

# PCoord: Network Position Estimation Using Peer-to-Peer Measurements

Li-wei Lehman and Steven Lerman  
Massachusetts Institute of Technology  
Cambridge, MA 02139  
Email: {lilehman, lerman}@mit.edu

**Abstract**—Several recently emerged Internet services make use of application-level or overlay networks. Examples of such services include overlay multicast, structured peer-to-peer lookup services, and peer-to-peer file sharing. Many of these services could benefit from a network distance prediction mechanism that estimates inter-host latencies. In this paper, we present PCoord, a distributed network coordinate system for overlay topology discovery and distance prediction. PCoord assigns coordinates to hosts on the Internet so that the Euclidean distances between hosts’ coordinates accurately predict their network latencies. Most of the existing network coordinate systems rely on a fixed set of landmark nodes for coordinate computation. PCoord, in contrast, is a fully decentralized system that allows participating hosts in an overlay network to collaboratively construct an accurate geometric model of the overlay topology using a small number of peer-to-peer measurements. Under the PCoord framework, we present three different peer sampling and coordinate mapping schemes: RandPCoord, ClusterPCoord, and ActivePCoord. Through extensive simulations using both real network measurements and simulated topologies, we show that the geometric model constructed by PCoord predicts pair-wise host distances accurately and efficiently.

## I. INTRODUCTION

Several recently emerged Internet services make use of application-level or overlay networks. Examples of such services include distributed content delivery services, overlay multicast [2], structured peer-to-peer lookup services [21], [24], [18], [16], and peer-to-peer file sharing. Topological information can benefit many of these services. To help with the performance of these services, much research has been done to allow end hosts to discover network topology and accurately predict network distances in a scalable and timely fashion. Most of the existing network distance prediction schemes rely to some extent on distance measurements to a common set of reference nodes. For example, in IDMaps [6], hosts called Tracers are deployed in the network to measure distances among themselves and to nearby hosts in a range of IP addresses. The Global Network Positioning (GNP) system [12] uses a host’s distance measurements to a fixed set of landmarks to compute absolute coordinates to characterize the host’s location on the Internet. More recently proposed coordinates-based approaches allow end hosts to be used as landmarks [15], [22], [10], [5], [3], [11]. However, most of

these schemes either are not fully decentralized, or do not address the peer selection problem for coordinate updates.

In this paper, we present PCoord, a peer-to-peer network coordinate system for overlay topology discovery and distance prediction. In PCoord, the network is modeled as a D-dimensional geometric space. Each end host computes its coordinates in this geometric space to characterize its network location in the overlay. In contrast to most existing works, our goal is not to provide an infrastructure service that performs network distance prediction between any arbitrary points on the Internet. Instead, the goal of PCoord is for participating peer nodes in an overlay network to collaboratively construct an accurate geometric model of the overlay network topology in a fully decentralized, peer-to-peer fashion. In PCoord, each host computes its own coordinates by probing only a small number of other peer nodes in the overlay.

PCoord differs from other decentralized coordinate systems in that PCoord peers actively exchange messages to collaboratively discover the overlay topology and their respective network neighborhoods. Each peer then utilizes the discovered topological information to select appropriate peers for coordinate updates. In order to distinguish from the GNP “landmarks”, which are fixed nodes embedded in the network, we call the set of peers selected by a PCoord host for coordinates computation *waypoints*.

This paper evaluates the performance characteristics of PCoord under several factors, including peer distance distribution, and the number of peer-to-peer distance measurements used. In particular, we explore the effects of different waypoint selection strategies. We present three PCoord-based schemes: RandPCoord, ClusterPCoord, and ActivePCoord. The three schemes differ mainly in how each host samples other peers to function as its waypoints.

RandPCoord and ClusterPCoord assume that an arbitrary subset of the peers function as bootstrap nodes to compute the initial set of reference coordinates. In RandPCoord, a peer node randomly selects from existing peer nodes to function as its waypoints. In ClusterPCoord, each peer node selects its waypoints by exploiting the topological information derived based on existing peer nodes’ coordinate values. ActivePCoord does not rely on any bootstrap nodes. Each node goes through an iterative calibration process to refine its coordinates. In each iteration, peers exchange messages to collaboratively discover the overlay topology and their respective network neighborhoods. Peers use triangulated distances to other peers

to help select appropriate peers as waypoints.

Through extensive simulations using both real network measurements and simulated topologies, we compare the performance of RandPCoord, ClusterPCoord, and ActivePCoord with the original GNP scheme (referred to as the FixedLM scheme from now on). Our simulation results suggest that PCoord can achieve competitive prediction accuracy in comparison to the FixedLM scheme without relying on a fixed set of landmark nodes. Our findings are summarized as below.

- RandPCoord prediction accuracy converges to that of the FixedLM scheme when a reasonably large number of waypoint set (e.g. 20 to 30 waypoints) is used. When the number of waypoints used is small (e.g., 10 waypoints), ClusterPCoord achieves performance comparable to that of the FixedLM scheme by selecting well-distributed set of peers as waypoints.
- The FixedLM and PCoord approaches have rather different performance characteristics. The FixedLM scheme tends to underpredict larger RTTs. RandPCoord and ClusterPCoord, in contrast, tend to overpredict small RTTs.
- ActivePCoord can achieve pair-wise distance prediction accuracy comparable to that of the FixedLM scheme without relying on a fixed set of landmarks or any bootstrap nodes. In particular, our simulation results show that in a simulated overlay network consisting of over 3,400 peer nodes, ActivePCoord can predict over 90% of the distances with relative prediction error less than 0.5 after each host performing approximately 15 iterations of coordinate updates.
- ActivePCoord outperforms the FixedLM scheme in finding nearest neighbors by aggressively probing and including each host’s estimated nearby neighbors in its waypoint set at each iteration. Using a simulated overlay network consisting of over 3,400 peer nodes, our results indicate that over 90% of the ActivePCoord peers can locate their closest peers within six to eight iterations of coordinate update by probing only a small fraction of the global peer population.
- The performance of the FixedLM approach can be very sensitive to the landmark placements. The FixedLM scheme performs substantially worse when the peer nodes being modeled are clustered (relative to the landmark locations) in the network, or when the landmarks chosen are not well distributed in the network topology. In particular, it tends to underpredict larger RTTs significantly. In PCoord, since the waypoints are dynamically chosen from the existing peer nodes, the waypoint selection automatically adapts to the topological distribution of the peer nodes.

In the following sections, we first briefly describe the FixedLM and the PCoord approach. We then evaluate the PCoord approach extensively through simulations using both real network measurements and simulated topologies.

## II. THE PCOORD APPROACH

The landmark-based architecture has been commonly adopted in the networking community as a mechanism to

measure and characterize a host’s location on the Internet [12], [13], [16], [17], [15], [8]. In most existing landmark based approaches, end hosts use the distance measurements to a common, fixed set of landmarks to derive location estimations on the Internet. However, using a fixed set of landmarks presents a potential performance bottleneck. More importantly, as we will show in this paper, the accuracy of the fixed landmark schemes often depends highly on the strategic placement of the landmarks.

In PCoord, there are no specially designated landmark nodes. Each PCoord node  $x$  measures its round trip latency to  $K$  other peer nodes, and obtains those  $K$  nodes’ coordinates. Peer node  $x$  then updates its own coordinates to minimize the squared normalized difference between the measured and computed distances with those  $K$  peer nodes using the Simplex Downhill algorithm.

We describe three PCoord-based schemes: RandPCoord, ClusterPCoord, and ActivePCoord. The first two schemes assume that an arbitrary subset of the peers function as bootstrap nodes and compute the bootstrap coordinates; nodes that join subsequently can compute their coordinates by measuring their latencies to any peers with known coordinates. The RandPCoord and ClusterPCoord schemes were previously proposed in [9], independent of PIC [3]. Although one of the PIC strategies (namely, the random strategy) has the same basic algorithm as RandPCoord, PIC [3] did not explore the behavior of the random strategy in depth. Further, it did not address the issue of selecting well-distributed peers as reference points. In this work, we examine the performance of RandPCoord as a function of the number of waypoints used, and explore the effect of bootstrap nodes placement.

### A. RandPCoord

As part of the bootstrap process, RandPCoord assumes that an arbitrary set of initial peer nodes function as bootstrap waypoints to provide reference coordinates to orient other nodes. The bootstrap waypoints measure the inter-node round-trip ping times to produce an  $M \times M$  distance matrix, where  $M$  is the number of bootstrap nodes. A set of coordinates are computed for the  $M$  bootstrap nodes to minimize the overall error between the measured distances and the computed distances. A peer node is said to have been **mapped** once it has derived its absolute coordinates. In order for a node  $x$  to compute its coordinates, it selects any  $K$  existing mapped peer nodes to function as its waypoints.

### B. ClusterPCoord

The idea of ClusterPCoord is to have each peer node select its waypoints by exploiting the topological information derived from existing coordinates. Each peer node uses the following heuristic for waypoint selection.

- Upon joining, a peer node  $x$  contacts any existing peer node  $y$  in the system to obtain a copy of the existing PCoord map. The map contains the IP addresses of existing peer nodes known to node  $y$ , and their respective coordinates.

- The existing peer nodes are divided into clusters based on their coordinates in the geometric space. The Euclidean distances between nodes' positions in the geometric space are used to define the cluster.
- Node  $x$  then randomly picks  $K$  clusters from the clusters formed above, and then randomly picks a node in each cluster as its waypoint. By picking each waypoint node from a different cluster, we attempt to achieve a well-dispersed set of peers as waypoints.

The clustering can be done offline by existing peer nodes in the system, so that a newly joined peer node can quickly select a set of peers as waypoints based on the clustering definition.

### C. ActivePCoord

ActivePCoord does not require a subset of peers to function as bootstrap waypoints. In ActivePCoord, peer nodes initialize their coordinates to the origin. Each peer node then goes through an iterative calibration process to compute its coordinates. At each iteration, each host selects  $K$  peers as reference points for coordinates computation.

Each ActivePCoord peer selects a well-distributed set of other peers to function as its waypoints. Unlike ClusterPCoord, which utilizes topological information from existing mapped coordinates, each ActivePCoord peer uses triangulated distances to other peers to help ensure that the peers selected are well distributed in network. The advantage of using triangulated distances is that there is no need for each peer to maintain a current database of other peers' coordinates, which can change over time as other peers iterate through the calibration process.

We assume that each peer node initially knows of  $M$  other peers in the overlay. In order for each peer to discover other peers in the same overlay, peers exchange a list of peers they know of at each iteration. At each round of the calibration process, each node  $x$  maintains two lists:  $R$ , a list of peers whose RTTs to  $x$  are known to  $x$ , and  $T$ , a list of peers which  $x$  has triangulated distances for so far. We first summarize the notations below and then describe how to compute triangulated distances.

$K$  = Number of waypoints

$R$  = Peer list with known RTT

$T$  = Triangulated peer list

$P = R \cup T$

$Y$  = Set of peers selected as waypoints

$C_x$  = Coordinates of host  $x$

$C_Y$  = Set of coordinates of hosts in  $Y$

$RTT(x, y)$  = RTT between  $x$  and  $y$

A peer node  $x$  can compute its triangulated distance to another peer  $y$  if they both have measured latency to a common node  $z$ . In particular, the distance between  $x$  and  $y$  is lower bounded by  $L = |RTT(x, z) - RTT(y, z)|$  and upper bounded by  $U = RTT(x, z) + RTT(y, z)$ . There are three common ways to estimate two peers' triangulated distance: upper bound  $U$ , lower bound  $L$ , and their average  $A (= \frac{L+U}{2})$  [12].

The algorithm used by each peer  $x$  to calibrate its coordinates is presented as follows.

// $R_x$  is the peer list  $R$  of host  $x$

// $R_{Y_i}$  is the peer list  $R$  of host  $Y_i$

For each iteration

$Y = \text{Select } K \text{ waypoints from } P$

For each  $Y_i$  in  $Y$

Measure RTT to  $Y_i$ ;

Send<sub>dest= $Y_i$</sub>  (RTT( $x, Y_i$ ),  $R_x$ );

Receive<sub>sender= $Y_i$</sub>  ( $C_{Y_i}$ ,  $R_{Y_i}$ );

T.add( $R_{Y_i}$ );

End for each  $Y_i$

Compute Coordinates ( $C_Y$ , RTT( $x, Y$ ));

End iteration

In the first iteration, the waypoints are randomly selected from its  $M$  logical overlay neighbors. For future iterations, the following section describes the strategy used by each peer to select its waypoints.

1) *Selecting Well-Distributed Peers as Waypoints:* At each iteration, a peer utilizes its peer list to select  $K$  well-separated peers to function as its waypoints. A peer node  $x$  selects its waypoints by sampling from the combined RTT peer list  $R$  and triangulated peer list  $T$ .  $P$  is the combined list of peers, with either known actual RTT measurements to  $x$  or triangulated distances to  $x$ . To select a list of peers that are well separated in network,  $x$  randomly selects a set of peers from  $P$  that are at least  $MinDist$  apart from each other in terms of their triangulated distances, where  $MinDist$  is an adjustable threshold set by  $x$ .

2) *Discovering Nearby Peers:* In order to improve the coordinates' accuracy in modeling short distances, each PCoord peer includes an estimated near peer in its waypoint set. However, locating a node's nearest neighbor in a large-scale distributed system is a difficult problem in itself. In PIC [3], each peer node performs a greedy walk to locate a nearby peer using the node's current coordinates to guide the walk. Although the strategy has been shown to work well in an MSPastry framework where each node points to a mix of near and far away nodes, it is unclear whether the strategy would be effective in an unstructured overlay, where a node's logical neighbors may not have such a convenient mix of near and far nodes.

In this section, we describe PCoord's algorithm in discovering nearest peers based on triangulated distances. A peer node periodically probes nodes in its triangulated peer list  $T$  to discover its nearest peer node. In particular, it probes the peers with minimum triangulated distances in its peer list  $T$ . The probed nodes and their corresponding RTTs are then stored in the peer list  $R$ . The following is the pseudocode used by each peer node to update its peer list  $R$  at each iteration to find its nearest peer node.

n1 = T.remove(peer with min upper bound  $U$ )

n2 = T.remove(peer with min lower bound  $L$ )

n3 = T.remove(peer with min average  $A$ )

Probe(n1, n2, n3)

R.add(n1, n2, n3)

The node with the closest actual RTT in the peer list  $R$  is then included in the waypoint set for the next calibration iteration.

### III. EVALUATION METHODOLOGY

We evaluate the PCoord approach extensively through simulations using both real network measurements and simulated topologies. We compare the performance of PCoord with the FixedLM scheme in terms of pairwise distance prediction and nearest neighbor selection.

#### A. Performance Metrics

As in GNP [12], we use the absolute relative error (RE) as our performance metric. For each pair of nodes, their absolute relative error is defined as  $\frac{|E-A|}{\min(E,A)}$ , where  $E$  is the predicted Euclidean distance, and  $A$  is the actual measured RTT (round trip time) between the two nodes. The directional relative error is  $\frac{E-A}{\min(E,A)}$ .

We use the stretch metric to evaluate the effectiveness of the proposed schemes in selecting nearest neighbor. Let  $RTT(x,y)$  denote the measured round-trip latency between node  $x$  and  $y$ . Given a Euclidean mapping of a set of peer nodes, the stretch of a node  $x$  is computed as  $\frac{RTT(x,p)}{RTT(x,q)}$ , where  $p$  is the node with minimal Euclidean distance to  $x$ , and  $q$  is the closest neighbor to  $x$  according to the actual latency measurement.

#### B. Data Collection

We evaluate our scheme using both real network measurements and simulated topologies:

- The Active Measurement Project (AMP) at the National Laboratory for Applied Network Research (NLNR) collects network measurements between over 100 active monitors distributed over the Internet [7]. We use the RTT measurements between 110 hosts on July 16, 2002, and between 104 hosts on January 30, 2003 for our experiments.
- We use the PlanetLab all-pairs-ping data set collected on May 10, 2004. After postprocessing to eliminate missing data, we derived end-to-end latency data among 127 nodes.
- The GT-ITM Internet Topology Generator was used to generate transit stub topologies of a 10,000 node network. We then randomly select 3492 out of the 10,000 nodes as peer nodes of our test overlay network.

Due to space constraints, we present only the GT-ITM results here.

#### C. Simulation Setup

For the RandPCoord and ClusterPCoord schemes, ten experiments with different selection of bootstrap waypoints were performed for each topology. For each experiment, the same set of hosts that serve as bootstrap waypoints in the RandPCoord and ClusterPCoord are also used as the fixed landmarks in

the FixedLM scheme. Unless otherwise noted, the landmark nodes and the bootstrap waypoints are randomly selected  $K$  hosts from the peer population. In one of the later sections, we examine the performance effect of biased selection of landmarks. The default dimension of the geometric space used is five.

### IV. PERFORMANCE EVALUATION

#### A. RandPCoord Performance

To understand the performance characteristics of the RandPCoord vs. the FixedLM scheme, we plot the summary statistics that describe the distance prediction error of both schemes as a function of the number of waypoints used. Figure 1 plots the median, 5th, 25th, 75th, and 95th percentile directional relative error (DRE) of both schemes as a function of the number of waypoints. We note that a zero value in DRE indicates a perfect prediction in the network distance. A positive DRE value indicates an over prediction in network distance, while a negative DRE value indicates an underestimation of actual network distance.

We note that RandPCoord performs worse than FixedLM when the number of waypoints is low. In particular, when six and ten waypoints are used, RandPCoord has a tendency to over predict network distances between hosts, as can be observed from the large positive 95th percentile DRE value in Figure 1. The FixedLM scheme, on the other hand, has a tendency to under predict inter-host distances when the number of landmarks is low. This can be observed from the large negative 5th percentile DRE values in Figure 1.

We note that for both schemes, the DRE values improve monotonically with increasing numbers of landmarks/waypoints. For RandPCoord, the performance improvement is especially significant when the number of waypoints is increased from 6 to 15. The performance of both schemes tends to flatten beyond 25 landmarks. An important observation is that the performance of RandPCoord comes close to that of the FixedLM scheme when increasingly large numbers of waypoints are used.

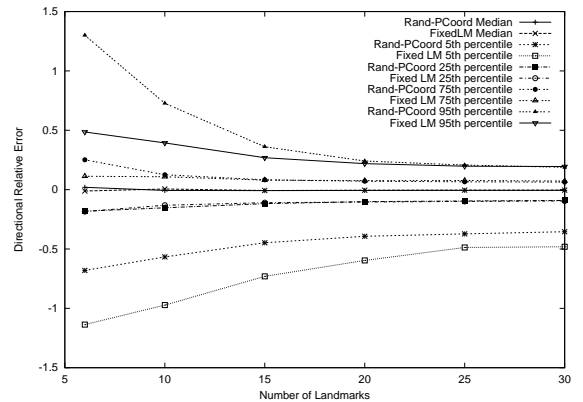


Fig. 1. Directional Relative Error. GT-ITM,  $N = 3492$ .

From the previous figure, we observe that the FixedLM scheme tends to under-predict inter-host distances while the RandPCoord scheme tends to over-predict. To understand

the sources of these under- and over-predictions, we further investigate the performance properties of both schemes by classifying the evaluated actual latencies into groups of 50 ms each.

We show the summary statistics of the *RTT prediction error*, defined as (predicted RTT - actual RTT), for each RTT group. Figures 2 and 3 show the median, mean, 5th, 25th, 75th and 95th percentile prediction error of each RTT group using FixedLM and RandPCoord respectively. Ten landmarks/waypoints are used for both figures. Figure 2 shows that the FixedLM scheme is very good at predicting the distances of less than 50 ms, but tends to over-predict distances that are beyond 250 ms.

Figure 3 shows that the RandPCoord scheme has the most trouble in predicting short distances. The 95th and 75th percentile prediction errors are as high as 694 and 385 ms respectively, showing a gross over-estimation of distances less than 50 ms. The RandPCoord scheme also tends to underestimate distances over 700ms, although the extent of the under-estimation is not nearly as bad as the over-estimation for the 50 ms group case.

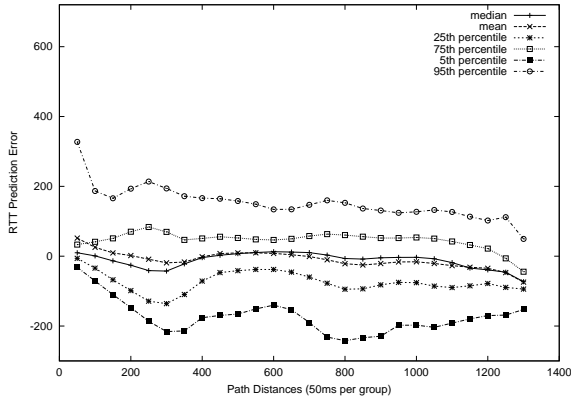


Fig. 2. FixedLM performance by RTT groups using 10 landmarks. GTITM,  $N = 3492$ , 5 Dimensions.

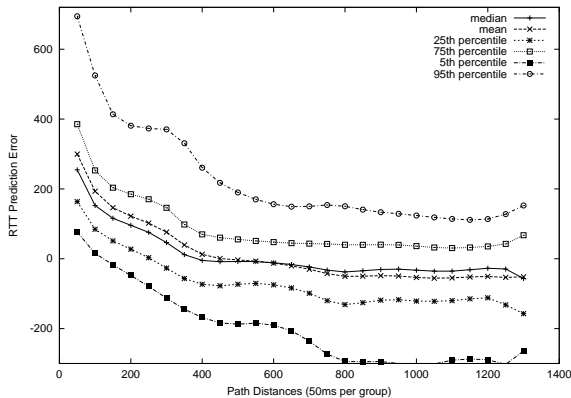


Fig. 3. RandPCoord performance by RTT groups using 10 waypoints. GTITM,  $N = 3492$ , 5 Dimensions.

## B. ClusterPCoord Performance

The difference between RandPCoord and ClusterPCoord is that, in RandPCoord, peer nodes randomly select  $K$  nodes from the PCoord map; ClusterPCoord selects the  $K$  nodes by exploiting the cluster information in the PCoord map. We assume that each existing peer node in the system has access to a copy of the current PCoord map upon joining. The PCoord map contains the IP addresses of existing peer nodes, and their coordinates values in the geometric space.

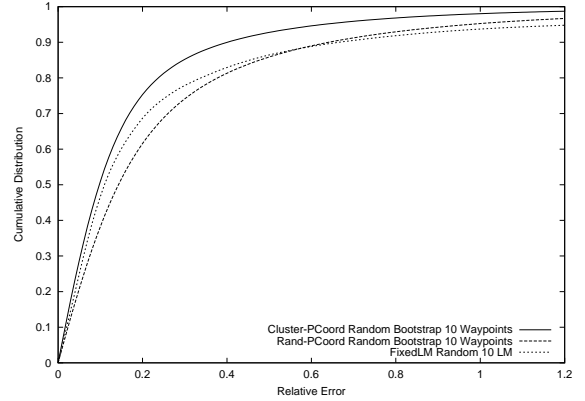


Fig. 4. Relative error distribution. GT-ITM,  $N = 3492$ , 10 waypoints.

Figure 4 compares ClusterPCoord with RandPCoord and FixedLM schemes using the GT-ITM topology. Figure 4 indicates that the performance of ClusterPCoord is better than both the RandPCoord and the FixedLM schemes when ten waypoints are used. Figure 5 shows the summary statistics of the ClusterPCoord scheme under random bootstrap node placement. ClusterPCoord performs significantly better than the RandPCoord scheme when the same number of waypoints is used (compare summary statistics in Figures 5 and 3).

We have examined the RandPCoord and ClusterPCoord performance using different number of waypoints. Interested readers are referred to [9] for more details. In general, the ClusterPCoord scheme significantly outperforms the RandPCoord scheme when the number of waypoints used is low. When the number of waypoints used is large (e.g., 30 waypoints used by each peer), RandPCoord performs as well as the ClusterPCoord.

## C. Nearest Peer Node Selection and Proximity Clustering

The ability to select the nearest node from a set of peer nodes is important to many applications, including nearest server/proxy selection, proximity routing in peer-to-peer networks and neighbor selection in overlay network construction. We use the stretch metric defined earlier as our performance metric.

Figure 6 shows the cumulative distribution of the stretch. We note that both RandPCoord and ClusterPCoord perform worse than the FixedLM scheme in the nearest neighbor selection. This result should not come as a surprise. As discussed in the earlier section, the RandPCoord scheme tends to grossly over-estimate RTT distances that are between 0 and 50 ms, which

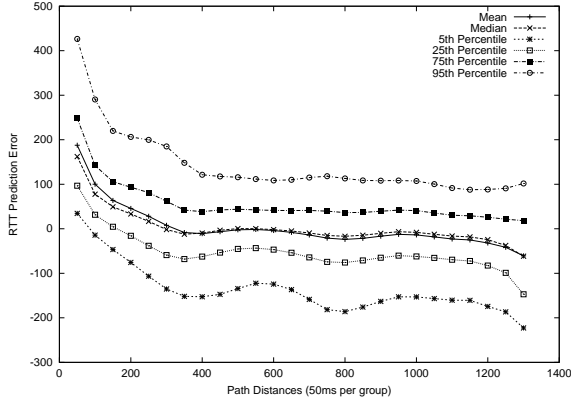


Fig. 5. Summary statistics of RTT prediction error for the ClusterPCoord scheme. GT-ITM,  $N = 3492$ , 10 waypoints.

negatively affects the nearest node selection performance of the RandPCoord scheme.

In order to further understand how well each scheme captures the network proximity relationships among hosts, we apply the KMeans clustering algorithm on the coordinates generated by each scheme. The clustering criterion is the inter-host Euclidean distances defined by the coordinates. We then compute the weighted intra-cluster RTT averages for each clustering assignment, where the weight is the number of peers in each cluster, and the averages are computed using actual RTTs among hosts assigned to the same cluster. Figure 7 shows the results of clustering performance when 10 and 30 landmarks/waypoints are used. The RandPCoord performs significantly worse than the other schemes when the number of waypoints used is small. We note that ClusterPCoord with the same number of waypoints yields cluster averages that are approximately 25% less than that of the RandPCoord. When the number of waypoints is increased to 30 both RandPCoord and ClusterPCoord yield cluster averages that are close to those of the FixedLM schemes.

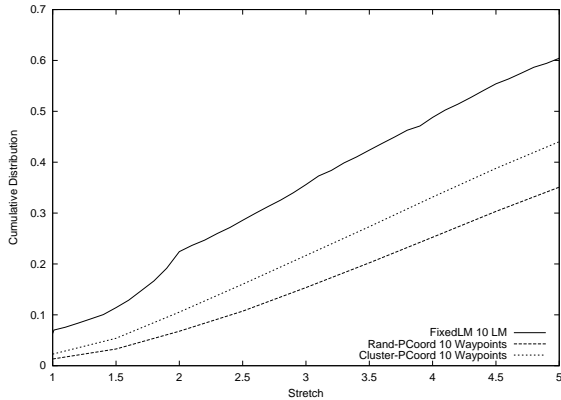


Fig. 6. Performance of selecting nearest peer node. GT-ITM,  $N = 3492$ .

#### D. Robustness in Landmark Placement

The results we have presented so far randomly select from a global pool of peer nodes to function as landmarks or bootstrap

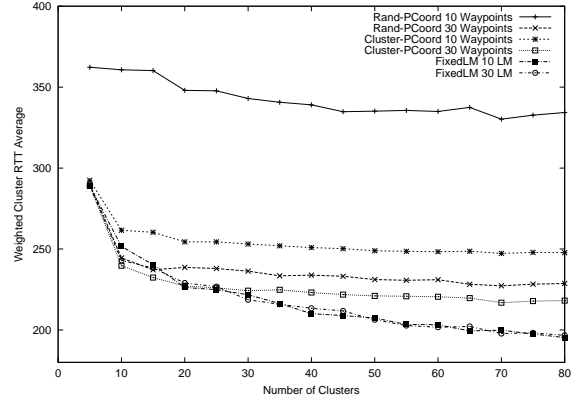


Fig. 7. Weighted Intra-Cluster RTT. Randomly selected bootstrap waypoints. GT-ITM,  $N = 3492$ , 10 waypoints.

waypoints. In the FixedLM scheme, this randomly selected set of peer nodes are used by all other peer nodes to construct their solution coordinates. In the RandPCoord and ClusterPCoord schemes, this randomly selected set of peer nodes function as the bootstrap nodes that provide a set of reference coordinates to other peer nodes.

In this section, we compare the performance of PCoord with the FixedLM scheme when the landmark placement is not well distributed. We generate ten different sets of badly placed landmarks or bootstrap waypoints, which tend to be clustered in network topology, and compare the performance of FixedLM, RandPCoord, and ClusterPCoord.

Figure 8 shows that when the bootstrap waypoints are clustered, both ClusterPCoord and RandPCoord greatly outperform the FixedLM scheme in terms of the relative error distribution. Figure 9 shows the summary statistics of the FixedLM scheme when a clustered landmark set is used. Comparing the summary statistics in Figure 2 using randomly selected landmarks/waypoints, we note that the FixedLM scheme has the tendency to grossly underestimate RTT groups larger than 50 ms when clustered landmarks are used.

We have also tried out a variety of other scenarios in landmark distributions. In general, the FixedLM scheme performs substantially worse when the peer nodes being modeled are clustered (relative to the landmark locations) in the network, or when the landmarks chosen are not well distributed in the network topology.

#### E. ActivePCoord Performance

In this section, we examine the performance of ActivePCoord using the GT-ITM topology. A total of 20 calibration periods were run in the simulation. In each calibration period, each node samples ten peer nodes from its own peer list  $P$  as its waypoints. The adjustable threshold  $MinDist$  is initialized to 300 ms, and is used by each host to select a well-distributed set of peers from its  $P$  list as waypoints; we use  $\frac{(L+U)}{2}$  as the triangulated distance between two peers in the simulations.

At each coordinate update, a peer includes its nearest peer discovered so far (i.e., peer with minimum RTT to itself in the peerlist  $R$ ) in its waypoint set. The RTT and coordinates of the ten selected peers are then used to update the peer's

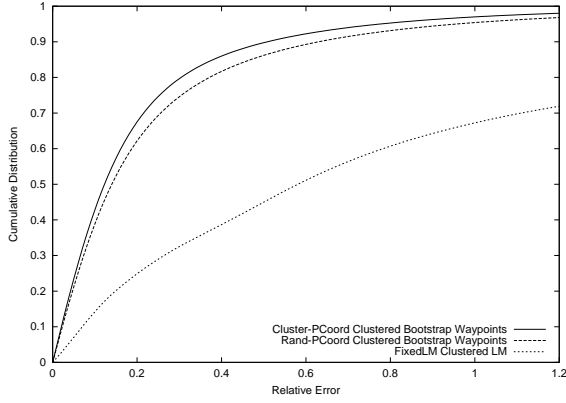


Fig. 8. Relative error distribution using bad landmark or bootstrap waypoints placement. GT-ITM,  $N = 3492$ , 10 landmarks/waypoints.

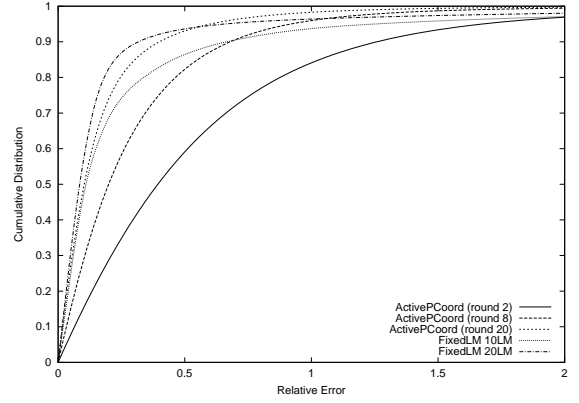


Fig. 10. Relative error for ActivePCoord. GT-ITM,  $N = 3492$ .

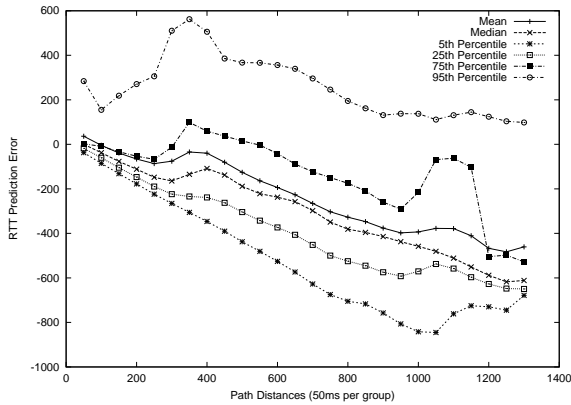


Fig. 9. Summary statistics of RTT prediction error for the FixedLM scheme under clustered landmark placement.  $N = 3492$ , 10 Landmarks.

own coordinates. To refine its search for its nearest neighbor, each peer probes peers with closest triangulated distances in its  $T$  list at each calibration phase. Results presented here are generated by having each peer probe six peers from its  $T$  list in each calibration phase.

The size of the peer list that each peer can exchange is restricted to 30 RTT entries. More specifically, for each selected peer  $Y_i$ ,  $x$  only sends  $Y_i$  the identities of 30 other peers  $x$  believes is closest to  $Y_i$  based on the triangulated distances. In other words,  $x$  selects 30 peers from its peer list  $R$  with minimum triangulated distances to  $Y_i$ , and sends the RTT information to  $Y_i$ . Similarly, each  $Y_i$  only sends  $x$  RTT information of 30 peers with minimum triangulated distances to  $x$ .

Figure 10 shows the cumulative distribution of relative error of ActivePCoord after various stages of the calibration process. Our results indicate that approximately 60%, 82%, 92% of the hosts have relative error less than 0.5 by the end of the second, eighth, and sixteenth iterations. By the end of the 20th iteration, the median relative error in latency prediction is approximately 11%, comparable to the FixedLM performance when 10 and 20 landmarks are used.

Figure 11 shows ActivePCoord's performance in predicting nearest peer node using the coordinates generated at the end of each calibration period. The figure indicates that, under the

FixedLM scheme only 10% of the peers can accurately predict their closest peers using the coordinates generated with 20 landmarks. Using the coordinates generated by ActivePCoord, over 50% of the peers can predict their closest peers by the end of the sixth calibration period. This is a factor of five improvement from the FixedLM performance.

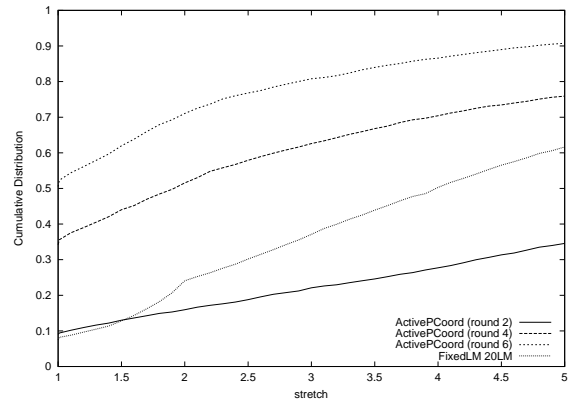


Fig. 11. Nearest peer node prediction. GT-ITM,  $N = 3492$ .

In Figure 12, the stretch metric is defined slightly differently as  $mstretch$  to account for the fact that each ActivePCoord peer maintains RTTs to a list of other peer nodes. For each peer node  $x$ , let  $q$  be its actual closest peer node,  $p1$  be its closest neighbor based on the Euclidean distance computed from the coordinates, and  $p2$  be the peer with minimum RTT in  $x$ 's peer list  $R_x$ . At the end of each round of calibration, we measure the  $mstretch$  metric as  $\frac{\min(RTT(x,p1), RTT(x,p2))}{RTT(x,q)}$ .

Figure 12 indicates that using the  $mstretch$  metric, approximately 18%, 64%, 86%, and 93% of the nodes can predict their nearest peer nodes at end of second, fourth, sixth, and eighth calibration round respectively.

1) *Discussion of ActivePCoord Communication Cost:* ActivePCoord outperforms the FixedLM scheme at the expense of additional communication cost. However, by the end of the eighth round, the cumulative number of peers probed by each node is only 3.7% of the total peer population, and the number of waypoints probed is only 2.3% of the total peer population. Assuming that the 30 peer list entries (IP address and RTT information) exchanged between a peer and each

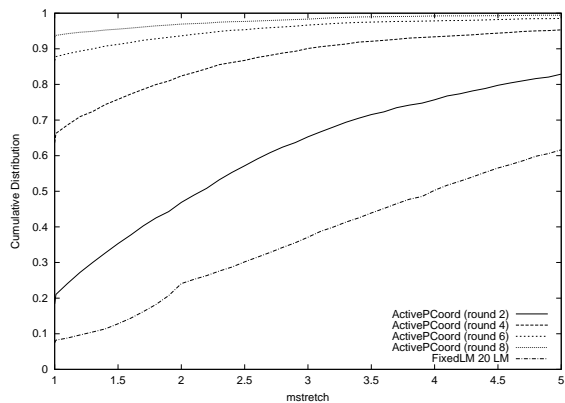


Fig. 12. Nearest peer node prediction with the mstretch metric. GT-ITM,  $N = 3492$ .

of its waypoints is less than 1 Kbytes, the total amount of information exchanged between a host and all of its waypoints over the eight iteration periods is approximately 80 Kbytes cumulatively. We believe this is a rather modest cost for a factor of five improvement in finding nearest neighbor.

## V. RELATED WORK

The IDMaps [6] and GNP [12] are both architectures for a global distance estimation service. Both IDMaps [6] and GNP [12] rely on the deployment of infrastructure nodes. King [14] uses direct online measurements using the DNS infrastructure to predict network latencies between arbitrary Internet end hosts. NPS [11] proposes a hierarchical network position architecture that enables decentralized coordinate computation. The goal of our work, in contrast, is to predict network distances using purely peer-to-peer measurements without relying on the infrastructure services.

To avoid the fixed landmark problem in GNP, several schemes [15], [22], [10] have been proposed that allow hosts to use different subsets of landmarks to construct a local coordinate system, which are then transformed to a global coordinate system. For example, the Lighthouse scheme [15] uses multiple local bases and a transition matrix in vector spaces to allow a host to determine its coordinates relative to any set of landmark nodes. Virtual Landmarks (VLM) and Internet Coordinate System (ICS) both use principal component analysis (PCA) to extract topological information. The above schemes, however, are not fully decentralized.

Several other works focus on the modeling and coordinates computation issues. For example, the Big Bang Simulation (BBS) [19] simulates the error as a potential force field. Shavitt and Tankel [20] recently proposed a hyperbolic coordinate space to model the Internet. The Mithos [23] system uses a spring relaxation technique for coordinate computation.

Several works provide network proximity or location estimates using the distance measurements to a set of well-known landmarks. Examples include the GeoPing algorithm [13], Internet Iso-bar [1], and the binning scheme in [17]. These schemes, in contrast to ours, do not attempt to model Internet hosts using absolute coordinates.

Similar to our work, Vivaldi [4], [5] and PIC [3] also use peer nodes as landmarks. Vivaldi, however, does not address the reference point selection issue. PIC examines the performance of landmark sampling with a mixture of random and close-by nodes. Our scheme differs from PIC in the following aspects. First, PIC does not address the issue of how to select a well-distributed set of peer nodes as landmarks. Secondly, it uses a different strategy to locate nearby peer nodes. Finally, PIC requires a set of peer nodes to compute the bootstrap coordinates. In contrast, ActivePCoord does not require a set of peer nodes to carry out the bootstrap process.

## VI. CONCLUSION

In this paper, we introduce PCoord, a peer-to-peer approach to construct network coordinates for network distance prediction. In PCoord, the network is modeled as a  $D$ -dimensional geometric space. PCoord assigns coordinates to hosts on the Internet so that the Euclidean distances between hosts' assigned coordinates accurately predict their network latencies. In PCoord, each host constructs its own coordinates based on a small number of peer-to-peer measurements without relying on a fixed set of landmarks. Three PCoord-based schemes are presented: RandPCoord, ClusterPCoord, and ActivePCoord.

Through extensive simulations using both real network measurements and simulated topologies, we compare the performance of the three PCoord-based schemes with the GNP scheme using fixed landmarks. Our simulation results indicate that RandPCoord prediction accuracy converges to that of GNP when a reasonably large number of waypoint set (e.g. 20 to 30 waypoints) is used. When the number of waypoints used is small (e.g., 10 waypoints), ClusterPCoord achieves performance comparable to that of GNP by selecting well-distributed set of peers as waypoints.

Using an iterative mapping technique, ActivePCoord is able to achieve pair-wise distance prediction accuracy competitive to that of GNP without relying on a fixed set of landmarks or any bootstrap nodes. In a simulated overlay network consisting of over 3,400 peer nodes, ActivePCoord can predict over 90% of the distances with relative prediction error less than 0.5 after each host performing approximately 15 iterations of coordinate updates. Further, our results indicate that ActivePCoord outperforms GNP in finding nearest neighbors by aggressively probing and including each host's estimated nearby neighbors in its waypoint set at each iteration. Using a simulated overlay network consisting of over 3,400 peer nodes, over 90% of the ActivePCoord peers can locate their closest peers within six to eight iterations of coordinate update by probing only a small fraction of the global peer population. As part of our future work, we plan to investigate the convergence behavior of PCoord under different overlay structure and network topologies.

## ACKNOWLEDGMENT

This research is supported in part by the Singapore-MIT Alliance.

## REFERENCES

- [1] Y. Chen, K. H. Lim, R. H. Katz, and C. Overton. On the stability of network distance estimation. In *ACM SIGMETRICS Performance Evaluation Review (PER)*, September 2002.
- [2] Y. Chu, S. Rao, and H. Zhang. A case for end system multicast. In *ACM Sigmetrics*, June 2000.
- [3] M. Costa, M. Castro, A. Rowstron, and P. Key. PIC: Practical internet coordinates for distance estimation. In *ICDCS'04*, March 2004.
- [4] R. Cox, F. Dabek, F. Kaashoek, J. Li, and R. Morris. Practical, distributed network coordinates. In *HotNets-II*, November 2003.
- [5] F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi: A decentralized network coordinate system. In *Sigcomm'04*, August 2004.
- [6] P. Francis, S. Jamin, C. Jin, Y. Jin, V. Paxson, D. Raz, Y. Shavitt, and L. Zhang. IDMaps: A global internet host distance estimation service. In *IEEE Infocom'99*, New York, NY, March 1999.
- [7] T. Hansen, J. Otero, T. Mcgregor, and H.-W. Braun. Active measurement data analysis techniques. <http://amp.nlanr.net/>, 2002.
- [8] S. Hotz. *Routing Information Organization to Support Scalable Inter-domain Routing with Heterogeneous Path Requirements*. PhD thesis, University of Southern California, 1994.
- [9] L. Lehman and S. Lerman. Predicting internet network distances using peer-to-peer measurements. Internal technical report, appeared in Annual Singapore-MIT Alliance Symposium, January 2004.
- [10] H. Lim, J. Hou, and C.-H. Choi. Constructing internet coordinate system based on delay measurement. In *Internet Measurement Conference(IMC'03)*, October 2003.
- [11] T. Ng and H. Zhang. A network positioning system for the internet. In *USENIX Conference*, June 2004.
- [12] T. E. Ng and H. Zhang. Predicting internet network distance with coordinates-based approaches. In *INFOCOM*, 2002.
- [13] V. N. Padmanabhan and L. Subramanian. An investigation of geographic mapping techniques for internet hosts. In *ACM SIGCOMM'01*, San Diego, CA, August 2001.
- [14] K. P. Gummadi, S. Saroiu, and S. D. Gribble. King: Estimating latency between arbitrary internet end hosts. In *ACM SIGCOMM Internet Measurement Workshop(IMW'02)*, November 2002.
- [15] M. Pias, J. Crowcroft, S. Wilbur, T. Harris, and S. Bhatti. Lighthouses for scalable distributed location. In *2nd International Workshop on Peer-to-Peer Systems (IPTPS'03)*, Berkeley, CA, February 2003.
- [16] S. Ratnasamy, P. Francis, M. Handley, and R. Karp. A scalable content-addressable network. In *SIGCOMM'01*, San Diego, CA, 2001.
- [17] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker. Topologically-aware overlay construction and server selection. In *INFOCOM'02*, New York, 2002.
- [18] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *International Conference on Distributed Systems Platforms*, November 2001.
- [19] Y. Shavitt and T. Tanel. Big-bang simulation for embedding network distances in euclidean space. In *IEEE Infocom'03*, April 2003.
- [20] Y. Shavitt and T. Tanel. On the curvature of the internet and its usage for overlay construction and distance estimation. In *IEEE Infocom'04*, April 2004.
- [21] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *SIGCOMM'01*, 2001.
- [22] L. Tang and M. Crovella. Virtual landmarks for the internet. In *Internet Measurement Conference(IMC'03)*, October 2003.
- [23] M. Waldvogel and R. Rinaldi. Efficient topology-aware overlay network. In *Hotnets-I*, 2002.
- [24] B. Zhao, J. D. Kubiatowicz, and A. D. Joseph. Tapestry: An infrastructure for fault-resilient wide-area location and routing. Technical report, UCB/CSD, 2001.