Predicting Internet Network Distance with Coordinates-Based Approaches

T. S. Eugene Ng and Hui Zhang
Carnegie Mellon University
Pittsburgh, PA 15213
{eugeneng, hzhang}@cs.cmu.edu

Abstract— **In this paper, we propose to use coordinates-based mechanisms in a peer-to-peer architecture to predict Internet network distance (i.e. round-trip propagation and transmission delay). We study two mechanisms. The first is a previously proposed scheme, called the triangulated heuristic, which is based on relative coordinates that are simply the distances from a host to some special network nodes. We propose the second mechanism, called Global Network Positioning (GNP), which is based on absolute coordinates computed from modeling the Internet as a geometric space. Since end hosts maintain their own coordinates, these approaches allow end hosts to compute their inter-host distances as soon as they discover each other. Moreover coordinates are very efficient in summarizing inter-host distances, making these approaches very scalable. By performing experiments using measured Internet distance data, we show that both coordinates-based schemes are more accurate than the existing state of the art system IDMaps, and the GNP approach achieves the highest accuracy and robustness among them.**

I. Introduction

As innovative ways are being developed to harvest the enormous potential of the Internet infrastructure, a new class of large-scale globally-distributed network services and applications such as distributed content hosting services, overlay network multicast [1][2], content addressable overlay networks [3][4], and peer-to-peer file sharing such as Napster and Gnutella have emerged. Because these systems have a lot of flexibility in choosing their communication paths, they can greatly benefit from intelligent path selection based on network performance. For example, in a peer-to-peer file sharing application, a client ideally wants to know the available bandwidth between itself and all the peers that have the wanted file. Unfortunately, although dynamic network performance characteristics such as available bandwidth and latency are the most relevant to applications and can be accurately measured on-demand, the huge number of wide-area-spanning end-to-end paths that need to be considered in these distributed systems makes performing on-demand network measurements impractical because it is too costly and time-consuming.

To bridge the gap between the contradicting goals of performance optimization and scalability, we believe a promising approach is to attempt to predict the network distance (i.e., round-trip propagation and transmission delay, a relatively stable characteristic) between hosts, and use this as a first-order discriminating metric to greatly reduce or eliminate the need for on-demand network measurements. Therefore, the critical problem is to devise techniques that can predict network distance accurately, scalably, and in a timely fashion.

In the pioneering work of Francis et al [5], the authors examined the network distance prediction problem in detail from a topological point of view and proposed the first complete solution called IDMaps. IDMaps is an infrastructural service in which special HOPS servers maintain a virtual topology map of the Internet consisting of end hosts and special hosts called Tracers. The distance between hosts $\mathcal{A}$ and $\mathcal{B}$ is estimated as the distance between $\mathcal{A}$ and its nearest Tracer $\mathcal{T}_1$, plus the distance between $\mathcal{B}$ and its nearest Tracer $\mathcal{T}_2$, plus the shortest path distance from $\mathcal{T}_1$ to $\mathcal{T}_2$ over the Tracer virtual topology. As the number of Tracers grow, the prediction accuracy of IDMaps tends to improve. Designed as a client-server architecture solution, end hosts can query HOPS servers to obtain network distance predictions. An experimental IDMaps system has been deployed.

In this paper, we explore an alternative architecture for network distance prediction that is based on peer-to-peer. Compared with client-server based solutions, peer-to-peer systems have potential advantages in scaling. Since there is no need for shared servers, potential performance bottlenecks are eliminated, especially when the system size scales up. Performance may also improve as there is no need to endure the latency of communicating with remote servers. In addition, this architecture is consistent with emerging peer-to-peer applications such as media files sharing, content addressable overlay networks [3][4], and overlay network multicast [1][2] which can greatly benefit from network distance information.

Specifically, we propose coordinates-based approaches for network distance prediction in the peer-to-peer architecture. The main idea is to ask end hosts to maintain *coordinates* (i.e. a set of numbers) that characterize their locations in the Internet such that network distances can be predicted by evaluating a *distance function* over hosts' coordinates. Coordinates-based approaches fit well with the peer-to-peer architecture because when an end host discovers the identities of other end hosts in a peer-to-peer application, their pre-computed coordinates can be piggybacked, thus network distances can essentially be com-

This research was sponsored by DARPA under contract number F30602-99-1-0518, and by NSF under grant numbers Career Award NCR-9624979, ANI-9730105, ITR Award ANI-0085920, and ANI-9814929. Additional support was provided by Intel. Views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of DARPA, NSF, Intel, or the U.S. government.

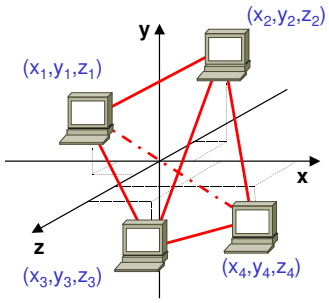

Fig. 1. Geometric space model of the Internet

puted instantaneously by the end host.[1]

Another benefit of coordinates-based approaches is that coordinates are highly efficient in summarizing a large amount of distance information. For example, in a multi-party application, the distances of all paths between K hosts can be efficiently communicated by K sets of coordinates of D numbers each (i.e. $O(K \cdot D)$ of data), as opposed to $K(K-1)/2$ individual distances (i.e., $O(K^2)$ of data). Thus, this approach is able to trade local computations for significantly reduced communication overhead, achieving higher scalability.

We study two types of coordinates for distance prediction. The first is a kind of *relative* coordinates, originally proposed by Hotz [6] to construct the *triangulated heuristic*. Hotz's goal was to apply this heuristic in the A^* heuristic search algorithm to reduce the computation overhead of shortest-path searches in interdomain graphs. The potential of this heuristic for network distance prediction has not been previously studied. The second is a kind of *absolute* coordinates obtained using a new approach we propose called Global Network Positioning (GNP). As illustrated in Figure 1, the key idea of GNP is to model the Internet as a geometric space (e.g. a 3-dimensional Euclidean space) and characterize the position of any host in the Internet by a point in this space. The network distance between any two hosts is then predicted by the modelled geometric distance between them.

As we will show in Section VI, the two coordinates-based approaches are both more accurate than the virtual topology map model used in IDMaps. Furthermore, GNP is the most accurate and robust of all three approaches. Because GNP is very general, it leads to many research issues. In this study, we will focus on characterizing its performance and provide insights on what geometric space should be used to model the Internet, and how to fine tune it to achieve the highest prediction accuracy.

The rest of this paper is organized as follows. In the next section, we explain the triangulated heuristic and discuss its use in a peer-to-peer architecture for Internet distance prediction. In Section III, we describe the GNP approach and its peer-to-peer realization in the Internet. In Section IV, we compare the properties of GNP, the triangulated heuristic, and IDMaps. In Section V, we describe the methodology we use to evaluate the accuracy of network distance prediction mechanisms and in Section VI, we present experimental results based on Internet measurements to compare the performance of the triangulated heuristic, GNP and IDMaps. Finally, we summarize in Section VII.

[1] Note that while we focus on the peer-to-peer architecture for coordinates-based approaches in this paper, nothing prevents coordinates-based approaches to be used in a client-server architecture when it is deemed more appropriate.

II. Triangulated Heuristic

The triangulated heuristic is a very interesting way to bound network distance assuming shortest path routing is enforced. The key idea is to select N nodes in a network to be *base nodes* $\mathcal{B}_i$. Then, a node $\mathcal{H}$ is assigned coordinates which are simply given by the N-tuple of distances between $\mathcal{H}$ and the N base nodes, i.e. $(d_{\mathcal{HB}_1}, d_{\mathcal{HB}_2}, .., d_{\mathcal{HB}_N})$. Hotz's coordinates are therefore *relative* to the set of base nodes. Given two nodes $\mathcal{H}_1$ and $\mathcal{H}_2$, assuming the triangular inequality holds, the triangulated heuristic states that the distance between $\mathcal{H}_1$ and $\mathcal{H}_2$ is bounded below by $L = \max_{i\in\{1,2,..,N\}}(|d_{\mathcal{H}_1\mathcal{B}_i} - d_{\mathcal{H}_2\mathcal{B}_i}|)$ and bounded above by $U = \min_{i\in\{1,2,..,N\}}(d_{\mathcal{H}_1\mathcal{B}_i} + d_{\mathcal{H}_2\mathcal{B}_i})$. Various weighted averages of L and U can then be used as distance functions to estimate the distance between $\mathcal{H}_1$ and $\mathcal{H}_2$.

Hotz's simulation study focused on tuning this heuristic to explore the trade-off between path optimality and computation overhead in A^* heuristic shortest path search problems and did not consider the prediction accuracy of the heuristic. L was suggested as the preferred metric to use in A^* because it is *admissible* and therefore optimality and completeness are guaranteed. In a later study, Guyton and Schwartz [7] applied $(L+U)/2$ as the distance estimate in their simulation study of the nearest server selection problem with only limited success. In this paper, we apply this heuristic to the Internet distance prediction problem and conduct a detailed study using measured Internet distance data to evaluate its effectiveness. We discover that the upper bound heuristic U actually achieves very good accuracy and performs far better than the lower bound heuristic L or the $(L+U)/2$ metric in the Internet.

To use the triangulated heuristic for network distance prediction in the Internet, we propose the following simple peer-to-peer architecture. First, a small number of distributed base nodes are deployed over the Internet. The only requirement of these base nodes is that they must reply to in-coming ICMP ping messages. Each end host that wants to participate measures the round-trip times between itself and the base nodes using ICMP ping messages and takes the minimum of several measurements as the distances. These distances are used as the end host's coordinates. When end hosts discover each other, they piggyback their coordinates and subsequently host-to-host distances can be predicted by the triangulated heuristic without performing any on-demand measurement.

III. Global Network Positioning

To enable the scalable computation of geometric host coordinates in the Internet, we propose a two-part architecture. In the first part, a small distributed set of hosts called Landmarks first compute their own coordinates in a chosen geometric space. The Landmarks' coordinates serve as a frame of reference and are disseminated to any host who wants to participate. In the second part, equipped with the Landmarks' coordinates, any end host can compute its own coordinates relative to those of the Landmarks. In the following sections, we describe this two-part architecture in detail. The properties of this architecture is summarized and compared to those of IDMaps and the triangulated heuristic in Section IV.

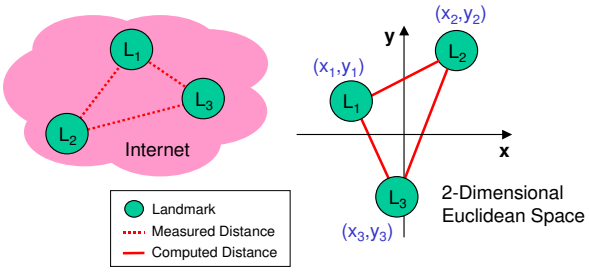

Fig. 2. Part 1: Landmark operations

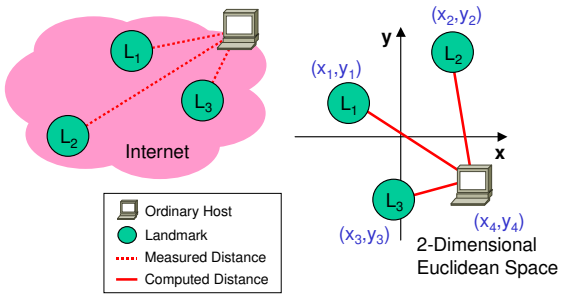

Fig. 3. Part 2: Ordinary host operations

A. Part 1: Landmark Operations

Suppose we want to model the Internet as a particular geometric space $\mathcal{S}$. Let us denote the coordinates of a host $\mathcal{H}$ in $\mathcal{S}$ as $c^{\mathcal{S}}_{\mathcal{H}}$, the distance function that operates on these coordinates as $f^{\mathcal{S}}(\cdot)$, and the computed distance between hosts $\mathcal{H}_1$ and $\mathcal{H}_2$, i.e. $f^{\mathcal{S}}(c^{\mathcal{S}}_{\mathcal{H}_1}, c^{\mathcal{S}}_{\mathcal{H}_2})$, as $\hat{d}^{\mathcal{S}}_{\mathcal{H}_1\mathcal{H}_2}$.

The first part of our architecture is to use a small distributed set of hosts known as Landmarks to provide a set of reference coordinates necessary to orient other hosts in $\mathcal{S}$. How to optimally choose the locations and the number of Landmarks remains an open question, although we will provide some insights in Section VI. However, note that for a geometric space of dimensionality D, we must use at least $D+1$ Landmarks because otherwise, as it will become clear in the next section, it is impossible to uniquely compute host coordinates.

Suppose there are N Landmarks, $\mathcal{L}_1$ to $\mathcal{L}_N$. The Landmarks simply measure the inter-Landmark round-trip times using ICMP ping messages and take the minimum of several measurements for each path to produce the bottom half of the $N \times N$ distance matrix (the matrix is assumed to be symmetric along the diagonal). We denote the measured distance between host $\mathcal{H}_1$ and $\mathcal{H}_2$ as $d_{\mathcal{H}_1\mathcal{H}_2}$. Using the measured distances, $d_{\mathcal{L}_i\mathcal{L}_j}, i > j$, a host, perhaps one of the N Landmarks, computes the coordinates of the Landmarks in $\mathcal{S}$. The goal is to find a set of coordinates, $c^{\mathcal{S}}_{\mathcal{L}_1}, .., c^{\mathcal{S}}_{\mathcal{L}_N}$, for the N Landmarks such that the overall error between the measured distances and the computed distances in $\mathcal{S}$ is minimized. Formally, we seek to minimize the following objective function $f_{obj1}(\cdot)$:

$$f_{obj1}(c^{\mathcal{S}}_{\mathcal{L}_1}, .., c^{\mathcal{S}}_{\mathcal{L}_N}) = \sum_{\mathcal{L}_i, \mathcal{L}_j \in \{\mathcal{L}_1, .., \mathcal{L}_N\} \mid i>j} \mathcal{E}(d_{\mathcal{L}_i\mathcal{L}_j}, \hat{d}^{\mathcal{S}}_{\mathcal{L}_i\mathcal{L}_j}) \tag{1}$$

where $\mathcal{E}(\cdot)$ is an error measurement function, which can be the simple squared error

$$\mathcal{E}(d_{\mathcal{H}_1\mathcal{H}_2}, \hat{d}^{\mathcal{S}}_{\mathcal{H}_1\mathcal{H}_2}) = (d_{\mathcal{H}_1\mathcal{H}_2} - \hat{d}^{\mathcal{S}}_{\mathcal{H}_1\mathcal{H}_2})^2 \tag{2}$$

or some other more sophisticated error measures. To be expected, the way error is measured in the objective function will critically affect the eventual distance prediction accuracy. In Section VI, we will compare the performance of several straight-forward error measurement functions. With this formulation, the computation of the coordinates can be cast as a generic multi-dimensional global minimization problem that can be approximately solved by many available methods such as the Simplex Downhill method [8], which we use in this paper. Figure 2 illustrates these Landmark operations for 3 Landmarks in the 2-dimensional Euclidean space. Note that there are infinitely many solutions for the Landmarks' coordinates because any rotation and/or additive translation of a set of solution coordinates will preserve the inter-Landmark distances. But since the Landmarks' coordinates are only used as a frame of reference in GNP, only their relative locations are important, hence any solution will suffice. When a re-computation of Landmarks' coordinates is needed over time, we can ensure the coordinates are not drastically changed if we simply input the old coordinates instead of random numbers as the start state of the minimization problem.

Once the Landmarks' coordinates, $c^{\mathcal{S}}_{\mathcal{L}_1}, .., c^{\mathcal{S}}_{\mathcal{L}_N}$, are computed, they are disseminated, along with the identifier for the geometric space $\mathcal{S}$ used and (perhaps implicitly) the corresponding distance function $f^{\mathcal{S}}(\cdot)$, to any ordinary host that wants to participate in GNP. In this discussion, we leave the dissemination mechanism (e.g. unicast vs. multicast, push vs. pull, etc) and protocol unspecified.

B. Part 2: Ordinary Host Operations

In the second part of our architecture, ordinary hosts are required to actively participate. Using the coordinates of the Landmarks in the geometric space $\mathcal{S}$, each ordinary host now derives its own coordinates. To do so, an ordinary host $\mathcal{H}$ measures its round-trip times to the N Landmarks using ICMP ping messages and takes the minimum of several measurements for each path as the distance. In this phase, the Landmarks are completely passive and simply reply to incoming ICMP ping messages. Using the N measured host-to-Landmark distances, $d_{\mathcal{H}\mathcal{L}_i}$, host $\mathcal{H}$ can compute its own coordinates $c^{\mathcal{S}}_{\mathcal{H}}$ that minimize the overall error between the measured and the computed host-to-Landmark distances. Formally, we seek to minimize the following objective function $f_{obj2}(\cdot)$:

$$f_{obj2}(c^{\mathcal{S}}_{\mathcal{H}}) = \sum_{\mathcal{L}_i \in \{\mathcal{L}_1, .., \mathcal{L}_N\}} \mathcal{E}(d_{\mathcal{H}\mathcal{L}_i}, \hat{d}^{\mathcal{S}}_{\mathcal{H}\mathcal{L}_i}) \tag{3}$$

where $\mathcal{E}(\cdot)$ is again an error measurement function as discussed in the previous section. Like deriving the Landmarks' coordinates, this computation can also be cast as a generic multi-dimensional global minimization problem. Figure 3 illustrates these operations for an ordinary host in the 2-dimensional Euclidean space with 3 Landmarks.

It should now become clear why the number of Landmarks N must be greater than the dimensionality D of the geometric space $\mathcal{S}$. If N is not greater than D, the Landmarks' coordinates are guaranteed to lie on a hyperplane of at most $D-1$ dimensions. Consequently, a point in the D-dimensional space and its reflection across the Landmarks' hyperplane cannot be distinguished by the objective function, leading to ambiguous host

		IDMaps	Triangulated heuristic	GNP
# Paths measured		$O(N^2 + N*AP)$	$O(N*H)$	$O(N^2 + N*H)$
Communication cost	Off-line	$O(N^2 + N*AP)$ data sent to S HOPS servers	None	$O(N^2)$ data sent to one Landmark; $O(N*D)$ data sent to H hosts
	On-line (K hosts)	$O(K^2)$	$O(K*N)$	$O(K*D)$
	Server latency	Yes	No	No
Computation cost	Off-line	$O(AP*N*\log N) + O(N^3)$ at S HOPS servers	None	$O(N^{2*}D)$ per $f_{obj1}()$ at one Landmark $O(N*D)$ per $f_{obj2}()$ at H hosts
	On-line	$O(1)$ with $O(N^2 + AP)$ storage at S HOPS servers	$O(N)$	$O(D)$
Deployment	End hosts	Implement query/reply protocol	Perform measurements, exchange coordinates, and compute distances	Retrieve Landmarks' coordinates, perform measurements, compute own coordinates, exchange coordinates, and compute distances
	Infrastructure	Tracers measure all paths, send results to HOPS servers; HOPS servers implement query/reply protocol, compute distances	Base nodes reply to pings	Landmarks measure inter-Landmark paths, compute own coordinates and send them to end hosts; reply to pings
	Firewall compatibility	No	Yes	Yes

Fig. 4. Properties of distance prediction schemes

coordinates. Note that in general there is no guarantee that the host coordinates will be unique. Using fewer dimensions than the number of Landmarks is simply to avoid obvious problems.

IV. IDMaps, Triangulated Heuristic and GNP Comparison

In this section, we discuss the differences between IDMaps, the triangulated heuristic, and GNP and illustrate the benefits of each approach and the trade-offs. First, let us briefly describe IDMaps' architecture. IDMaps is an infrastructural service in which hosts called Tracers are deployed to measure the distances between themselves, possibly not the full mesh to reduce cost, and each Tracer is responsible for measuring the distances between itself and the set of IP addresses or IP address prefixes in the world that are closest to it. These raw distance measurements are broadcasted over IP multicast to hosts call HOPS servers which use the raw distances to build a virtual topology consisting of Tracers and end hosts to model the Internet. HOPS servers perform distance prediction computations and interact with client hosts via a query/reply protocol.

Common to all three approaches is the need for some infrastructure nodes (i-nodes), i.e. the Tracers of IDMaps, the base nodes of the triangulated heuristic, or the Landmarks of GNP. Thus, a key parameter of these architectures is the number of these i-nodes, N. In addition to N Tracers, the IDMaps architecture is further characterized by the number of HOPS servers, S, and the number of address prefixes, AP, for Tracers to probe. For GNP and the triangulated heuristic, in addition to N base nodes or Landmarks, they are characterized by the number of end hosts, H, that need distance predictions. GNP is further characterized by the dimensionality, D, of the geometric space used in computing host coordinates. Figure 4 summarizes the differences between the three schemes in terms of measurement cost, communication cost, computation cost, and deployment. To clarify, the off-line computation cost of IDMaps is $O(AP \cdot N \cdot \log N) + O(N^3)$ because the AP address prefixes need to be associated with their nearest Tracers and the all-pair shortest path distances between the N Tracers need to be computed. For GNP, in computing Landmarks' coordinates, each evaluation of $f_{obj1}(\cdot)$ takes $O(N^2 \cdot D)$ time. In computing end host coordinates, each evaluation of $f_{obj2}(\cdot)$ takes $O(N \cdot D)$ time. In our experiments, on a 866 MHz Pentium III, computing all 15 Landmarks' coordinates takes on the order of a second, and computing an ordinary host's coordinates takes on the order of ten milliseconds.

Since the measurement overhead and the off-line costs of all three schemes are acceptable, what differentiate them are their on-line scalability, their prediction accuracy (which we shall discuss in Section VI) and other qualitative differences. The main difference between the distance prediction techniques is scaling. The coordinates-based approaches have higher scalability because the communication cost of exchanging coordinates to convey distance information among a group of K hosts grows linearly with K as opposed to quadratically. In addition, the peer-to-peer architecture also helps to achieve higher scalability because on-line computations of network distances are not performed by shared servers. Since end hosts coordinates can be piggybacked when end hosts discover each other, distance predictions in the peer-to-peer architecture are essentially instantaneous and will not be subjected to the additional communication latency required to contact a server or delays due to server overload. Finally, the peer-to-peer architecture is easier to deploy because the i-nodes are passive and therefore do not require detailed knowledge of the Internet in order to choose IP addresses to probe. An added benefit is that end hosts behind firewalls can still participate in the peer-to-peer architecture.

The peer-to-peer architecture however does have several disadvantages. First, there is nothing to prevent an end host from lying about its coordinates in order to avoid being selected by other end hosts. Thus, this architecture may not be suitable in an uncooperative environment. In contrast, in the client-server architecture, an i-node can verify an end host's ping response time against the response time of its neighbors. Another potential issue is that because the i-nodes in the peer-to-peer architecture do not control the arrival of round-trip time measurements from end hosts, they can potentially be overloaded if the arrival pattern is bursty.

A common concern that affects all three approaches is that if the fundamental assumption about the stability of network distance (i.e. round-trip propagation delay) does not hold due to frequent network topology changes, all three distance prediction approaches would suffer badly in prediction accuracy. The level of impact such problem has on each distance prediction technique is out of the scope of this paper. However, we do believe that Internet paths are fairly stable as Zhang et al's Internet path study in 2000 reported that roughly 80% of Internet routes studied were stable for longer than a day [9]. In addition, because propagation delay is somewhat related to geography, a route change need not directly imply a large change in propagation delay excepting for pathological cases.

A. Other Applications of GNP

We want to point out that using GNP for network distance predictions is only one particular application. The fundamental difference between GNP and other approaches is that GNP computes *absolute* geometric coordinates to characterize positions of end hosts. In other words, GNP is able to generate a simple mathematical structure that maps extremely well onto the Internet in terms of distances. This structure can greatly

benefit a variety of applications. For example, many scalable overlay routing schemes such as CAN [3] and Delaunay triangulation based overlay [2] achieve scalability by organizing end hosts into a simple abstract structure. The problem is that it is not easy to build such an abstract structure that simultaneously reflects the underlying network topology so as to increase performance [10]. GNP coordinates can be *directly* used in these overlay structures and can potentially improve their performance significantly. Another interesting application of GNP is to build a proxy location service. For example, the GNP coordinates of a large number of network proxies can be organized as a kd-tree data structure. Then, to locate a proxy that is nearest to an end host at a particular set of coordinates, only an efficient lookup operation in this data structure is required. No expensive sorting of distances is needed.

V. Evaluation Methodology

In this section, we describe the methodology we use to evaluate the accuracy of GNP, the triangulated heuristic, and IDMaps using measured Internet distance data.

A. Data Collection

We have login access to 19 hosts we call *probes* in research institutions distributed around the world.[2] Twelve of these probes are in North America, 5 are in Asia Pacific, and 2 are in Europe. In addition to probes, we have compiled several sets of IP addresses that respond to ICMP ping messages. We call these IP addresses *targets*.

To collect a data set, we measure the distances between the 19 probes and the distances from each probe to a set of targets. To measure the distance between two hosts, we send 220 84-byte ICMP ping packets at one second apart and take the minimum round-trip time estimate from all replies as the distance. This raw data is then post-processed to retain only the targets that are reachable from all probes. Correspondingly, there is a bias against having targets that are not always-on (e.g. modem hosts) or do not have global connectivity in our final targets set.

We have collected two data sets. The first set, collected over a two-day period in the last week of May 2001, is based on a set of targets that contains 2000 "ping-able" IP addresses obtained at an earlier time. These IP addresses were chosen via uniform probing over the IP address space such that any valid IP address has an equal chance of being selected. After post-processing, we are left with 869 targets that are reachable from all probes. The relatively low yield is partially due to the case where some targets are not on the Internet during our measurements, and partially due to the possibility that some targets are not globally reachable due to partial failures of the Internet. Using the NetGeo [11] tool from CAIDA, we have found that the 869 targets span 44 different countries. 467 targets are in the United States, and each of the remaining countries contributes fewer than 40 targets. In summary, 506 targets are in North America, 30 targets are in South America, 138 targets are in Europe, 94 targets are in Asia, 24 targets are in Oceania, 12 targets are in Africa, and 65 targets have unknown locations. This *Global* data set allows us to evaluate the global applicability of the different distance prediction mechanisms.

Our second data set, collected over an 8-hour period in the first week of June 2001, is based on a set of 164 targets that are web servers of institutions connected to the Abilene backbone network. After post-processing, we are left with 127 targets that are reachable from all probes. The vast majority of these targets are located in universities in the United States. Note that 10 of our 19 probes are also connected to Abilene. This *Abilene* data set allows us to examine the performance of the different mechanisms in a more homogeneous environment.

B. Experiment Methodology

All three distance prediction mechanisms considered in this paper require the use of some special infrastructure nodes (i-nodes). To perform an experiment using a data set, we first select a subset of the 19 probes to use as i-nodes, and use the remaining probes and the targets as ordinary hosts. This way, we can evaluate the performance of a mechanism by directly comparing the predicted distances and the measured distances from the remaining probes to the targets. Because the particular choice of i-nodes can potentially affect the resulting prediction accuracy, in Section V-C, we propose 3 strawman selection criteria to consider in this study.

There is however an important and subtle issue that we must address. Suppose we want to compare GNP to IDMaps. We can pick a selection criterion to select N i-nodes and conduct one experiment using GNP and one using IDMaps. Unfortunately, when we compare the results, it is difficult to conclude whether the difference is due to the inherent difference in these mechanisms, or simply due to the fact that the particular set of i-nodes happens to work better with one mechanism. To increase the confidence in our results, we use a technique that is similar to k-fold validation in machine learning. Instead of choosing N i-nodes based on a criterion, we choose $N+1$ i-nodes. Then by eliminating one of the $N+1$ i-nodes at a time, we can generate $N+1$ different sets of N i-nodes that are fairly close to satisfying the criterion for N. We then compare different mechanisms by using the overall result from all $N+1$ sets of N i-nodes.

To solve the multi-dimensional global minimization problems in computing GNP coordinates, we use the Simplex Downhill method [8]. In our experience, this method is highly robust and quite efficient. To ensure a high quality solution, we repeat the minimization procedure for 300 iterations when computing Landmarks' coordinates, and for 30 iterations when computing an ordinary host's coordinates. In practice, 3 iterations is enough to obtain a fairly robust estimate.

C. Infrastructure Node Selection

Intuitively, we would like the i-nodes to be well distributed so that the useful information they provide is maximized. Based on this intuition, we propose three strawman criteria to choose N i-nodes from the 19 probes. The first criterion, called maximum separation, is to choose the N probes that maximize the total inter-chosen-probe distances. The second criterion, called N-medians, is to choose the N probes that minimize the total distance from each not-chosen probe to its nearest chosen probe. The third criterion, called N-cluster-medians, is to form

[2] We would like to thank our colleagues in these institutions for granting us host access. We especially thank ETH, HKUST, KAIST, NUS, and Politecnico di Torino for their generous support for this study.

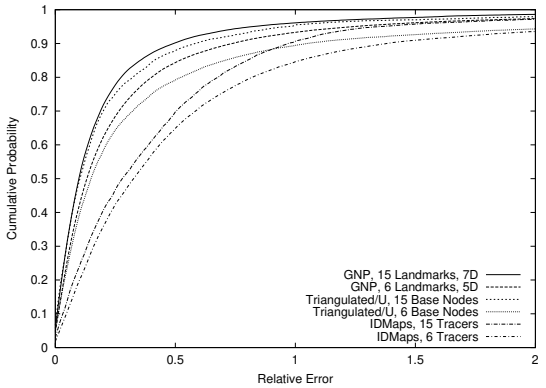

Fig. 5. Relative error comparison (Global)

N clusters of probes and then choose the median of each cluster as the i-nodes. The N clusters are formed by iteratively merging the two nearest clusters, starting with 19 probe clusters, until we are left with N clusters.

In addition, to observe how each prediction mechanism reacts to a wide range of unintelligent i-node choices, we will also use random combinations of i-nodes in this study.

D. Performance metrics

To measure how well a predicted distance matches the corresponding measured distance, we use a metric called **directional relative error** that is defined as:

$$\frac{predicted\ distance - measured\ distance}{\min(measured\ distance, predicted\ distance)} \quad (4)$$

Thus, a value of zero implies a perfect prediction, a value of one implies the predicted distance is larger by a factor of two, and a value of negative one implies the predicted distance is smaller by a factor of two. Compared to simple percentage error, this metric can guard against the "always predict zero" policy. When considering the general prediction accuracy, we will also use the **relative error** metric, which is simply the absolute value of the directional relative error.

To measure the effectiveness of using predicted distances for server-selection type of applications, we use a metric called **rank accuracy**. The idea is that, after each experiment, we have the predicted distances and measured distances for the paths between the non-i-node probes and the targets. We then sort these paths based on the predicted distances to generate a predicted ranked list, and also generate a measured ranked list based on the measured distances. The rank accuracy is then defined as the percentage of paths correctly selected when we use the predicted ranked list to select some number of the shortest paths. If the predicted ranking is perfect, then the rank accuracy is 100% regardless of the number of shortest paths we are selecting. Note that a prediction mechanism can potentially be extremely inaccurate with respect to the directional relative error metric but still have high rank accuracy because the ranking of the paths may still be preserved.

VI. Experimental Results

In this section, we present our experimental results. First, by using the same set of i-nodes (unless otherwise noted, we always use the N-cluster-medians selection criterion with k-fold validation) for each mechanism, we present results to compare the accuracy of GNP, the triangulated heuristic, and IDMaps. Then we compare the effectiveness of the three i-node selection criteria under each mechanism. After that, we present a series of results that are aimed to highlight several interesting aspects of GNP.

# I-Nodes	15	12	9	6
GNP	0.5/7D	0.59/7D	0.69/5D	0.74/5D
Tri./U	0.59	0.69	0.8	1.05
IDMaps	0.97	1.09	1.16	1.39

TABLE I

Summary of 90 percentile relative error (Global)

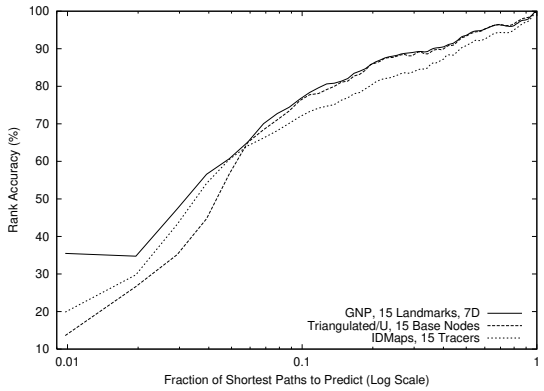

Fig. 6. Rank accuracy comparison (Global)

A. Comparisons Using the Global Data Set

We have conducted a set of experiments using the Global data set to compare the three mechanisms. Figure 5 compares the three mechanisms using the relative error metric when 6 and 15 i-nodes are used. For GNP, the best results are achieved with the Euclidean space model of 5 and 7 dimensions respectively; for the triangulated heuristic, the upper bound heuristic (U) performs by far the best. Note that U is simply the shortest distance between two end hosts via one i-node. Both coordinates-based mechanisms perform significantly better than IDMaps, with GNP achieving the highest overall accuracy in all cases. With 15 Landmarks, GNP can predict 90% of all paths with relative error of 0.5 or less. We will defer the explanation for the differences in accuracy of the three schemes until Section VI-E.

We have also conducted experiments when 9 and 12 i-nodes are used. To summarize all the results, we report the 90 percentile relative error value for all three mechanisms at 6, 9, 12 and 15 i-nodes in Table I. Clearly as the number of i-nodes increase, all three mechanisms benefit, with GNP being the most accurate in all cases. However, the accuracy of IDMaps and triangulated heuristic will eventually become higher than that of GNP as the number of i-nodes increases. Without larger data sets, it will be difficult to understand the asymptotic behavior of each scheme. Nevertheless, it is safe to conclude that with a small number of Landmarks, these differences will be observed.

Figure 6 compares the three mechanisms in terms of the rank accuracy metric when 15 i-nodes are used. The ability to rank the shortest paths correctly is desirable because it is important to server-selection problems. Overall, GNP is most accurate at ranking the paths. In particular, GNP is significantly more

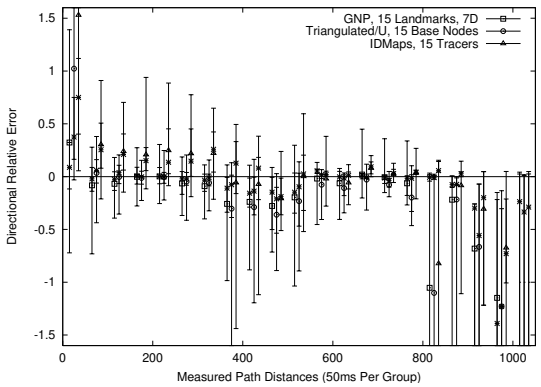

Fig. 7. Directional relative error comparison (Global)

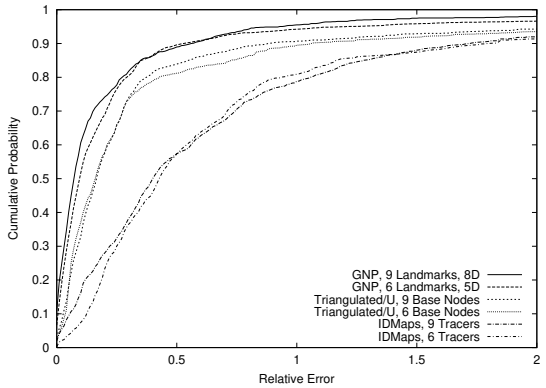

Fig. 8. Relative error comparison (Abilene)

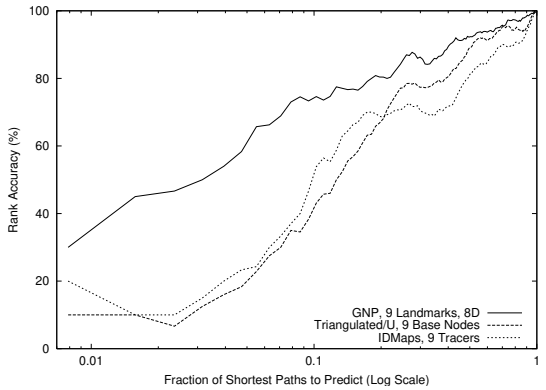

Fig. 9. Rank accuracy comparison (Abilene)

accurate at ranking the shortest 5% of the paths than the triangulated heuristic even though their difference by the relative error measure is small. In fact, even though IDMaps has poor performance in terms of relative error, it is better at ranking the shortest paths than the triangulated heuristic.

The explanation to this seemingly contradictory result can be found in Figure 7. In this figure, we classify the evaluated paths into groups of 50ms each (i.e. (0ms, 50ms], (50ms, 100ms],...,(1000ms,∞]), and plot the summary statistics that describe the distribution of the directional relative error of each mechanism in each group. Each set of statistics is plotted on a vertical line. The mean directional relative error of each mechanism is indicated by the squares (GNP), circles (triangulated heuristic) and triangles (IDMaps). The 5th percentile and 95th percentile are indicated by the outer whiskers of the line, the 25th percentile and 75th percentile are indicated by the inner whiskers. Note that in some cases these whiskers are off the chart. Finally, the asterisk (*) on the line indicates the median.

We can see that GNP is more accurate in predicting short distances than the other mechanisms. Although the triangulated heuristic is more accurate than IDMaps in predicting distances of less than 50ms, IDMaps is very *consistent* in its over-predictions for distances of up-to 350ms. This consistent over-prediction behavior causes IDMaps to rank the shortest paths better than the triangulated heuristic. Beyond 800ms, we see large under-predictions by all mechanisms. However, because these paths account for less than 0.7% of all evaluated paths, the result here is far from being representative. In the last group, there are several outliers of distances of over 6000ms, contributing to the large under-predictions (the means are off the chart between -5 and -6). Finally, notice that paths between 350ms and 550ms appear to be much harder to predict than their immediate neighbors. We will conduct further investigations to try to understand this behavior.

B. Comparisons Using the Abilene Data Set

Now we turn our attention to experiments we have conducted with the Abilene data set using *only* the subset of 10 Abilene-attached probes. Figure 8 compares the three mechanisms when 6 and 9 i-nodes are used. The 6 i-nodes are selected using the N-cluster-medians criterion with k-fold validation, but the 9 i-nodes are obtained simply from eliminating one of the 10 Abilene-attached probes at a time, providing 10 different combinations of 9 i-nodes. For GNP, the best performance is achieved with the Euclidean space model of 5 and 8 dimensions respectively, and for the triangulated heuristic, again the upper bound U heuristic achieves better accuracy than the lower bound or the average of the two. Notice that in the homogeneous environment of Abilene, the accuracy of all three mechanisms barely improves from 6 to 9 i-nodes. We believe that the additional i-nodes simply do not add much more information in such a homogeneous environment.

Comparing to previous results based on the Global data set with 9 i-nodes, the 90 percentile relative error for GNP, the triangulated heuristic and IDMaps are 0.69, 0.8 and 1.16 respectively. Using the Abilene data set with 9 i-nodes, those figures are 0.56, 0.88 and 1.72 respectively. In other words, only GNP's accuracy improves in the more homogeneous environment of Abilene. We believe this is because the paths in Abilene are all very short, 90% of the paths are shorter than 70ms. As a result, the advantage GNP has in prediction short distances is amplified.

Figure 9 compares how well each mechanism rank paths in Abilene when 9 i-nodes are used. The advantage that GNP has in predicting the shortest paths is clear. This is confirmed again in the directional relative error comparison shown in Figure 10. Again, IDMaps' consistent over-predictions for paths of up-to 80ms allow it to be better at ranking the shortest paths than the triangulated heuristic even though it is not accurate in terms of relative error.

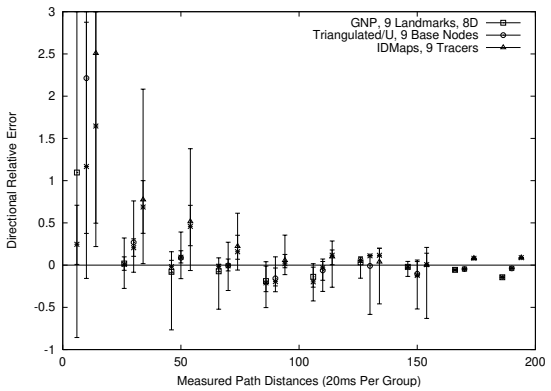

Fig. 10. Directional relative error comparison (Abilene)

	Max	Min	Mean	Std Dev
GNP	0.94	0.65	0.7375	0.06906
Triangulate/U	1.37	0.66	0.8685	0.1686
IDMaps	1.84	1.0	1.287	0.2308

TABLE II

STATISTICAL SUMMARY OF 90 PERCENTILE RELATIVE ERROR UNDER RANDOM I-NODE PLACEMENT

C. Sensitivity to Infrastructure Node Placement

Although the triangulated heuristic is very simple, it lacks robustness because its accuracy is highly dependent on the number and the locations of the base nodes in the network.

To study how sensitive are GNP, the triangulated heuristic, and IDMaps to unintelligent placement of i-nodes, we conduct a set of experiments with 20 random combinations of 6 i-nodes using the Global data set. For each mechanism and each of the 20 random combinations, we compute the 90 percentile relative error value. Table II shows the key statistics of the 90 percentile relative error for each mechanism. Of the three mechanisms, GNP's accuracy is the highest by all measures and also has the smallest spread. Because GNP does not use the virtual topology model, it is highly robust in producing accurate predictions even under random i-nodes placement.

D. Infrastructure Node Selection

In the previous experiments we have been using the N-cluster-medians i-node selection method whenever appropriate. In this section, we go back to examine the differences in the 3 proposed i-node selection criteria. Using the Global data set, we conduct experiments using the 3 criteria under 6 and 9 i-nodes (with k-fold validation) and compute the 90 percentile relative error for each set of experiments. We also take the opportunity here to compare the different triangulated heuristics. Table III summarizes the results.

The N-cluster-medians and N-medians perform very similarly. On the other hand, the Max separation criterion works very poorly because this criterion tends to select probes only in Europe and Asia, and therefore they are not necessarily very well distributed. A comparison with the results reported in Table II reveals that the N-cluster-medians criterion is not optimal because there exists some combinations of 6 infrastructure nodes that can lead to relative error as low as 0.65, 0.66 and 1.0 for GNP, the triangulated heuristic, and IDMaps respectively. Note that the triangulated lower bound heuristic L has poor

$N = 6$	N-cluster-medians	N-medians	Max sep.
GNP	0.74	0.78	1.04
Triangulated/U	1.05	1.08	4.64
Triangulated/L	1.85	1.53	1.93
Triangulated/(L+U)/2	1.53	1.31	3.3
IDMaps	1.39	1.43	5.57
$N = 9$	N-cluster-medians	N-medians	Max sep.
GNP	0.68	0.7	0.83
Triangulated/U	0.8	0.77	1.19
Triangulated/L	2.06	2.0	2.11
Triangulated/(L+U)/2	1.43	1.38	1.69
IDMaps	1.16	1.09	1.74

TABLE III

SUMMARY OF 90 PERCENTILE RELATIVE ERROR UNDER DIFFERENT I-NODE SELECTION CRITERIA

predictive power in general compared to the upper bound U heuristic (the average of U and L always leads to accuracy in between the two bounds). Intuitively, since the $\max$ filter is used in the L metric, it is more sensitive to large outliers in the data. The fact that U works well implies that shortest path routing is still a reasonably close approximation for the majority of cases. There is however an exception. When 6 i-nodes chosen by the maximum separation criterion is used, the L metric performs much better than the U metric. Looking at the set of i-nodes, we discover that except for one i-node in Canada, all other i-nodes are located in Asia and Europe. This is interesting because since the majority of our targets are in North America, they are in between most of the i-nodes. Thus, we have the exact configuration where the L metric is most accurate!

We have also looked at the rank accuracy of the triangulated heuristics in these experiments. For 6 i-nodes, there is no surprise, the difference in rank accuracy of the U, L and $(L+U)/2$ metrics agrees with their difference in relative error. However, for 9 i-nodes, under all three different i-node selection criteria, the L and $(L+U)/2$ metrics have higher rank accuracy by 5 to 12 percents than the U metric for only the shortest 1% of paths. Beyond the shortest 1%, the difference in rank accuracy again agrees with the difference in relative error. Further studies need to be conducted to analyze this anomaly.

E. Sources of Inaccuracy

So far we have only shown the differences in accuracy of the three distance prediction schemes, but where the inaccuracy and differences originate is not clear. In this section, we discuss several sources for the inaccuracy.

1) Inefficient Routing: Since all three distance prediction schemes rely in some degree on shortest (by propagation delay) path routing in the Internet, we believe the largest source for inaccuracy is the inefficient routing behavior in the Internet stemming from BGP policy routing and hop count based routing. To assess the level of inefficient routing in our global data set, we conducted the same triangular inequality test as in [5]. That is for all the triangular closed loop paths (a, b), (b, c), and (a, c) that we measured, we computed all the $(a, c)/((a, b) + (b, c))$ ratios. We found that 7% of the ratios are greater than one, which is consistent with the previous findings. To measure the impact of this on prediction accuracy, we performed the following experiment. For each target t in the global data set, we remove t from consideration if t is in $\{a, b, c\}$ and $(a, c)/((a, b) + (b, c)) > 1.5$. After applying this filter, we are

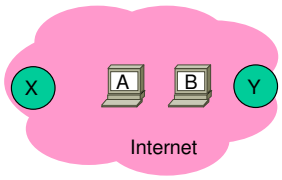

Fig. 11. Predicting short distances

left with 392 targets. We performed the 15 i-nodes experiments again, and found that all three distance prediction schemes' performance improves. For GNP, the 90 percentile relative error is improved from 0.5 to 0.33; for the triangulated heuristic/U, the relative error improved from 0.59 to 0.42; and finally for IDMaps, the relative error improved from 0.97 to 0.89.

2) Predicting Short Distances: A major difference between the performance of the three schemes lie in their ability to predict short distances. As we have shown, GNP is the most accurate in this category and IDMaps is the least accurate and tend to heavily over-predict short distances. The difference is actually easy to explain. Consider the example in Figure 11. X and Y are i-nodes, and A and B are two end hosts that are very nearby. Clearly, IDMaps gives the most pessimistic prediction of $(A,X)+(B,Y)+(X,Y)$. The triangulated heuristic U metric is slightly less pessimistic, since it predicts the distance to be $(A,Y)+(B,Y)$. In contrast, with a one-dimensional model, GNP will be able to perfectly predict the distance between A and B. Although the triangulated heuristic L metric would have given a perfect prediction in this example, in practice it is too easily influenced by a single large distance to an i-node, thus, as we have shown, it works very poorly in practice. GNP is more robust against outliers in measurements since it takes all measurements into account when computing coordinates. In summary, GNP performs better because it exploits the relationships between the positions of Landmarks and end hosts rather than depending on the exact topological locations of the i-nodes, thus it is highly accurate and robust.

F. Exploring the GNP Framework

1) Error Measurement Function: Recall that when computing GNP coordinates, an error measurement function $\mathcal{E}(\cdot)$ is used in the objective functions. Appropriately characterizing the goodness of a set of coordinates is key to the eventual predictive power of those coordinates. In Section III, we mentioned the squared error measure (Eq. 2). However, intuitively, this error measure might not be very desirable because one unit of error in a very short distance accounts for just as much as one unit of error in a very long distance. This leads us to experiment with two other relative error measures. The first one is the normalized error measure:

$$\mathcal{E}(d_{\mathcal{H}_1\mathcal{H}_2}, \hat{d}^{\mathcal{S}}_{\mathcal{H}_1\mathcal{H}_2}) = \left(\frac{d_{\mathcal{H}_1\mathcal{H}_2} - \hat{d}^{\mathcal{S}}_{\mathcal{H}_1\mathcal{H}_2}}{d_{\mathcal{H}_1\mathcal{H}_2}}\right)^2 \quad (5)$$

and the second one is the logarithmic transformed error measure:

$$\mathcal{E}(d_{\mathcal{H}_1\mathcal{H}_2}, \hat{d}^{\mathcal{S}}_{\mathcal{H}_1\mathcal{H}_2}) = (\log(d_{\mathcal{H}_1\mathcal{H}_2}) - \log(\hat{d}^{\mathcal{S}}_{\mathcal{H}_1\mathcal{H}_2}))^2 \quad (6)$$

We perform experiments using the Global data set with 6 and 15 Landmarks selected using the N-cluster-medians criterion (with k-fold validation) and compare the three error measures. Table IV reports the 90 percentile relative error for each experiment. The results confirm our intuition. The normalized measure and the logarithmic measure are very similar because they are both a form of relative error measure. It is clear that the squared error measure is not very suitable. Thus, throughput this paper, to compute GNP coordinates, we have always used the normalized error measure.

# Landmarks	6	15
Normalized error	0.74	0.5
Logarithmic transform	0.75	0.51
Squared error	1.03	0.74

TABLE IV

SUMMARY OF 90 PERCENTILE RELATIVE ERROR FOR DIFFERENT ERROR MEASUREMENT FUNCTIONS

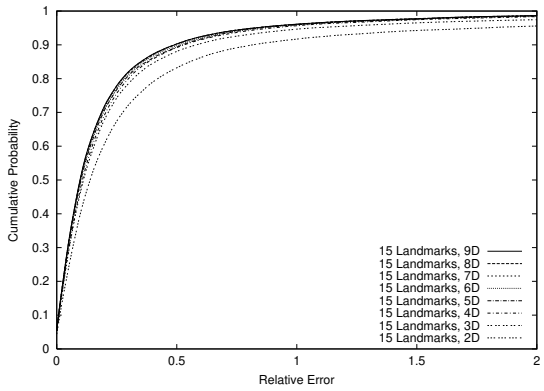

Fig. 12. Convergence of GNP performance

2) Choosing the Geometric Space: Although in the previous experiments we have always reported results with the Euclidean space model of various dimensions, we have also experimented with the spherical surface and the cylindrical surface as potential models. The spherical surface makes sense because the Earth is roughly a sphere, and since almost certainly no major communication paths pass through the two Poles, the cylindrical surface may also be a good approximation. The GNP framework is flexible enough to accommodate these models, the only change is that the distance functions are different. With the Global data set and 6 Landmarks chosen with the N-cluster-medians criterion, we conduct experiments to examine the fitness of the spherical and cylindrical surface of various sizes. For the spherical surface, we specify the radius; for the cylindrical surface, we specify the circumference and the height is taken to be half the circumference. It turns out that both of these models' performance increases as the size of their surface increases, and in the limit approaches the performance of the 2-dimensional Euclidean space model. We believe this is a consequence of the fact that we have no probes in central Asia or Africa, and there are also very few targets in those regions, hence a curved surface does not help.

Focusing on the Euclidean space models, we turn our attention to the question of how many dimensions we should use in GNP. To answer this question, we conduct experiments with the Global data set using 6, 9, 12, and 15 Landmarks chosen with the N-cluster-medians criterion (with k-fold validation) under various number of dimensions. Figure 12 shows the result for

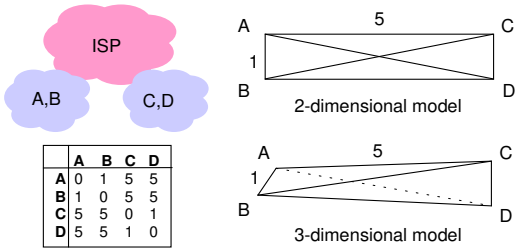

Fig. 13. Benefit of extra dimensions

the case of 15 Landmarks. Generally, as the number of dimensions is increased, GNP's accuracy improves, but the improvement diminishes with each successive dimension. To characterize this effect, consider the cumulative probability distribution functions of the relative error under two different dimensions D and $D+1$. Between the 70 and 90 percentile, if the performance of $D+1$ dimensions is not strictly greater than that of D dimensions, or if the average improvement is less that 0.1%, then we say the results have converged at D dimensions. Using this criterion, for 6, 9, 12, and 15 Landmarks, the results converge at 5, 5, 7, and 7 dimensions respectively.

Intuitively, adding more dimensions increases the model's flexibility and allows more accurate coordinates to be computed. To illustrate, consider the situation shown in Figure 13 where there are four hosts, $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, and $\mathcal{D}$, with $\mathcal{A}$ in the same network as $\mathcal{B}$, and $\mathcal{C}$ in the same network as $\mathcal{D}$. The hypothetical measured distances between them are shown in the matrix. Clearly, in a 2-dimensional space, the distances cannot be perfectly modeled. One possible approximation is the rectangle of width 5 and height 1, preserving most of the distances, except the diagonal distances are over-estimated. However, in a 3-dimensional space, we can perfectly model all the distances with a tetrahedron. Of course, any Euclidean space model is still constrained by the triangular inequality, which is generally not satisfied by Internet distances. As a result, adding more dimensions beyond a certain point does not help.

3) Reducing Measurement Overhead: So far we have assumed that an end host must measure its distances to all Landmark hosts in order to compute its coordinates. However, only $D+1$ host-to-Landmark distances are really required for the coordinates computation in a D-dimensional space. To expose the trade-offs, we conducted an experiment with 15 Landmarks and a 7-dimensional Euclidean space model, where we randomly chose 8 out of 15 Landmarks for each end host for the coordinates computation. We found that the 90 percentile relative error of GNP increases from 0.5 to 0.65. However, when we chose the 8 Landmarks that are nearest to each end host for the computations, the prediction accuracy is virtually unchanged! While further study of this technique is needed, it seems feasible to greatly reduce the measurement overhead without sacrificing accuracy.

4) Why Not Geographical Coordinates?: Finally, we ask whether GNP is simply discovering the geographical relationships between hosts. If so, then a straight forward alternative is to use the geographical coordinates (longitude and latitude) of end hosts to perform distance estimates. We obtain the approximate geographical coordinates for our probes and targets in the Global data set from NetGeo [11]. Although more sophisticated techniques than NetGeo have been proposed [12], the NetGeo tool is publicly available and so we use it as a first approximation. We compute the linear correlation coefficient between geographical distances and measured distances, and also between GNP computed distances and measured distances. Excluding the outliers of measured distances greater than 2500ms, the overall correlation between geographical distances and measured distances is 0.638, while the overall correlation between GNP distances and measured distances is 0.915. Knowing that the NetGeo tool is not 100% accurate, we note with caution that the performance gap between GNP distances and the geographical distances led us to believe that GNP is indeed discovering network specific relationships beyond geographical relationships.

VII. Summary

In this paper, we have studied a new class of solutions to the Internet distance prediction problem that is based on end hosts-maintained coordinates, namely the previously proposed triangulated heuristic and our new approach called Global Network Positioning (GNP). We propose to apply these solutions in the context of a peer-to-peer architecture. These solutions allow end hosts to perform distance predictions in a timely fashion and are highly scalable. Using measured Internet distance data, we have conducted a realistic Internet study of the distance prediction accuracy of the triangulated heuristic, GNP and IDMaps. We have shown that both the triangulated heuristic and GNP out-perform IDMaps significantly. In particular, GNP is most accurate and robust.

We have also explored a number of key issues related to the GNP approach to maximize performance. The main finding is that a relative error measurement function combined with a Euclidean space model of an appropriate number of dimensions achieves good performance. We will continue to develop solutions around the GNP framework in the future.

References

[1] Y. Chu, S. Rao, and H. Zhang, "A case for end system multicast," in *Proceedings of ACM Sigmetrics*, June 2000.
[2] J. Liebeherr, M. Nahas, and W. Si, "Application-layer multicast with delaunay triangulations," Tech. Rep., University of Virginia, Nov. 2001.
[3] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," in *Proceedings of ACM SIGCOMM'01*, San Diego, CA, Aug. 2001.
[4] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for Internet applications," in *Proceedings of ACM SIGCOMM'01*, San Diego, CA, Aug. 2001.
[5] P. Francis, S. Jamin, V. Paxson, L. Zhang, D.F. Gryniewicz, and Y. Jin, "An architecture for a global Internet host distance estimation service," in *Proceedings of IEEE INFOCOM '99*, New York, NY, Mar. 1999.
[6] S.M. Hotz, "Routing information organization to support scalable interdomain routing with heterogeneous path requirements," 1994, Ph.D. Thesis (draft), University of Southern California.
[7] J.D. Guyton and M.F. Schwartz, "Locating nearby copies of replicated Internet servers," in *Proceedings of ACM SIGCOMM'95*, Aug. 1995.
[8] J.A. Nelder and R. Mead, "A simplex method for function minimization," *Computer Journal*, vol. 7, pp. 308–313, 1965.
[9] Y. Zhang, V. Paxson, and S. Shenker, "The stationarity of internet path properties: Routing, loss, and throughput," Tech. Rep., ACIRI, May 2000.
[10] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, "Topologically-aware overlay construction and server selection," in *Proceedings of IEEE INFOCOM'02*, New York, NY, June 2002.
[11] CAIDA, "NetGeo - The Internet geographic database," http://www.caida.org/tools/utilities/netgeo/.
[12] V.N. Padmanabhan and L. Subramanian, "An investigation of geographic mapping techniques for internet hosts," in *Proceedings of ACM SIGCOMM'01*, San Diego, CA, Aug. 2001.