Distance Maps [4].

Finally, it is interesting to see that geographical distance proximity based clustering performs better than network distance proximity based clustering. This may be because most sites in our experimental data set are educational and research institutes which are connected by the same backbone: Internet2. The correlation between geographical distance and network distance need to be further verified using the Keynote data. Previous work has indicated that estimation techniques based on geographic distance in general results in poor accuracy [1].

6 Conclusion

In this project we propose the framework for a novel clustering-based overlay monitoring service, the *Internet Iso-bar*. It clusters hosts based on the similarity of their perceived distance to a small number of landmarks. Compared with the state-of-the-art work, such as GNP, Internet Iso-bar has much better scalability and smaller measurement overhead for online monitoring without much loss in estimation accuracy and stability.

As future work, we plan to explore other clustering methods to improve the estimation accuracy and have more comprehensive comparison between NLANR and Keynote data. In addition, we want to design and test the dynamic service model for Internet Iso-bar, i.e., how to adapt to the dynamic joining and leaving of the clients. One simple way could have the joining client to ping the landmark sites to get its network distance vector. Then it will collect the distance vectors of the monitoring sites, and choose the one with best correlation to join. We may also need to dynamically set up or turn off monitors, i.e., split or merge of clusters. Moreover, we are going to consider other performance metrics besides latency, such as congestion or failure detection, and study the fault tolerance of Internet Iso-bar.

7 Acknowledgments

We graciously acknowledge sponsorship and grants from DARPA (grant N66061-99-2-8913), California Micro Grant #01-042, Ericsson, Nokia, Siemens, Sprint, NTTDoCoMo and HRL laboratories. We thank Chris Karlof and Yaping Li for discussion and help with implementation. We thank Vern Paxon and the anonymous reviewers for their valuable suggestions. Tony McGregor kindly provides us the NLANR AMP data to make the study possible.

References

[1] T. S. E. Ng and H. Zhang, "Predicting Internet network distance with coordinates-based approaches," in *Proc.of Infocom*, 2002.

[2] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker, "Topologically-aware overlay construction and server selection," in *Proc. of IEEE Infocom*, 2002.

[3] 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 Trans. on Networking*, Oct. 2001.

[4] W. Theilmann and K. Rothermel, "Dynamic distance maps of Internet," in *Proceedings of IEEE Infocom*, 2000.

[5] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris, "Resilient overlay networks," in *Proc. of ACM SOSP*, 2001.

[6] C. Labovitz, A. Ahuja, A. Abose, and F. Jahanian, "An experimental study of delayed Internet routing convergence," in *Proc. of ACM SIGCOMM*, 2000.

[7] NLANR, "http://amp.nlanr.net/," .

[8] Keynote Inc., "Assure the quality of your Web site performance," `http://www.keynote.com/solutions/html/web_site_perspective.html`.

[9] P. Danzig, ," Former V. P. Techonology of Akamai, Personal communication.

[10] T. Leighton, "The challenges of delivering content and applications on the Internet," Talk at UC Berkeley, Oct. 2002.

[11] V. Varirani, *Approximation Methods*, Springer-Verlag, 1999.

[12] T. Grossman and A. Wool, "Computational experience with approximation algorithms for the set covering problem," *Euro. J. Operational Research*, vol. 101, no. 1, pp. 81–92, August 1997.

[13] M. Charikar and S. Guha, "Improved combinatorial algorithms for the facility location and k-median problems," in *Proceedings of FOCS*, 1999.

[14] A. Medina, I. Matta, and J. Byers, "On the origin of power laws in Internet topologies," in *ACM Computer Comm. Review*, Apr. 2000.

[15] Y. Chen, K. Lim, C. Overton, and R. H. Katz, "On the stability of network distance estimation," in *UCB/CSD Tech Report No. CSD-02-1182*, 2002.

[16] Abilene Network Backbone, "`http://www.ucaid.edu/abilene/home.html`," .

[17] Y. Zhang, N Duffield, V. Paxson, and S. Shenker, "On the constancy of Internet path properties," in *Proc. of SIGCOMM Internet Measurement Workshop 2001.*