Constructing Internet Coordinate System Based on Delay Measurement

Hyuk Lim, Jennifer C. Hou, and Chong-Ho Choi

Abstract

In this paper, we consider the problem of how to represent the network distances between Internet hosts in a Cartesian coordinate system to facilitate estimate of network distances among arbitrary Internet hosts. We envision an infrastructure that consists of beacon nodes and provides the service of estimating network distance between pairs of hosts without direct delay measurement. We show that the principal component analysis (PCA) technique can effectively extract topological information from delay measurements between beacon hosts. Based on PCA, we devise a transformation method that projects the raw distance space into a new coordinate system of (much) smaller dimensions. The transformation retains as much topological information as possible and yet enables end hosts to determine their coordinates in the coordinate system. The resulting new coordinate system is termed as the *Internet Coordinate System (ICS)*. As compared to existing work (e.g., IDMaps and GNP), ICS incurs smaller computation overhead in calculating the coordinates of hosts and smaller measurement overhead (required for end hosts to measure their distances to beacon hosts). Finally, we show via experimentation with both real-life and synthetic data sets that ICS makes robust and accurate estimates of network distances, incurs little computational overhead, and its performance is not susceptible to the number of beacon nodes (as long as it exceeds certain threshold) and the network topology.

Index Terms

Internet, modeling techniques, network topology, measurement techniques

A preliminary version of this paper appeared in ACM Internet Measurement Conference, Miami Beach, Florida, 2003.

The work reported in this paper has been funded in part by NSF under Grant Number CNS-0305537, by DARPA under a subcontract to U.C. Berkeley under Subaward No. SA3158-25622, and by the Brain Korea 21 Information Technology Program.

H. Lim and J. C. Hou are with the Department of Computer Science, University of Illinois at Urbana-Champaign, IL 61801, USA. E-mail: {hyuklim, jhou}@cs.uiuc.edu.

C.-H. Choi is with the School of Electrical Engineering and Computer Science, Seoul National University, Seoul 151-744, KOREA. E-mail: chchoi@csl.snu.ac.kr.

I. INTRODUCTION

Discovery of the Internet topology has many advantages for design and deployment of topology sensitive network services and applications, such as nearby server selection, overlay network construction, routing path construction, and peer-to-peer computing. The knowledge of network topology enables each host to make better decisions by exploiting its topological relations with other hosts. For example, in peer-to-peer file sharing services such as *Napster*, *Gnutella*, and *eDonkey*, a client can download shared files from a peer that is closer to itself, if the topology information is available. Among several categories of approaches to infer network topology, the measurement based approach may be the most promising, whereby the network topology can be constructed based on several network properties, such as bandwidth, round-trip time, and packet loss rate. In this paper, we focus on topology construction based on end-to-end delay (round-trip time) measurement, and use the term ”network distance” for the round-trip time between two hosts.

The primary goal of constructing network topology is to enable estimation of the network distance between arbitrary hosts without direct measurement between these hosts. Several approaches have been proposed, among which IDMaps [2] and GNP [3] may have received the most attention. Both assume a common architecture that consists of a small number of well-positioned infrastructure nodes (called *beacon nodes* in this paper). Every beacon node measures its distances to all the other beacon nodes and uses these measurement results to infer the network topology. A host estimates its distance to the other ordinary hosts by measuring its distances to beacon nodes (rather than to the other hosts). A host benefits from using this architecture, as it needs only to perform a small number of measurements and will be able to infer its network distance to a large number of hosts (such as servers).

One important issue in realizing these measurement architectures is how to represent the location of a host. IDMaps and Hotz’s triangulation [4], [5], for example, use the original distances to beacon nodes to represent the location of a host, while GNP [3] and Lighthouse [7] transform the original distance data space into a Cartesian coordinate system and uses coordinates in the coordinate system to represent the location. As will be discussed in Section III, the major advantage of representing network distances in a coordinate system is that it enables extraction of topological information from the measured network distance data. As a result, the accuracy

in estimating the distance between two arbitrary hosts will be improved. This is especially true when the number of available beacon nodes is small. To construct a new coordinate system, GNP formulates an optimization problem that minimizes the discrepancy between the measured network distance and the distance computed by a distance function in a coordinate system, and applies the Simplex Downhill method to solve the minimization problem. In spite of its many advantages, as will be elaborated on in Section III, GNP does not guarantee that a host has a unique coordinate in a coordinate system. Depending on the initial value used in the Simplex Downhill method, a single host may have different coordinates.

In this paper, we present a new coordinate system called the *Internet Coordinate System (ICS)*. The distances from a host to beacon nodes are expressed as a distance vector, where the dimension of the distance vector is equal to the number of beacon nodes. As each beacon node defines an axis in the distance data space, the bases may be correlated. We apply the principal component analysis (PCA) to projects the distance data space into a new, uncorrelated and orthogonal Cartesian coordinate system of (much) smaller dimensions. The linear transformation essentially extracts topology information from delay measurements between beacon nodes and retains it in a new coordinate system. By taking the first several principal components (obtained in PCA) as the bases, we can construct the Cartesian coordinate system of smaller dimensions while retaining as much topology information as possible.

Based on the PCA-derived Cartesian coordinate system, we then propose a method to estimate the network distance between arbitrary hosts on the Internet. The network distances between beacon nodes are first analyzed to retrieve the principal components. The first several components are scaled by a factor (such that the Euclidean distances in the new coordinate system approximate the measured distances) and used as the new bases in the coordinate system. The coordinate of a host is then determined by multiplying its original distance vector to (a subset of) beacon nodes with the linear transformation matrix consisting of the principal components. As compared to GNP, ICS is more computationally efficient because it only requires linear algebra operations. In addition, the location of a host is uniquely determined in the coordinate system. Another advantage of ICS is that it incurs smaller measurement overhead, as a host does not have to make delay measurement to *all* the beacon nodes, but only to a subset of beacon nodes. This is especially desirable in the case that some of the beacon nodes are not available (due to transient network partition and/or node failure). Finally, we show via Internet experimentation with real-

life data sets that ICS is robust and accurate, regardless of the number of beacon nodes (as long as it exceeds certain threshold) and the complexity of network topology.

The rest of the paper is organized as follows. In Section II, we provide the background material and define a distance coordinate system using linear algebra. In Section III, we give a summary of related work in the literature and motivate the need for a new coordinate system. In Sections IV–V, we first introduce PCA and then elaborate on ICS. Following that, we present in Section VI experimental results, and conclude the paper in Section VII.

II. PRELIMINARY

The topology of the Internet can be modeled in a coordinate system based on the delay measured between hosts. First we consider a *raw distance space*. Each host measures the network distance (i.e., the round trip delay) to the other hosts using *ping* or *traceroute*. Under the assumption that there exist m hosts, the coordinate of a host $\mathcal{H}_i$ in an m-dimensional system can be represented by the distance vector:

$$\mathbf{d}_i = [d_{i1}, \ldots, d_{im}]^T , \tag{1}$$

where d_{ij} is the network distance measured by the i^{th} host to the j^{th} host and $d_{ii} = 0$. In general, $d_{ij} \neq d_{ji}$ because the forward and reverse paths may have different characteristics. The overall system is represented by an m-by-m distance matrix $\mathbf{D}$, whose i^{th} column is the coordinate of host $\mathcal{H}_i$:

$$\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_m] . \tag{2}$$

Here $\mathbf{D}$ is a non-symmetric square matrix with zero diagonal entries. This representation is quite simple and intuitive, but contains too much redundant information as every host defines its own dimension in the coordinate system.

To reduce the redundancy of the above representation, we then represent the network distances between hosts in a *geometric coordinate system*. In this paper, we will study how to construct a coordinate system of the least possible dimension, while retaining as much topological information as possible. Under the assumption that a host $\mathcal{H}_i$ has the coordinate $\mathbf{x}_i$ in a coordinate system, the network distance d_{ij} from the host $\mathcal{H}_i$ to a host $\mathcal{H}_j$ can be estimated without direct measurement by computing a distance metric function f^d, (i.e., $d_{ij} \approx \tilde{d}_{ij} = f^d(\mathbf{x}_i, \mathbf{x}_j)$). The

generalized distance metric function [8] is defined as

$$L_p(\mathbf{x}_i, \mathbf{x}_j) = \left(\sum_{k=1}^{m} |x_{ik} - x_{jk}|^p \right)^{\frac{1}{p}}. \quad (3)$$

Some of the most important metrics are the Manhattan distance L_1, the Euclidean distance L_2, and the Chebyshev distance L_∞. In particular, it has been shown that L_∞ can be expressed as

$$L_\infty(\mathbf{x}_i, \mathbf{x}_j) = \lim_{p \to \infty} L_p(\mathbf{x}_i, \mathbf{x}_j) = \max_k |x_{ik} - x_{jk}|.$$

Note that for a coordinate based approach, violation to the triangle inequality of network distance measurements may degrade the performance of the distance estimation. Fortunately it has been shown in [6] that violation to the triangle inequality violations is not particularly frequent through various measurement data sets.

III. Related work

A. Methods in the distance data space

Several methods have been proposed to estimate the network distance between hosts on the Internet. These methods envision an infrastructure in which servers (beacon nodes) measure network distances between one another, and a client $\mathcal{H}_a$ (ordinary host) infers its distance to some other host $\mathcal{H}_b$ based on the distance information between servers. Hotz defined, for a host $\mathcal{H}_a$, a distance vector $\mathbf{d}_a = [d_{a1}, \ldots, d_{am}]^T$ [4], where d_{ai} is the measured distance to the i^{th} beacon node for $i \in \{1, \ldots, m\}$ and m is the number of beacon nodes. Then, the network distance d_{ab} between hosts $\mathcal{H}_a$ and $\mathcal{H}_b$ was shown to be bound by:

$$\max_i |d_{ai} - d_{bi}| \leq d_{ab} \leq \min_i (d_{ai} + d_{bi}). \quad (4)$$

Note that the lower bound is the Chebyshev distance between the two vectors, $\mathbf{d}_a$ and $\mathbf{d}_b$. Hotz also showed that the average of the upper and lower bounds generally gives a better estimate of the distance than either bound. Guyton *et al.* later applied Hotz's triangulation method to calculate the distances to various servers and to locate nearby ones on the Internet [5].

A global architecture for estimating Internet host distances, called *the Internet Distance Map Service, IDMaps*, was first proposed by Francis *et al.* [2]. The architecture separates beacon nodes (called *tracers*) that collect and distribute distance information from clients that use the distance map. Each tracer measures the distances to IP address prefixes (APs) that are close to itself.

A client first determines its own AP and the autonomous system (AS) the AP is connected to. The client then runs a spanning-tree algorithm over the distance information gathered by tracers to find the shortest distance between its AS and the AS that the AP of the destination belongs to. This distance is taken as the estimated distance. Methods of this type (i.e., methods that represent network distances in a data space) neither analyze delay measurements nor infer network topology. Consequently, their performance depends heavily on the number and placement of beacon nodes. If the number of beacon nodes is small, the estimation performance may not be good.

In order to extract topological information, Ratnasamy *et al.* [9] proposed a binning scheme. A bin is defined as the list of beacon nodes in the order of increasing delay. The bin of a host indicates the relative distances to all the beacon nodes. For example, if the bin of a host is "$\mathcal{H}_a\mathcal{H}_c\mathcal{H}_b$", beacon node $\mathcal{H}_a$ is the closest to the host, and $\mathcal{H}_b$ is the farthest to the host. The authors applied the binning scheme to the problems of constructing overlay networks and selecting servers. A host joins an overlay network node or selects a server whose bin is most similar to its own bin.

B. Methods in the geometric coordinate system

Ng *et al.* proposed a coordinate-based approach, called *Global Networking Positioning (GNP)* [3]. Instead of using the raw network distances, GNP represents the location of each host in a geometric space, in which the distance between two hosts is defined as a distance function f^d. The major advantage of representing network distances in a coordinate system is its capability to extract topological information from the measured network distances. As a result, the accuracy in estimating the distance between two arbitrary hosts will be improved especially in the case that the number of beacon nodes is small.

Two optimization problems have been considered in GNP in order to obtain the coordinates of beacon nodes and hosts in the coordinate system. The first problem obtains the coordinates of beacon nodes in GNP by minimizing the following objective function:

$$J_1 = \sum_{i,j} \mathcal{E}\left(d_{ij}, f^d(\mathbf{x}_i, \mathbf{x}_j)\right), \tag{5}$$

where $\mathcal{E}$ is an error function (e.g., square error), d_{ij} is the measured distance between the i^{th} and j^{th} beacon nodes, and $\mathbf{x}_i$ is the coordinate of the i^{th} beacon node in the coordinate system. The

second optimization problem determines the coordinate of an ordinary host $\mathcal{H}_h$ by minimizing the following cost function:

$$J_2 = \sum_i \mathcal{E}\left(d_{hi}, f^d(\mathbf{x}_i, \mathbf{x}_h)\right), \tag{6}$$

where d_{hi} is the measured distance between host $\mathcal{H}$ and the i^{th} beacon nodes, and $\mathbf{x}_h$ is the coordinate of the host $\mathcal{H}$. GNP tackles both optimization problems using the Simplex Downhill method [10]. Unfortunately, the Simplex Downhill method only gives a local minimum that is close to the starting value and does not guarantee that the result is unique in the case that the cost functions are not (strictly) convex. (The cost functions expressed in Eqs. (5) and (6) are not strictly convex.) It is stated in [3] that the first optimization problem may have an infinite number of solutions, and any solution is sufficient. If the solution to the first optimization problem is a good approximation of a global minimum, the coordinates of beacon nodes thus calculated suffice in the first problem. However, this is not the case in the second optimization problem. A host in GNP may have different coordinates depending on the starting values used in the Simplex Downhill method. The fact that ordinary hosts may have non-unique coordinates may lead to estimation inaccuracy. We demonstrate the problem in the following example.

Example 1: (Problem with GNP) Consider four hosts, two of which are located in one autonomous system (AS), and the other two in another AS. Also assume (for demonstration purpose) that the distance between two hosts in the same AS is 1 while the distance between two hosts in different ASs is 3. Then the topology can be expressed using the following distance matrix $\mathbf{D}$:

$$\mathbf{D} = \begin{bmatrix} 0 & 1 & 3 & 3 \\ 1 & 0 & 3 & 3 \\ 3 & 3 & 0 & 1 \\ 3 & 3 & 1 & 0 \end{bmatrix}.$$

In the Euclidean space model of GNP, the first cost function J_1 in two-dimensional coordinate system can be written as

$$J_1 = \sum_{(i,j)=(1,2),(3,4)} \left(1 - \sqrt{\sum_{k=1}^{2}(x_{ik} - x_{jk})^2}\right)^2 + \sum_{(i,j)=(1,3),(1,4),(2,3),(2,4)} \left(3 - \sqrt{\sum_{k=1}^{2}(x_{ik} - x_{jk})^2}\right)^2.$$

We solve the optimization problem using the 'fminsearch' function in Matlab, which implements the Simplex Downhill method, with the starting values, $\mathbf{x}_1^s = [0,0]^T$, $\mathbf{x}_2^s = [1,1]^T$, $\mathbf{x}_3^s =$

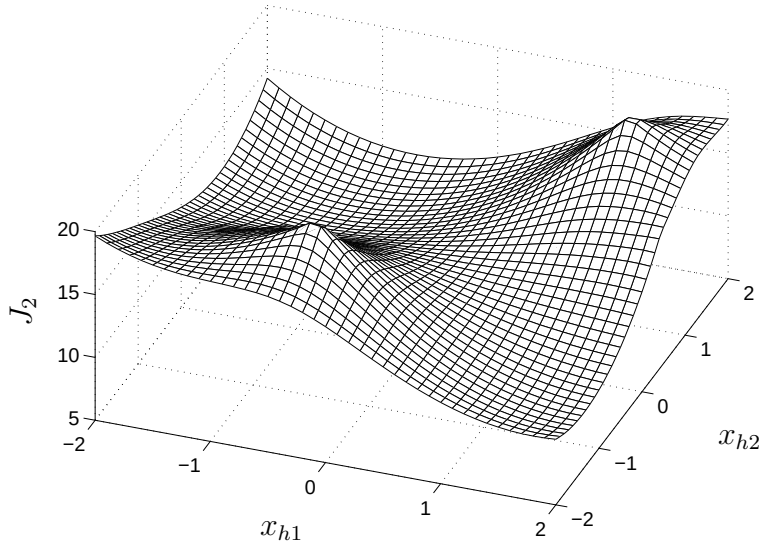


Fig. 1. The cost function for the coordinate of an ordinary host in Example 1

.

$[-1,-1]^T$, and $\mathbf{x}_4^s = [0,0]^T$. The coordinates of the beacon nodes calculated with this set of starting values are $\mathbf{x}_1 = [0.4433, 2.0048]^T$, $\mathbf{x}_2 = [1.2262, 1.4248]^T$, $\mathbf{x}_3 = [-0.5137, -0.9240]^T$, and $\mathbf{x}_4 = [-1.2966, -0.3440]^T$. Note that $L_2(\mathbf{x}_1, \mathbf{x}_2) = 0.9743 \approx 1$, $L_2(\mathbf{x}_1, \mathbf{x}_3) = 3.0812 \approx 3$ and so on.

Now assume that a host $\mathcal{H}$ measures its distances to four beacon nodes, and obtains a distance vector $\mathbf{d}_h = [1,4,1,4]^T$. The cost function J_2 in the second optimization problem (Eq. (6)) becomes

$$J_2 = \sum_{i=1,3} \left(1 - \sqrt{\sum_{k=1}^{2} (x_{ik} - x_{hk})^2} \right)^2 + \sum_{i=2,4} \left(4 - \sqrt{\sum_{k=1}^{2} (x_{ik} - x_{hk})^2} \right)^2 .$$

Figure 1 depicts the cost function J_2 with respect to x_{h1} and x_{h2}. The cost function has two local minima at $(1.2866, -0.9130)$ and $(-1.3571, 1.9938)$. Therefore, $\mathbf{x}_h$ can be either $[1.2866, -0.9130]^T$ or $[-1.3571, 1.9938]^T$ depending on the starting values of the Simplex Downhill method. If the starting value is (1,-1), the Simplex Downhill method renders the former local minimum $(1.2866, -0.9130)$. This implies that GNP does not guarantee a unique mapping from the raw distance vector to the Cartesian coordinate. □

Our proposed approach, ICS, shares the similarity with GNP in that it also represents locations

of hosts in a Cartesian coordinate system instead of a raw distance space, and consequently, can extract topological information from measured network distances. ICS, however, provides a unique mapping from the distance space to the Cartesian coordinate system (and thus yields a more accurate representation). In addition, it has the following advantages:

- With the use of principal component analysis (PCA), a host can calculate its coordinates by means of basic linear algebra (e.g., the singular value decomposition and matrix multiplication). The computational overhead is reduced.
- Unlike all the other previous work, a host does not have to measure its distance to *all* the beacon nodes, but instead to a subset of beacon nodes. The measurement overhead is reduced.

It has come to our attention that Tang and Crovella [6] also applied principal component analysis to project distance measurements into a Cartesian coordinate system with smaller dimensions. The authors considered the coordinate of a host in the coordinate system as the distances to *virtual landmarks* while the coordinate in the distance data space represents the distances to actual beacon nodes (landmarks). For the sake of scalability, the authors also devised a coordinate exchanging method among multiple coordinate systems.

Another technique that embeds the Internet graph into a vector space is *lighthouse* [7]. Similarly, lighthouse uses a linear transformation to compute the coordinates of hosts. However, unlike ICS and virtual landmarks, lighthouse applies the Gram-Schmidt process to compute an orthogonal basis based on the intra-lighthouses distances. This is achieved through the QR decomposition as opposed to singular value decomposition (SVD, used in principal component analysis). The key advantage of lighthouse is that a host has flexibility in choosing its set of landmarks (termed as lighthouses) in a distributed manner.

IV. Principal Component Analysis (PCA)

We now discuss how to extract topological information from the distance matrix $\mathbf{D}$ (Eq. (2)). In Example 1, the dimension of the distance matrix $\mathbf{D}$ is four. As hosts in the same AS are very close to each other, the distance can be represented in a two-dimensional space by projecting their coordinates into two-dimensional space. The dimensionality depends not on the dimension m of the distance matrix $\mathbf{D}$ but on the network topology, and can be much smaller than m.

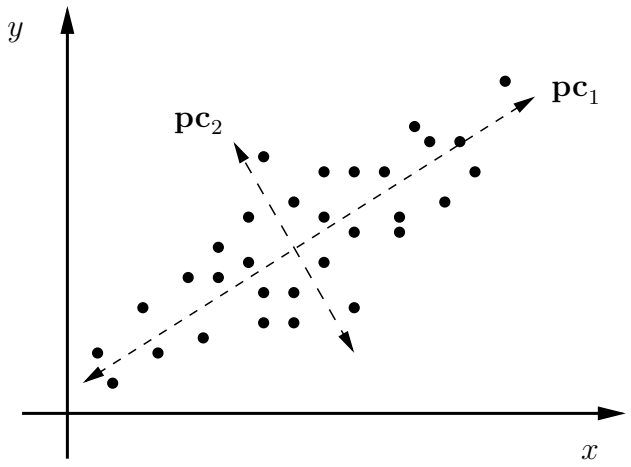


Fig. 2. Example of the principal component analysis

We apply principal component analysis (PCA) [11], [13], [14] to reduce the dimension of the distance matrix while retaining as much topological information as possible. In a nutshell, PCA transforms a data set that consists of a large number of (possibly) correlated variables to a new set of uncorrelated variables, *principal components*, which can characterize the network topology. The principal components are ordered so that the first several components have the most important features of the original variables. In particular, the k^{th} principal component can be interpreted as the direction of maximizing the variation of projections of measured distance data while orthogonal to the first $(k-1)^{th}$ principal components [13]. We use the following example to illustrate the concept.

Example 2: Figure 2 gives an example of performing PCA for two correlated variables, x and y. With the use of PCA, we obtain two principal components, $\mathbf{pc}_1$ and $\mathbf{pc}_2$. As shown in Fig. 2, the first principal component $\mathbf{pc}_1$ represents the direction of the maximum variance. The one-dimensional linear representation is calculated by projecting the original data onto $\mathbf{pc}_1$. □

Now the question is how to determine these principal components. The most common approach is to use singular value decomposition (SVD). Specifically, the SVD of $\mathbf{D}$ in Eq. (2) is obtained

by

$$\mathbf{D} = \mathbf{U} \cdot \mathbf{W} \cdot \mathbf{V}^T, \tag{7}$$

$$\mathbf{W} = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_m \end{bmatrix},$$

where $\mathbf{U}$ and $\mathbf{V}$ are column and row orthogonal matrices, and σ_i's are the singular values of $\mathbf{D}$ in the decreasing order (i.e., $\sigma_i \geq \sigma_j$ if $i < j$). Note that $\mathbf{D}^T\mathbf{D} = (\mathbf{U}\mathbf{W}\mathbf{V}^T)^T(\mathbf{U}\mathbf{W}\mathbf{V}^T) = \mathbf{V}(\mathbf{W}^T\mathbf{W})\mathbf{V}^T$. This means that the eigenvectors of $\mathbf{D}^T\mathbf{D}$ make up $\mathbf{V}$ with the associated (real nonnegative) eigenvalues of the diagonal of $\mathbf{W}^T\mathbf{W}$ [12]. Similarly, $\mathbf{D}\mathbf{D}^T = \mathbf{U}^T(\mathbf{W}\mathbf{W}^T)\mathbf{U}$. The columns of the $m \times m$ matrix $\mathbf{U} = [\ \mathbf{u}_1, \ldots, \mathbf{u}_m\ ]$ are the principal components and the orthogonal basis of the new subspace. By using the first n columns of $\mathbf{U}$ denoted by $\mathbf{U}_n$, we project the m-dimensional space into a new n-dimensional space:

$$\mathbf{c}_i = \mathbf{U}_n^T \cdot \mathbf{d}_i = [\ \mathbf{u}_1,\ \ldots,\ \mathbf{u}_n\ ]^T \cdot \mathbf{d}_i. \tag{8}$$

We re-visit Example 1 to illustrate the procedure.

Example 3: (Example revisited) Consider the four hosts and the corresponding distance matrix in Example 1. We obtain the principal components via singular value decomposition (Eq. (7)):

$$\mathbf{U} = \begin{bmatrix} -\frac{1}{2} & -\frac{1}{2} & \frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{2} & \frac{1}{2} & 0 & -\frac{1}{\sqrt{2}} \\ -\frac{1}{2} & \frac{1}{2} & 0 & \frac{1}{\sqrt{2}} \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} 7 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The original distance vector of the first host is $\mathbf{d}_1 = [0, 1, 3, 3]^T$. With the use of Eq. (8), we can calculate the coordinate of the first host in a two-dimensional coordinate system as

$$\mathbf{c}_1 = \mathbf{U}_2^T\mathbf{d}_1 = \begin{bmatrix} -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \\ 3 \\ 3 \end{bmatrix} = \begin{bmatrix} -\frac{7}{2} \\ \frac{5}{2} \end{bmatrix}.$$

Similarly $\mathbf{c}_2 (= \mathbf{c}_1) = [-\frac{7}{2}, \frac{5}{2}]^T$ and $\mathbf{c}_3 = \mathbf{c}_4 = [-\frac{7}{2}, -\frac{5}{2}\ ]^T$. Note that PCA assigns the same coordinate to the two hosts in the same AS because of the low dimensionality. When $n = 4$, $\mathbf{U}_4 =$

TABLE I

AVERAGE PROXIMITY IN ORIGINAL GEOMETRY SPACE D

Metric	NPD (m = 33)	NLANR (m = 113)
L_1	5.818	6.964
L_2	6.545	6.495
L_∞	12.151	5.504

$\mathbf{U}$, $\mathbf{c}_1 = [-\frac{7}{2}, \frac{5}{2}, -\frac{\sqrt{2}}{2}, 0]$, $\mathbf{c}_2 = [-\frac{7}{2}, \frac{5}{2}, \frac{\sqrt{2}}{2}, 0]$, $\mathbf{c}_3 = [-\frac{7}{2}, -\frac{5}{2}, 0, \frac{\sqrt{2}}{2}]$, and $\mathbf{c}_4 = [-\frac{7}{2}, -\frac{5}{2}, 0, -\frac{\sqrt{2}}{2}]$. In this case ($n$=$m$), the mapping $\mathbf{c}_i = \mathbf{U}^T \cdot \mathbf{d}_i$ is isometric (e.g., $L_2(\mathbf{d}_1, \mathbf{d}_3) = L_2(\mathbf{c}_1, \mathbf{c}_3) = 5.0990$), and thus the two spaces spanned by $\mathbf{d}_i$'s and $\mathbf{c}_i$'s are the same from the perspective of geometry (i.e., $L_2(\mathbf{d}_i, \mathbf{d}_j) = L_2(\mathbf{c}_i, \mathbf{c}_j)$).

A. Dimensionality

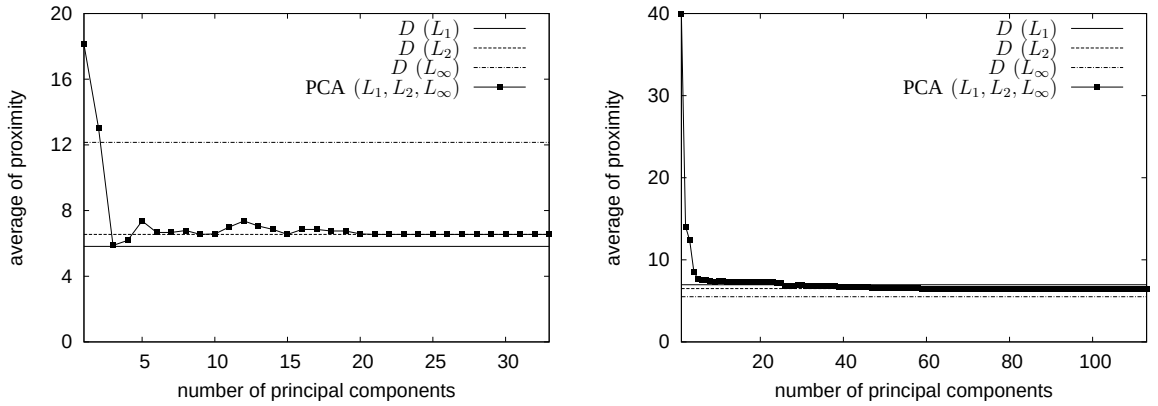
Another important issue that should be addressed in representing network distances in a n-dimensional coordinate system is how to determine the adequate degree, n, of dimensions in the coordinate system. This problem has not been extensively studied, and is usually application-dependent [15]. One of the commonly adopted criteria is the cumulative percentage of variation that selected principal components contribute to [11]. The percentage, t_k, of variation accounted for by the first k principal components is defined by

$$t_k = 100 \times \frac{\sum_{j=1}^{k} \sigma_j}{\sum_{j=1}^{m} \sigma_j}. \tag{9}$$

One may pre-determine a cut-off value, t^* of cumulative percentage of variation, and calculate n to be the smallest integer such that $t_n \geq t^*$. In the previous example, t_1 = 50 %, t_2 = 85.7 %, t_3 = 92.9 %, and t_4 = 100 %. If t^* is set to 80 %, then the degree of dimensions should be set to n = 2.

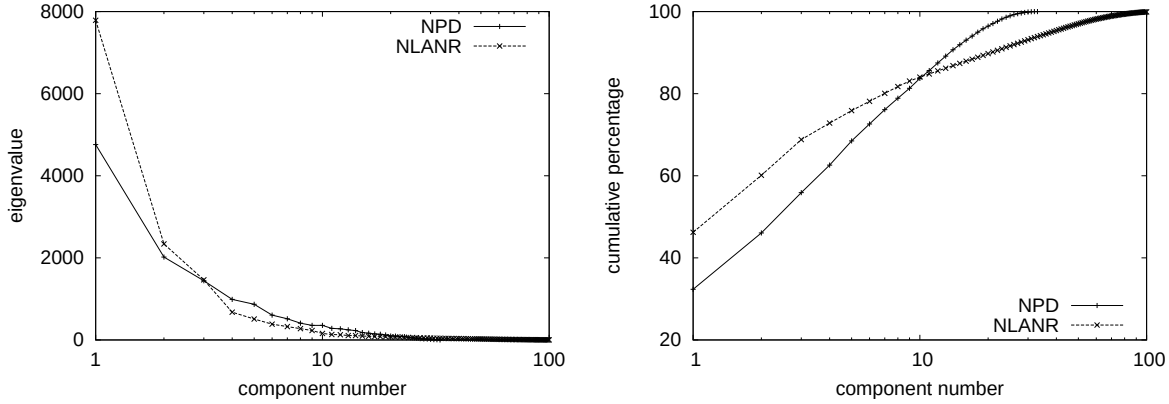
B. Experimental Results

To investigate whether or not PCA can be used to transform network distances on the Internet to coordinates in a coordinate system of smaller dimensions and still retain as much topological information as possible, we apply PCA to two real-life data sets:



(a) (b)

Fig. 3. Average proximity for the NPD data set ((a)) and the NLANR data set ((b)) under different distance metrics.



(a) (b)

Fig. 4. Eigenvalues and cumulative percentage of variation for the NPD data set and the NLANR data set.

- NPD-Routes-2 data set [16]: contains Internet route measurements obtained by *traceroute*. The measurements were made between 33 Internet hosts in the Network Probe Daemon (NPD) framework from November 3, 1995, to December 21, 1995. We obtain the distance matrix $\mathbf{D}$ in Eq. (2) by taking (for each pair of hosts) the minimum value of measured round trip times (RTTs) in order to filter out the queuing delay.
- NLANR: contains the RTT, packet loss, topology, and on-demand throughput measurements made under the Active Measurement Project (AMP) at National Laboratory for Applied Network Research (NLANR). More than 100 AMP monitors are used to make the measurements [17]. The round trip times between all the monitors are measured every minute,

and are processed once a day. We use one of the NLANR RTT data sets measured between 113 AMP monitors on April 9, 2003.

We first compare different distance metrics with respect to their quality of representing topological information. Given that the number of hosts in the data set is m, each host has an m dimensional distance vector as its coordinate in the raw distance space, and an n dimensional distance vector in the coordinate system obtained by PCA ($1 \leq n \leq m$). We calculate for each host the distances L_1, L_2, and L_∞ (Eq. (3)) to all the other hosts, and determine its closest host based on the distance calculated in the coordinate system. As the "closest" host calculated under the various distance metrics may not be the actual closest host, we define the notion of *proximity* to measure the quality of representing topological information. If the host calculated to be the closest is the k^{th} closest, the proximity is set to k, $k \geq 1$. We average, for each distance metric used, the proximity over all the hosts.

Table I gives the average proximity in the raw distance space, whose dimension is m = 33 and 113 for the NPD and NLANR data sets, respectively. In the NPD data set, L_1 gives the best performance — the host calculated to be the closest is the 5.818^{th} closest host averagely. In the NLANR data set, L_∞ gives the best performance. These results show that the accuracy of representing topological information in a raw distance space depends heavily on the distance metric.

Next we study the (in)effectiveness of using PCA to represent network distances. Figure 3 gives the average proximity with respect to the number of principal components for the NPD and NLANR data sets. As shown in Fig. 3 (a), when the number of principal components is greater than 3, the proximity is almost the same as that in the raw distance data space. This means that the topological information can be effectively represented in a 3-dimensional space instead of in a 33-dimensional space. Another important observation is that the average proximity in the new coordinate system of smaller dimensions remains the same regardless of the distance metric used. The reason why the proximity is independent of the distance metric used is due to the fact that PCA finds a set of uncorrelated bases to represent the topological information. A similar trend can be observed in Fig. 3 (b) in which the proximity is almost the same as that in the distance space when the number of principal components is larger than 10.

Figure 4 plots the eigenvalues and their corresponding cumulative percentage of variation. The largest eigenvalues are 4760.0 and 7787.3, respectively, for the NPD and NLANR data sets. If

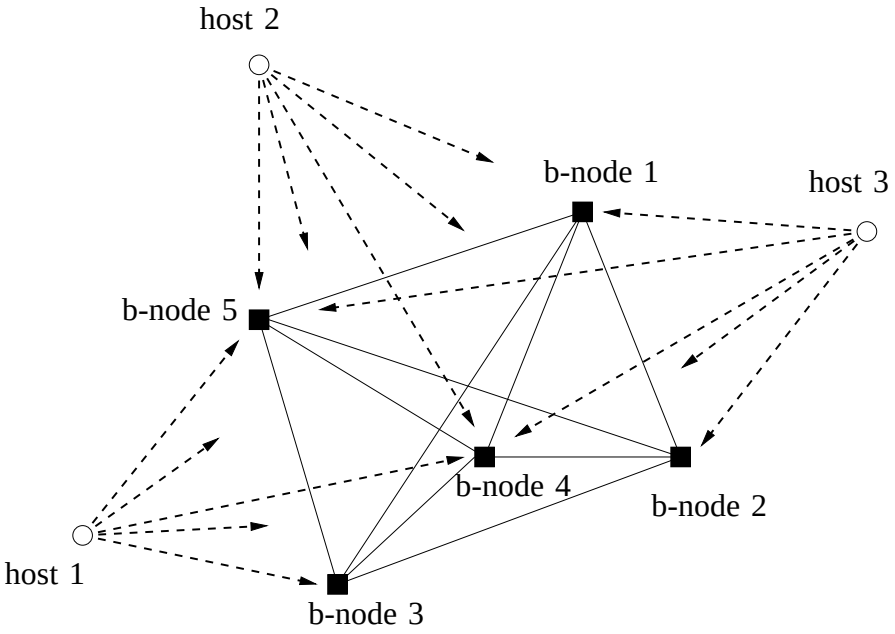


Fig. 5. An example architecture for the proposed Internet Coordinate System (five beacon nodes and three ordinary hosts).

we set a cut-off threshold of $t^* = 80$ %, the smallest value of n that achieves the threshold for each data set is, respectively, 9 and 7. In this case, σ_9 = 354.7, and the average proximity is 6.54 for the NPD data set, and σ_7 = 325.2 and the average proximity is 7.49 for the NLANR data set.

In summary, we show in this section that the network distance on the Internet can be represented, with the use of PCA, in a Cartesian space that uses a (smaller) set of uncorrelated bases. Moreover, we show that the new coordinate system is less susceptible to the distance metrics used in representing topological information.

V. Internet Coordinate System

A. Overview

We first present a basic architecture for the Internet Coordinate System (ICS). As mentioned in Section I, the objectives of ICS are i) to infer the network topology based on delay measurement and ii) to estimate the distance between hosts without direct measurement. Succinctly, the architecture for ICS consists of a number of beacon nodes that collect and analyze the distance information. Figure 5 gives an example architecture of ICS with five beacon nodes. Beacon nodes

periodically measure round trip times (RTTs) to other beacon nodes and construct a coordinate system. The coordinates of beacon nodes are then calculated, with the use of PCA, based on the measured RTT data among the five beacon nodes. We will elaborate on how to calculate the coordinates of beacon nodes in Section V-B.

An ordinary host determines its own location in ICS by measuring its delays to the entire or partial set of beacon nodes and obtains a distance vector. As exemplified in Fig. 5, host 1 measures its distance to five beacon nodes, and obtains a five-dimensional distance vector. The location of the host in ICS is then calculated by multiplying the distance vector with a transformation matrix. (We will elaborate on how the transformation matrix is derived and distributed in Section V-C.) After calculating its own coordinate, host 1 may report its coordinate to a DNS-like server that keeps coordinates of ordinary hosts. To estimate the network distance to some other host, host 1 may query this DNS-like server which then determines the estimated distance as long as the coordinate of the other host is kept at the server. In the same manner, host 1 can also infer which other host is closer to itself.

B. Calculating the Coordinates of Beacon Nodes

We now elaborate on how we construct ICS based on the measured network distances between m beacon nodes, and apply PCA (Section IV) to "transform" the raw distance space to a new coordinate system of (much) smaller dimensions.

Each beacon node measures its distances to the other beacon nodes, and obtains a m-dimensional distance vector $\mathbf{d}_i$ in Eq. (1), of which the j^{th} element d_{ij} is the measured distance to the j^{th} beacon node. An administrative node, which can be elected among beacon nodes, aggregates the distance vectors of all the beacon nodes, and obtains the distance matrix $\mathbf{D}$ in Eq. (2). Then, the distance matrix is decomposed into three matrices $\mathbf{U}$, $\mathbf{W}$, and $\mathbf{V}$ in Eq. (7). Using the first n principal components, the coordinate of a beacon node is calculated as $\mathbf{c}_i = \mathbf{U}_n\mathbf{d}_i$ in Eq. (8). As shown in Section IV-B, this coordinate preserves topological information.

Note that the distance between two beacon nodes calculated by Eq. (8) does not coincide with its actual measured distance. For instance, $L_2(\mathbf{c}_1, \mathbf{c}_3) = 5 \neq d_{13} = 3$ when $n = 2$ in Example 3. To use the coordinates for distance estimation, we apply a simple linear operation, $\bar{\mathbf{c}}_i = \alpha\mathbf{c}_i + \beta$, so as to minimize the discrepancy between the distance represented in the coordinate system and the measured distance. As a scaling operation does not affect the distance between two

coordinates, we only consider the scaling operation with a scaling factor α, i.e., $\beta = 0$. The optimal scaling factor $\alpha^*(n)$ that minimizes the discrepancy between the Euclidean distance in the new coordinate system of dimension n and the measured delay, i.e., $L_2(\bar{\mathbf{c}}_i, \bar{\mathbf{c}}_j) \approx d_{ij}$ for all i and $j \in \{1, \ldots, m\}$, can be determined by minimizing the following objective function $J(\alpha)$:

$$J(\alpha) = \sum_{i}^{m} \sum_{j}^{m} \left(L_2(\alpha \mathbf{c}_i, \alpha \mathbf{c}_j) - d_{ij} \right)^2. \tag{10}$$

After a few algebraic operations, the positive solution, α^*, can be shown to be

$$\alpha^*(n) = \frac{\sum_i^m \sum_j^m d_{ij} L_2(\mathbf{c}_i, \mathbf{c}_j)}{\sum_i^m \sum_j^m L_2(\mathbf{c}_i, \mathbf{c}_j)^2}. \tag{11}$$

The transformation matrix $\bar{\mathbf{U}}_n$ from a distance vector in the distance space to the coordinate in ICS is then defined as

$$\bar{\mathbf{U}}_n = \alpha^*(n) \mathbf{U}_n = \frac{\sum_i^m \sum_j^m d_{ij} l_{ij}}{\sum_i^m \sum_j^m l_{ij}^2} \mathbf{U}_n, \tag{12}$$

where $l_{ij} = L_2(\mathbf{U}_n^T \mathbf{d}_i, \mathbf{U}_n^T \mathbf{d}_j)$ and the transformation matrix $\mathbf{U}_n = [\mathbf{u}_1, \ldots, \mathbf{u}_n]$ is obtained from the distance matrix $\mathbf{D}$ between beacon nodes and its SVD. The coordinates of beacon nodes are then calculated as $\bar{\mathbf{c}}_i = \bar{\mathbf{U}}_n^T \mathbf{d}_i$ for all $i \in \{1, \ldots, m\}$.

In summary, the procedure taken to calculate the coordinates of beacon nodes is as follows:

(S1) Every beacon node measures the round trip times to the other beacon nodes periodically.

(S2) An administrative node aggregates the delay information and obtains the distance matrix $\mathbf{D}$ in Eq. (2).

(S3) The administrative node applies PCA in Eq. (7) to obtain the transformation matrix $\mathbf{U}$.

(S4) The administrative node determines the dimension of the coordinate system using the cumulative percentage of variation defined in Eq. (9) (with a pre-determined threshold value).

(S5) The administrative node calculates the transformation matrix $\overline{\mathbf{U}}_n$ in Eq. (12) from Eq. (7) and Eq. (11).

Note that the administrative node may be replicated (perhaps in a hierarchical manner) to enhance fault tolerance and availability. This subject is outside the scope of this paper, but is warrant of further investigation. We illustrate the above procedure by revisiting Example 1.

Example 4: Assume that the four hosts in Example 1 are beacon nodes. When n = 2, $\mathbf{c}_1 = \mathbf{c}_2 = [-3.5, 2.5]$ and $\mathbf{c}_3 = \mathbf{c}_4 = [-3.5, -2.5]^T$. By Eq. (11), the scaling factor α is 0.6, and the

transformation matrix $\bar{\mathbf{U}}_2$ is

$$\bar{\mathbf{U}}_2 = \begin{bmatrix} -0.3 & -0.3 & -0.3 & -0.3 \\ -0.3 & -0.3 & 0.3 & 0.3 \end{bmatrix}^T.$$

Therefore, $\bar{\mathbf{c}}_1 = \bar{\mathbf{c}}_2 = [-2.1, 1.5]$ and $\bar{\mathbf{c}}_3 = \bar{\mathbf{c}}_4 = [-2.1, -1.5]$. The distances between two hosts in different ASs is exactly 3. When n=4, α = 0.5927, $L_2(\bar{\mathbf{c}}_1, \bar{\mathbf{c}}_2) = L_2(\bar{\mathbf{c}}_3, \bar{\mathbf{c}}_4) = 0.8383$, and $L_2(\bar{\mathbf{c}}_1, \bar{\mathbf{c}}_3)$ = $L_2(\bar{\mathbf{c}}_1, \bar{\mathbf{c}}_4)$ = $L_2(\bar{\mathbf{c}}_2, \bar{\mathbf{c}}_3)$ = $L_2(\bar{\mathbf{c}}_2, \bar{\mathbf{c}}_4) = 3.0224$. □

C. Determining The Coordinate of A Host

The procedure that a host takes to determine its coordinate in ICS is as follows: A host $\mathcal{H}_a$

(H1) obtains the list of beacon nodes and the transformation matrix $\overline{\mathbf{U}}_n$ (Eq. (12)) from the administrative node.

(H2) measures the network distances, $\mathbf{l}_a = [l_{a1}, \ldots, l_{am}]^T$, to all the beacon nodes using *ping* or *traceroute*, where l_{ai} denotes the delay measured between host $\mathcal{H}_a$ and the i^{th} beacon node. (We will discuss how to reduce the number of measurements in Section V-D.)

(H3) calculates the coordinate, $\mathbf{x}_a$, by multiplying the measured distance vector with the transformation matrix, i.e., $\mathbf{x}_a = \bar{\mathbf{U}}_n^T \cdot \mathbf{l}_a$.

Example 5: Consider the ICS system in Example 4. Assume that host $\mathcal{H}_a$ is closer to the AS where the first two beacon nodes reside, and obtains a distance vector of $\mathbf{l}_a = [1, 1, 4, 4]^T$. In (H3), $\mathbf{x}_a = [-3, 1.8]^T$. In the case of $n = 2$, the estimated distances between host $\mathcal{H}_a$ and beacon nodes are $L_2(\bar{\mathbf{c}}_1, \mathbf{x}_a) = L_2(\bar{\mathbf{c}}_2, \mathbf{x}_a) = 0.94$ and $L_2(\bar{\mathbf{c}}_3, \mathbf{x}_a) = L_2(\bar{\mathbf{c}}_4, \mathbf{x}_a) = 3.42$. Assume that host $\mathcal{H}_b$ is far from all four beacon nodes, and obtains a distance vector of $\mathbf{l}_b = [10, 10, 10, 10]^T$. In this case, $\mathbf{x}_b = [-12, 0]^T$, and $L_2(\bar{\mathbf{c}}_i, \mathbf{x}_b) = 10.01$ for $i = 1, \ldots, 4$. □

D. Reducing The Number of Measurements

To discover accurately the topology of the Internet, a sufficient number of beacon nodes should be judiciously placed on the Internet. (Note that PCA is able to extract essential topological information from a set of (perhaps correlated) delay measurements. However, it does not preclude the important task of placing beacon nodes properly on the Internet so as to represent the network topology accurately. We will comment on this issue in Section V-E.) On the other hand, for the sake of scalability, it is not desirable that a client has to measure its round trip times to *all* the

beacon nodes. To reduce the measurement overhead incurred by a host, it would be desirable that a host measures the distance from itself to *only* a subset of beacon nodes. This also allows ICS to operate even in the case that some of the beacon nodes are not available (due to, for example, transient network partition and/or node failure).

In (H3), the transformation matrix $\bar{\mathbf{U}}_n^T$ and the original distance vector $\mathbf{l}_a$ are needed to calculate the coordinate of a host. The transformation matrix is fixed in ICS once it is calculated by the administrative node. If host $\mathcal{H}_a$ makes delay measurements only to a subset, $\mathcal{N}$, of beacon nodes, the missing elements in $\mathbf{l}_a$, i.e., l_{ai}, $i \notin \mathcal{N}$, have to be inferred.

The procedure for partial measurement is as follows: Host $\mathcal{H}_a$ randomly chooses k beacon nodes ($k < m$) and measures its distances to this subset, $\mathcal{N}$, of beacon nodes. (In our experiments, we will investigate the effect of the value of k on the estimation performance.) Instead of calculating the coordinate by itself, host $\mathcal{H}_a$ transmits the distance vector $\mathbf{l}_a$ with $m-k$ missing elements to the administrative node. For each missing element l_{ai} in $\mathbf{l}_a$, the administrative node (i) selects in $\mathcal{N}$ a beacon node (say the j^{th} beacon node) that is closest to the i^{th} beacon node, (ii) replaces the missing element l_{ai} with a function of l_{aj} (to be discussed below), and (iii) calculates the coordinate on the behalf of host $\mathcal{H}_a$.

The performance of the partial measurement method depends heavily on how well the missing elements in $\mathbf{l}_a$ are represented (step (ii)). In order to improve the performance, instead of directly using the network distance measured to the closest beacon node, we can leverage Hotz's triangulation method (Section III) as follows: As a beacon node $\mathcal{H}_b$ that is not in $\mathcal{N}$ has already measured its distances to other beacon nodes, the distance between host $\mathcal{H}_a$ and node $\mathcal{H}_b$ can be estimated using Hotz's triangulation method.

E. Enhancing ICS by Clustering

If beacon nodes are well distributed and selected with respect to certain clustering criterion, the performance is expected to be better [3] because the basis of the coordinate system is constructed based on the measurements between beacon nodes. There are essentially two aspects in which the notion of clustering can be applied in selecting beacon nodes. On the one hand, if the distances among hosts that are available to serve as beacon nodes can be measured, a clustering algorithm can be applied to group hosts that are close to one another into clusters [18]. Each host is initially assigned to its own cluster, and pairs of neighboring clusters are repeatedly merged into a single

cluster until k clusters remain. The median node in each cluster is selected as a beacon node. This approach serves as a guideline for placement of beacon nodes.

On the other hand, if the beacon nodes have been placed *a priori*, the clustering technique can be incorporated into the partial measurement method (Section V-D) as follows: Instead of randomly selecting of beacon nodes in Section V-D, the administrative node specifies, for a host $\mathcal{H}_a$, a list of beacon nodes to which host $\mathcal{H}_a$ should make delay measurements. The administrative node applies the clustering technique to form clusters among beacon nodes, and selects for each cluster a median beacon node. The administrative node then sends host $\mathcal{H}_a$ a list of median beacon nodes. The rest of the operations follow the procedure given in Section V-D.

VI. Empirical Study

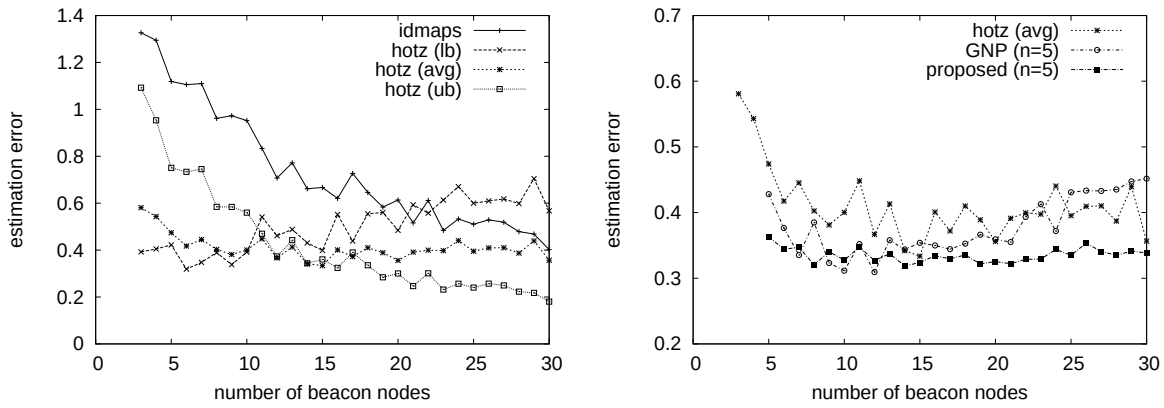
To validate the effectiveness of ICS in inferring the Internet topology, we conduct experiments using both an empirical data set (NLANR) [16] and a synthetic data set (GT-ITM) [19]. As discussed in Section IV-B, the NLANR data set contains real delay data measured by *ping*. The GT-ITM data set, on the other hand, is obtained using the GT-ITM topology generator [19] and the *ns*-2 simulator [20]. The quality of a coordinate system can be affected by several factors such as the number and distribution of beacon nodes and the complexity of the network topology. With the use of the GT-ITM topology generator, we are able to study ICS under a wide variety of network topologies, and investigate the effect of network topology on the performance of ICS. For each data set, we randomly select m beacon nodes ($3 \leq m \leq 30$).

We compare ICS against with IDMaps, Hotz's triangulation, and GNP with respect to the average of estimation errors $\mathcal{E}_{ij}$ defined as

$$\mathcal{E}_{ij} = \frac{|d_{ij} - L(i,j)|}{d_{ij}},$$

for $i, j \in \{1, \ldots, H\}$ and $i \neq j$. Here H is the number of hosts in the data sets, d_{ij} is the measured distance, and $L(i,j)$ is the estimated distance between the i^{th} and j^{th} hosts. IDMaps, Hotz's triangulation, GNP, and ICS are implemented as follows:

- IDMaps: Suppose hosts $\mathcal{H}_a$ and $\mathcal{H}_b$ are close to the i^{th} and the j^{th} beacon nodes (called *tracers* in IDMaps), respectively, and their corresponding distances are denoted as d_{ai} and d_{bj}. Then the estimated distance is $d_{ai} + d_{bj} + d_{ij}$, where d_{ij} is the distance between the i^{th} and j^{th} beacon nodes.



(a) IDMaps and Hotz's triangulation (b) GNP and ICS

Fig. 6. Comparison of IDMaps, Hotz's triangulation, GNP, and ICS for the NLANR data set (n: dimension of coordinate system).

- Hotz's triangulation: With Eq. (4), we calculate three Hotz's distances, i.e., the lower bound (denoted as *lb*), the upper bound (denoted as *ub*), and the average of the two bounds (denoted as *avg*).
- GNP: We solve the two optimization problems minimizing J_1 in Eq. (5) and J_2 in Eq. (6) using the 'fminsearch' function in Matlab (which implements the Simplex Downhill method). We vary the dimension of the coordinate system from n = 2 to 10, and report the most representative results.
- ICS: We evaluate both the full and partial measurement methods in a coordinate system with dimension varying from $n = 2$ to 10. In the partial measurement method, we compare the performance between the cases where beacon nodes are randomly selected and are determined by clustering. The number of beacon nodes which a host measures its distance to is set to $k = n + 1, 2n,$ and $3n$, where n is the dimension of the coordinate system. In the partial measurement method, the missing elements in the distance vector $\mathbf{l}_a$ of host $\mathcal{H}_a$ are estimated by Hotz's triangulation (as was discussed in Section V).

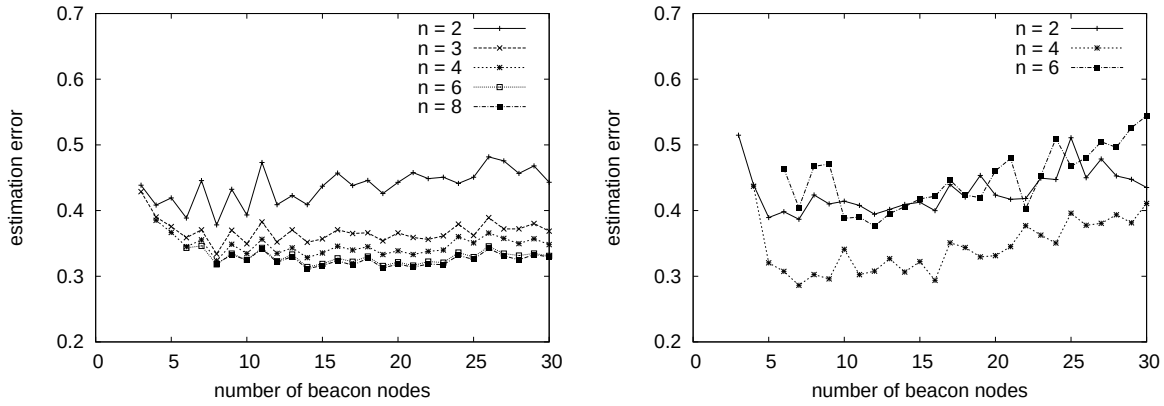
A. Results for the NLANR data

Comparison in terms of estimation errors: Figure 6 (a) gives the estimation errors of IDMaps and Hotz's triangulation. The error obtained by IDMaps is quite large, but gradually decreases from 1.32 at m = 3 to 0.40 at m = 30. As the estimate is calculated by the sum of the

three distances $d_{ai} + d_{bj} + d_{ij}$, if the two beacon nodes are on the shortest path, the estimate well approximates the network distance. This accounts for the fact that the estimate becomes more accurate as m increases. The upper bound of Hotz's triangulation exhibits the same trend as IDMaps. As m increases, the probability that the beacon nodes are on the shortest path between two hosts also increases. The lower bound is quite accurate when m is small. However, the estimation error increases as m increases. Consistent with the findings in [5], the average of the two bounds renders a more accurate estimate of the network distance, and is less susceptible to the number of beacon nodes.

Figure 6 (b) gives the estimation errors of GNP, ICS with the full measurement method, and Hotz's triangulation with the average of the two bounds. GNP performs better than Hotz's triangulation when the number of beacon nodes is small ($m \leq 15$). However, its estimation error increases as m increases, and becomes almost the same as that of Hotz's triangulation. This is probably due to the fact that a local minimum (rather than a global minimum) is selected in the optimization problems. Consider, for example, the case that there exist twenty beacon nodes and the dimension of the coordinate system is five. The cost function J_1 is minimized in a hundred-dimensional vector space, i.e., the number of variables in the coordinates of beacon nodes is 100. In general, an optimization problem of high dimensions easily converges to a local minimum, which in turn leads to inaccuracy in the coordinates of hosts, as explained in Section III-B. ICS gives the best performance. Considering that the RTT measurement between two hosts usually exhibits a large variation (the average of the standard deviation of RTT measurements is approximately 32 % of the RTT measurement), the delay estimated by ICS is quite accurate. In most cases, it incurs lower estimation errors than IDMaps. It gives the same performance as GNP when $m < 15$ and better performance when $m \geq 15$. Here, we select the dimension of the coordinate system to be five as the improvement is marginal when $n \geq 5$ as shown in the next figures.

Effect of the coordinate system dimension on the performance: Figure 7 depicts the effect of the dimension of the coordinate system on the performance of ICS ((a)) and GNP ((b)). The estimation error of ICS is the largest when the dimension of the coordinate system is two (n = 2), and improves as the network topology is represented in higher dimensional space. However, the improvement levels off when $n \geq 6$. Note that the cumulative percentage t_6 for n = 6 is 78.14 % in Fig. 4. The estimation error of GNP is the smallest when n = 4, and is even slightly

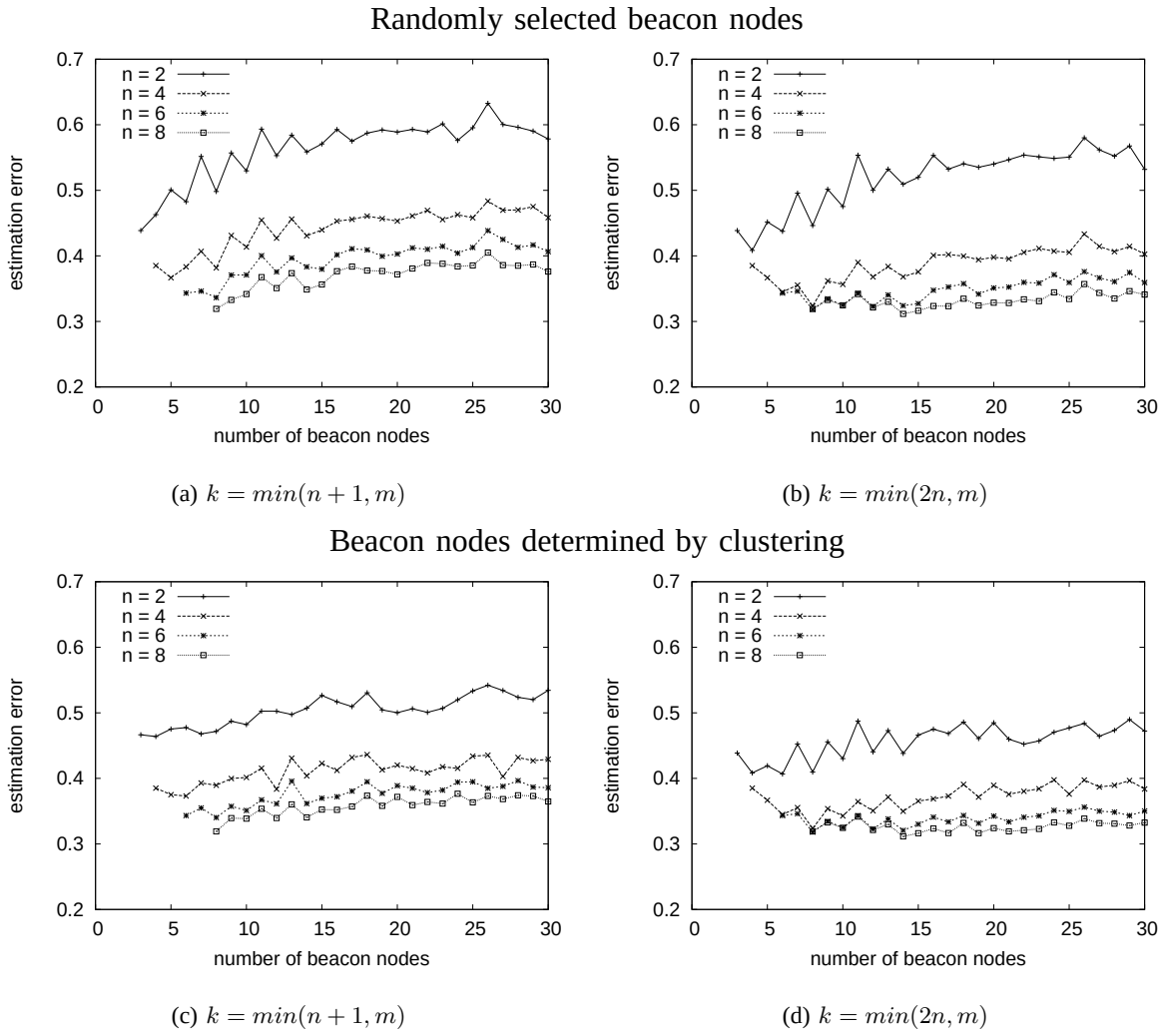


(a) ICS (b) GNP

Fig. 7. The effect of the dimension of the coordinate system on the performance of ICS and GNP for the NLANR data set (n: dimension of coordinate system).

better than that of ICS in the range of $5 \leq m \leq 16$. Note also that the estimation error of GNP when $n = 6$ is much larger than that when $n = 4$. This is again due to the reason that the number of variables increases as n increases. This shows that the accuracy of GNP depends on the selection of the dimension of the coordinate system.

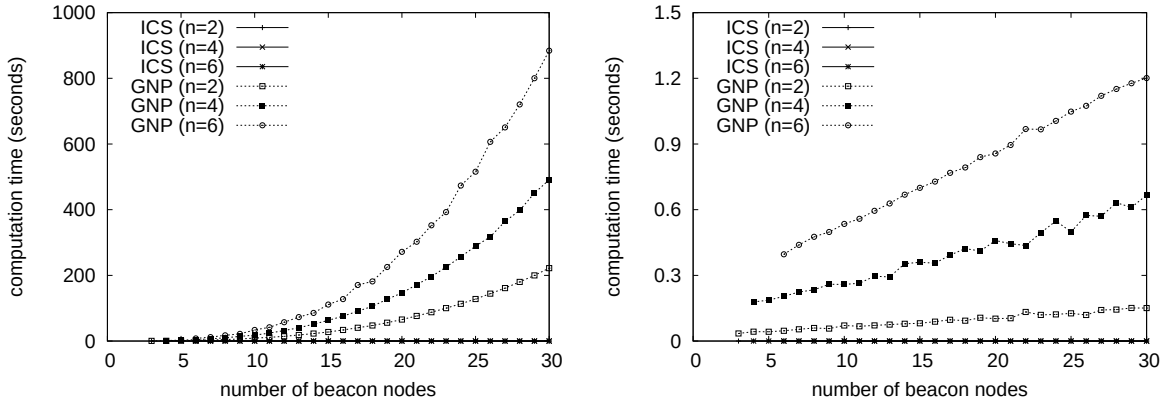
Figure 8 gives the results of ICS with the use of partial measurement method. Among m beacon nodes, k beacon nodes are either randomly selected ((a) and (b)), or determined by the clustering technique [18] ((c) and (d)). The number of measurements made by a host is now proportional to the coordinate dimension n, i.e., $k = \min(n + 1, m)$ in (a) and (c), and $k = \min(2n, m)$ in (b) and (d). As shown in Fig. 8 (a), when $n = 6$, a client measures its distances to six beacon nodes regardless of the value of m, and the average of the estimation errors is increased by 20.2 % (from 0.3287 in Fig. 7 (a) to 0.3951). When the number of measurements is doubled in Fig. 8 (b), the average of the estimation errors is increased only by 6.0 % (as compared to Fig. 7 (a)) in the case of $n = 6$. An encouraging result is that the estimation error does not significantly increase even when the number of measurements is small (i.e., $m \gg k$). This is perhaps due to the fact that measurements made in a coordinate system with the use of more beacon nodes are more accurate. As shown in Fig. 8 (c) and (d), when the median node of each cluster is chosen as a beacon node, the estimation errors are comparatively smaller than those in Fig. 8 (a) and (b), respectively. This implies that the partial measurement method benefits from choosing most representative beacon nodes (i.e., the median node of each cluster).



(a) $k = min(n + 1, m)$ (b) $k = min(2n, m)$

(c) $k = min(n + 1, m)$ (d) $k = min(2n, m)$

Fig. 8. Performance of ICS with the partial measurement method for the NLANR data set (n: the dimension of coordinate system, k: the number of measurements, and m: number of beacon nodes).



(a) Beacon nodes (b) Ordinary hosts

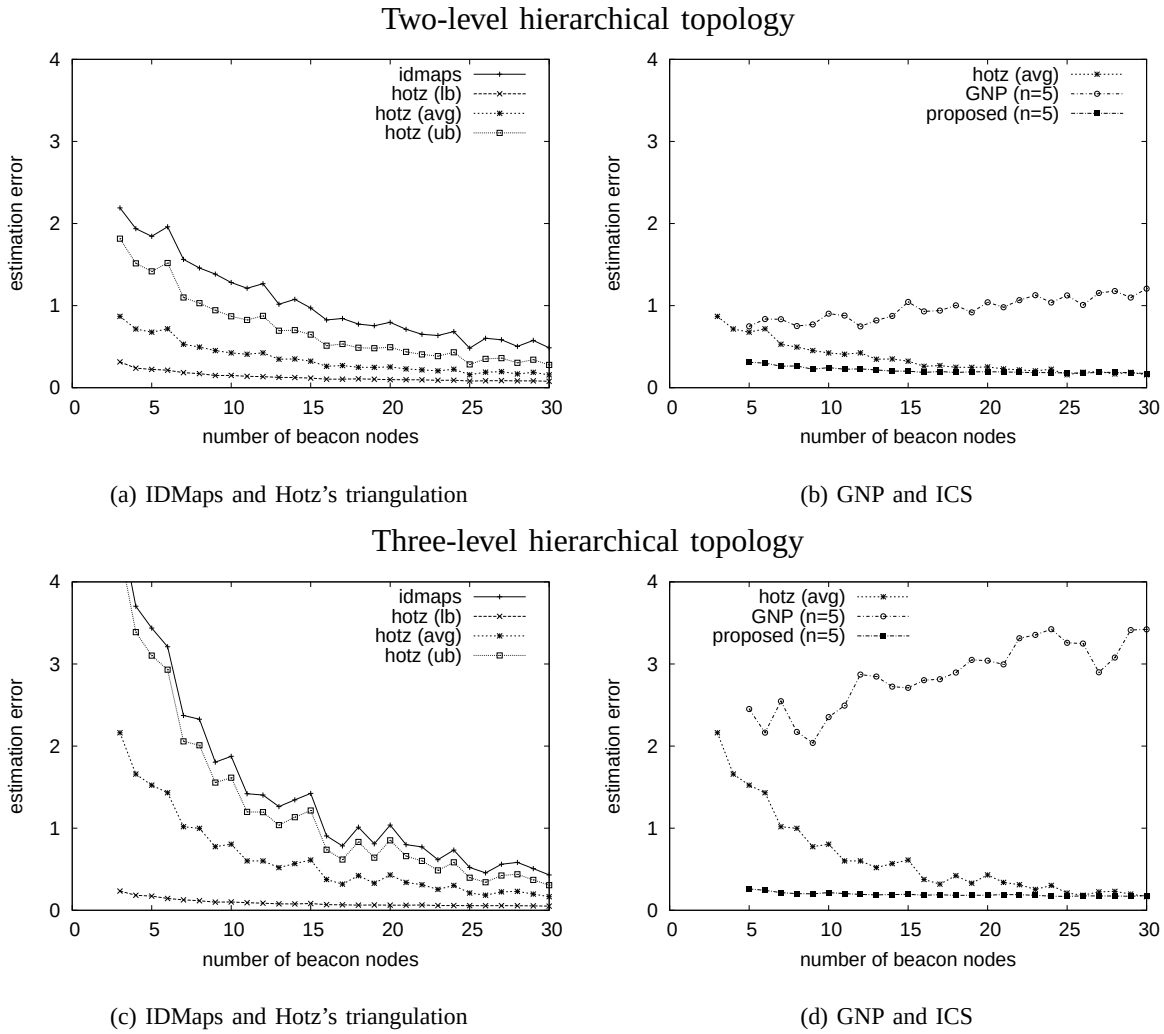
Fig. 9. Comparison between ICS and GNP with respect to computational costs incurred in calculating the coordinates of beacon nodes and ordinary hosts (n: dimension of coordinate system).

Comparison between ICS and GNP in terms of computational costs: To study the computational costs incurred by ICS and GNP, we have implemented their functions of computing coordinates with the 'svd' and 'fminsearch' functions in Matlab, and made the measurement by using the 'cputime' function on an IBM Thinkpad T30 (with a single 1.8 GHz Pentium IV processor and 512 MBytes main memory) that runs Microsoft Windows XP. Figure 9 shows the average CPU time consumed in computing the coordinates of beacon nodes ((a)) and ordinary hosts ((b)) under ICS and GNP. As shown in Fig. 9 (a), as the number of beacon nodes increases, the computation time of GNP for calculating the coordinates of beacon nodes exponentially increases. When the dimension is 6 and the number of beacon nodes is 30, the computation time of GNP is 884.06 seconds (about 15 minutes). With ICS, the maximal computation time is approximately 17.1 milliseconds. Similarly, as shown in Fig. 9 (b), the computational time incurred in calculating the coordinate of an ordinary host can be up to 1.2 seconds under GNP, while remaining less than 30 microseconds for all cases under ICS. This suggests that ICS incurs at least an order of magnitude smaller computation overhead in calculating the coordinates than GNP.

B. Results for the GT-ITM data

We now investigate the effect of topology complexity on the estimation. As mentioned in [19], the GT-ITM topology generator can be used to create three types of graphs: flat random graphs, hierarchical graphs, and transit-stub graphs. We generate two-level and three-level hierarchical graphs, each with 400 nodes. Note that each graph has the same number of nodes; however, three-level hierarchical graphs represent more complex network topologies.

Effect of topology complexity on the performance: Figure 10 (a) and (b) depict the performance of IDMaps, Hotz's triangulation, GNP, and ICS under the two-level hierarchical topology. As shown in Fig. 10 (a), methods that represent the network topology in a distance space give large estimation errors when the number of beacon nodes m is small, and their performance gradually improves as m increases. Among IDMaps and the three versions of Hotz's triangulation, the lower bound of Hotz's triangulation gives the best performance. As shown in Figure 10 (b), between the two coordinate-system-based approaches, GNP renders large estimation errors, and the errors increase as m increases. The estimation error of ICS, on the other hand, is 0.30 at $m = 5$, decreases as m increases, and becomes 0.17 at $m = 30$.



(a) IDMaps and Hotz's triangulation

(b) GNP and ICS

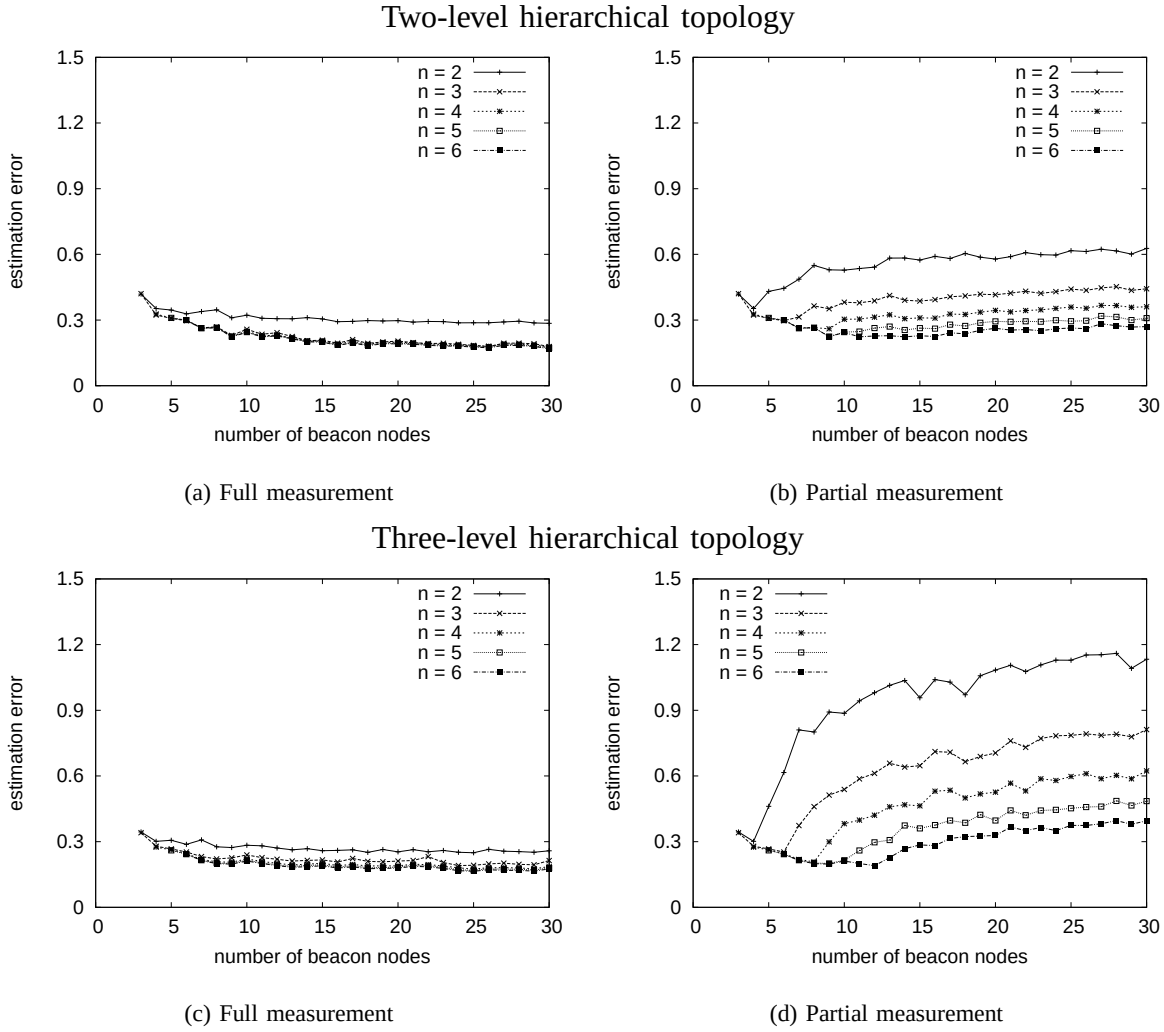
(c) IDMaps and Hotz's triangulation

(d) GNP and ICS

Fig. 10. Comparison of IDMaps, Hotz's triangulation, GNP, and ICS for the GT-ITM data set (n: dimension of coordinate system).

As shown in Fig. 10 (c) and (d), all the approaches, except ICS, give larger estimation errors under three-level hierarchical topologies. In particular, the performance of GNP deteriorates quite significantly. ICS gives almost the same performance as in two-level hierarchical topologies. This result shows that PCA (upon which ICS is built) can effectively extract topological information than the minimization optimization of cost functions J_1 and J_2 in Eq. (5) and Eq. (6) used in GNP.

Effect of the dimension of coordinate systems on the performance: Figure 11 depicts the effect of the coordinate system dimension on the performance of ICS with the full and partial measurement methods. The number of measurements made in the partial measurement method



(a) Full measurement

(b) Partial measurement

(c) Full measurement

(d) Partial measurement

Fig. 11. Effect of the dimension of the coordinate system on the performance of ICS for the GT-ITM data set (n: dimension of coordinate system).

is set to $k = min(2n, m)$, and beacon nodes to which the distance measurement is made are randomly selected among m beacon nodes. Under all the cases, as the dimension, n, of the coordinate system increases, the estimation errors decrease. As shown in Fig. 11 (a), there is virtually no performance improvement when $n \geq 3$, which implies that a three-dimensional coordinate space is sufficient to represent the two-level hierarchical topology. However, when the partial measurement method is applied, the estimation error increases from 0.209 to 0.407 in the case of $n = 3$. This means that even though a three-dimensional space is sufficient to represent the network topology, the number of measurements required should be larger than six in order to determine the coordinates of hosts accurately. As shown in Fig. 11 (c), the estimate made by

ICS is quite accurate under the three-level hierarchical topology, and the errors decrease as n increases. As shown in Fig. 11 (d), the estimation errors become larger when partial measurement is made, but if the number of measurement is larger than $k = 12$, the estimation error can be controlled to fall below 0.32.

In summary, IDMaps and the upper bound of Hotz's triangulation are inaccurate in the case that the number, m, of beacon nodes is small. Their performance improves as m increases. In contrast, the lower bound of Hotz's triangulation is accurate in the case that m is small for the NLANR and GT-ITM data sets, and the errors become larger for the NLANR data set as m increases. As compared with the two bounds of Hotz's triangulation, the average of the two bounds is less sensitive to the number of beacon nodes. GNP can estimate distances accurately only when the number of variables in the corresponding optimization problems is small, i.e., the number of beacon nodes and the dimension of the coordinate systems are small. ICS provides accurate estimates under most cases, regardless of the number of beacon nodes (as long as it exceeds a certain threshold), the dimension of the coordinate systems, and the level of topology complexity. ICS with the partial measurement method reduces the number of measurements required, while not significantly degrading the performance. This is especially true when the number of beacon nodes and the dimension of the coordinate systems are large. Moreover, more accurate estimation can be made with the partial measurement method if beacon nodes are chosen with respect to certain clustering criterion.

VII. Conclusion

In this paper, we present a new coordinate system, called the *Internet Coordinate System (ICS)*, for measuring the network distance over the Internet. We show that the principal component analysis (PCA) technique can effectively extract topological information from delay measurements between beacon hosts. Based on PCA, we devise a transformation method that projects the raw distance space into a new coordinate system of (much) smaller dimensions. The transformation retains as much topological information as possible and yet enables end hosts to determine their locations in the coordinate system based on a small number of measurements. We show via experiments using both real measured and synthetic data sets that ICS can make accurate and robust estimates of network distances between end hosts and is much less computationally expensive, regardless of the number of beacon nodes, the dimension of coordinate systems, and

the level of topology complexity. Finally, we show the number of measurements made by a host can be further reduced without significant loss of accuracy.

REFERENCES

[1] H. Lim, J. C. Hou, and C.-H. Choi, "Constructing Internet coordinate system based on delay measurement," in *Proceedings of ACM Internet Measurement Conference*, 2003.

[2] 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*, 1999.

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

[4] S. Hotz, *Routing information organization to support scalable interdomain routing with heterogeneous path requirements*, Ph.d. thesis, Univ. of Southern California, 1994.

[5] J. D. Guyton and M. F. Schwartz, "Locating nearby copies of replicated Internet servers," in *Proceedings of ACM SIGCOMM*, 1995.

[6] L. Tang and M. Crovella, "Virtual landmarks for the Internet," in *Proceedings of ACM Internet Measurement Conference*, 2003.

[7] M. Pias, J. Crowcroft, S. Wilbur, T. Harris, and S. Bhatti, "Lighthouses for scalable distributed location," in *Proceedings of International Workshop on Peer-to-Peer Systems (IPTPS)*, 2003.

[8] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, MIT Press, 1990.

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

[10] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Computer Journal*, vol. 7, pp. 308–313, 1965.

[11] I. T. Jolliffe, *Principal component analysis*, New York: Springer-Verlag, 1986.

[12] B. Noble and J. W. Daniel, *Applied Linear Algebra*, Prentice Hall, 1988.

[13] K. Y. Yeung and W. L. Ruzzo, "Principal component analysis for clustering gene expression data," *Bioinformatics*, vol. 17, no. 9, pp. 763–774, 2001.

[14] C. Ding, X. He, H. Zha, and H. Simon, "Adaptive dimension reduction for clustering high dimensional data," in *Proceedings of the 2nd IEEE Int'l Conf. Data Mining*, 2002, pp. 147–154.

[15] T. P. Minka, "Automatic choice of dimensionality for PCA," in *Technical report 514, MIT Media Laboratory*, 2000.

[16] V. Paxson, "End-to-end routing behavior in the Internet," in *Proceedings of SIGCOMM '96*, August 1996.

[17] National laboratory for applied network research, "Active measurement project (AMP)," http://watt.nlanr.net/.

[18] A. Jain and R. C. Dubes, *Algorithms for clustering data*, Prentice Hall, 1988.

[19] E. W. Zegura, K. Calvert, and S. Bhattacharjee, "How to model an Internetwork," in *Proceedings of IEEE INFOCOM*, 1996.

[20] The Network Simulator - ns-2, http://www.isi.edu/nsnam/ns/ns-documentation, 2001.