

Impact of the Inaccuracy of Distance Prediction Algorithms on Internet Applications—an Analytical and Comparative Study

Rongmei Zhang^{1,†}, Chunqiang Tang,[§] Y. Charlie Hu,[†] Sonia Fahmy,^{*} and Xiaojun Lin[†]

[†] School of Electrical and Computer Engineering, Purdue University

[§] IBM T.J. Watson Research Center

^{*} Department of Computer Science, Purdue University

Abstract—Distance prediction algorithms use $O(N)$ Round Trip Time (RTT) measurements to predict the N^2 RTTs among N nodes. Distance prediction can be applied to improve the performance of a wide variety of Internet applications: for instance, to guide the selection of a download server from multiple replicas, or to guide the construction of overlay networks or multicast trees. Although the accuracy of existing prediction algorithms has been extensively compared using the relative prediction error metric, their impact on *applications* has not been systematically studied.

In this paper, we consider distance prediction algorithms from an application’s perspective to answer the following questions: (1) Are existing prediction algorithms adequate for the applications? (2) Is there a significant performance difference between the different prediction algorithms, and which is the best from the application perspective? (3) How does the prediction error propagate to affect the user perceived application performance? (4) How can we address the fundamental limitation (i.e., inaccuracy) of distance prediction algorithms?

We systematically experiment with three types of representative applications (overlay multicast, server selection, and overlay construction), three distance prediction algorithms (GNP, IDES, and the triangulated heuristic), and three real-world distance datasets (King, PlanetLab, and AMP). We find that, although using prediction can improve the performance of these applications, the achieved performance can be dramatically worse than the optimal case where the real distances are known. We formulate statistical models to explain this performance gap. In addition, we explore various techniques to improve the prediction accuracy and the performance of prediction-based applications. We find that selectively conducting a small number of measurements based on prediction-based screening is most effective.

I. INTRODUCTION

In recent years, network distance (latency) prediction has been proposed as an alternative to on-demand network measurement. Distance prediction uses $O(N)$ measurements of Round Trip Time (RTT) to predict the N^2 RTTs among N nodes. It can be applied to improve the performance of a wide variety of Internet applications, for instance, to guide the selection of a download server from several replicas, or to guide the construction of overlay networks or multicast trees.

Several approaches to predicting network distances (e.g., [7], [8], [9], [16], [18], [29]) have been proposed. Most

of these use synthetic coordinates in a geometric space to characterize node locations in the Internet, and predict the distance between two nodes as the distance between their coordinates. Existing proposals for network distance prediction have been shown to achieve a good overall prediction accuracy. For example, the relative prediction error of GNP [16] can be 0.5 or less for up to 90% of predicted links.

Although the accuracy of existing prediction algorithms has been extensively compared using the relative prediction error metric, their impact on *applications* has not been systematically studied. Previous work has suggested using prediction to guide the selection of nearby nodes [19], [24], [25], and has shown performance gains over random selection. In this paper, we will demonstrate that there still exists a significant performance gap between prediction-based versions of the applications and the optimal versions where the real distances between nodes are known.

Specifically, we study three types of representative Internet applications: overlay multicast, server selection, and overlay construction. Our results suggest that entirely relying on predicted distances can lead to inferior application performance due to the inaccuracy of the prediction. For instance, when predicted distances are used to guide the construction of a multicast tree, the cost of the tree can be several times higher than that of the tree constructed based on real distances. We show that this phenomenon is closely tied to the high prediction error for short distances, which is found to be intrinsic to existing prediction algorithms.

We formulate statistical models to analyze the impact of the prediction error on the performance of the applications, including nearest neighbor selection and multicast tree construction. Our analysis reveals that, independent of the datasets used in our evaluation, the performance gap between the prediction-based versions of the applications and the optimal versions is significant, given the accuracy of existing prediction algorithms. The accuracy in selecting the shortest links (or the closest neighbors) has a major impact on application performance. However, existing prediction algorithms fail to accurately predict short links.

After experimenting with various enhancements to existing prediction algorithms, such as smart landmark selec-

¹Part of this work was done during Rongmei Zhang’s internship at IBM T.J. Watson Research Center.

tion [7] and alternative error functions, we find that selectively conducting a small number of measurements based on the prediction is most effective in improving the performance of prediction-based applications. Specifically, the candidate pool for the shortest link can be narrowed down based on predicted distances, and then the shortest can be selected based on measurements of the top candidates. Our evaluation shows that selective measurement can bring the quality of the predicted shortest link close to that of the actual shortest link, and therefore allow the applications to perform comparably to the optimal case. Moreover, with selective measurement, recently proposed prediction algorithms (i.e., IDES [15] and GNP [16]) perform similarly to each other. Even the simple triangulated heuristic [11] proposed 11 years ago can achieve a performance reasonably close to the more sophisticated prediction algorithms.

The remainder of the paper is organized as follows. We first give a brief overview of related work in section II. In section III, we describe our evaluation methodology. We evaluate the overall prediction accuracy of selected distance prediction algorithms in section IV, and then evaluate the impact of the prediction inaccuracy on overlay multicast in section V. We formulate and analyze the propagation of distance prediction error in section VI. We then study the selective measurement mechanism and evaluate its effectiveness in remedying the impact of prediction error in section VII. Section VIII studies the impact of applying distance prediction on server selection and overlay construction. We conclude the paper in section IX.

II. BACKGROUND

A number of systems have been proposed for network distance prediction. The triangulated heuristic [11] estimates network distance assuming that the triangle inequality holds. Specifically, each node measures the distances to well-known *landmarks*. The distance between two nodes can be lower and upper bounded based on the triangle inequality; the lower bound and upper bound can then be combined in various ways to estimate the distance between the two nodes. In [19], each node sorts the landmarks in order of increasing distances; two nodes with the same landmark ordering can be estimated to be close to each other. In [13], [26], a node coordinate is first assigned as the distances to the landmarks; principal component analysis (PCA) is applied to reduce the dimensionality of the coordinates.

In GNP [16], the coordinates of the landmarks are first computed by minimizing the error between the measured distances and the estimated distances between the landmark nodes. An ordinary host derives its coordinate by minimizing the error between the measured distances and the estimated distances to the landmarks. GNP uses the Simplex Downhill method to compute node coordinates. NPS [17] builds a hierarchical network positioning system based on GNP. In [7], different strategies for choosing landmarks are studied, including using random nodes, closest nodes, and a hybrid of both types. It shows that choosing nearby nodes as landmarks can improve the prediction accuracy for short links.

In Lighthouse [18], a new node can contact any nodes

already in the system to obtain a coordinate relative to these nodes, and then convert this coordinate into the global coordinate relative to the global landmarks through mapping between coordinate spaces. Node coordinates are computed by solving systems of linear equations.

In Mithos [28], the closest neighbors in the network are selected as landmarks to compute the coordinate of a new node, and this coordinate is assigned as the ID for the new node. The spring relaxation technique is used to calculate node coordinates. In [21], the Big-Bang simulation method is used for embedding network distances in a multi-dimensional space. In [22], the authors observe the curvature property of the Internet and embed network distances in a hyperbolic space with an optimal curvature. In [8], a new coordinate model that consists of a Euclidean coordinate augmented with a height value is proposed. This model is shown to better capture the latency characteristics of the Internet.

In IDES (Internet Distance Estimation Service) [15], a node is associated with both an incoming and an outgoing coordinate vector; the distance between two nodes is estimated as the inner product of the source's outgoing vector and the destination's incoming vector. IDES is proposed to overcome the limitations of the Euclidean space model, i.e., triangle inequality and distance symmetry. Matrix factorization, i.e., Singular Value Decomposition (SVD) or Non-negative Matrix Factorization (NMF), is used to compute node coordinates.

In contrast to coordinate-based prediction mechanisms, IDMaps [9] exploits an infrastructure of servers and specialized hosts called tracers to provide distance estimation service. The distance between two hosts is estimated as the distance to their associated tracers plus the distance between the two tracers. In [27], a set of specialized servers (mServers) measure the distances between themselves; an ordinary host measures the distances to these mServers and associates itself with the closest one; the distance between two ordinary hosts is estimated to be the distance between their assigned mServers. For the purpose of scalability, the mServers are organized into a cluster tree structure. Similarly in [5], hosts are grouped into clusters based on several distance metrics; the distance between a pair of hosts is estimated using intra- and inter-cluster distance. Recently, Meridian [29] has been proposed as a lightweight measurement-based framework for performing node selection based on network location; it can be used to address three commonly encountered node selection problems, i.e., closest node discovery, central leader election, and locating nodes satisfying target latency constraints.

Network distance prediction is potentially useful in many Internet applications, e.g., building topology-aware overlay networks and selecting the closest server [17], [19], [24], [25]. In a recent study [14], Lua et al. have observed the inadequacy of the commonly used relative prediction error metric in calibrating the prediction quality. Alternative performance metrics, such as relative rank loss and nearest neighbor loss, have been introduced to better capture the prediction quality.

III. EVALUATION METHODOLOGY

In this paper, we study the impact of using distance prediction on Internet applications through both experimental evaluation and theoretical analysis. In this section, we briefly describe our evaluation methodology.

A. Distance Prediction Mechanisms

We have selected three distance prediction mechanisms, i.e., the triangulated heuristic, GNP, and IDES, to conduct the evaluations. (Due to space limitation, we omit the results for other algorithms we have implemented, e.g., classical multidimensional scaling.) We select these three because they are representative in their classes. The triangulated heuristic does not involve virtual coordinates; instead, the measured distances to the landmarks are directly used for distance estimation. GNP assigns a virtual coordinate for each node based on measurements to the landmarks. In IDES, a node is assigned two coordinates (incoming and outgoing) in order to address the distance symmetry problem; in addition, the distance between two nodes is computed using the inner product of their coordinates to avoid the limitation of triangle inequality. We compare the three prediction algorithms both in terms of the overall prediction accuracy and the impact on various applications. For each prediction mechanism, 20 nodes are randomly selected as landmarks. The dimensionality of the coordinate space is set to 10 for GNP and IDES.

In IDES, by default, each node computes its coordinate based on the measured distances to and from the landmarks. Given any dimension, the optimal coordinates can be derived by factorizing the full distance matrix between all node pairs. Although the full distance matrix is unavailable in practice, the prediction from the optimal IDES coordinates indicates the best accuracy that these three prediction systems can achieve. Therefore, we plot the results from the optimal IDES coordinates, referred to as “IDES optimal”, in some of our graphs for comparison, while the results from the default IDES prediction are simply labeled as “IDES”.

B. Internet Latency Datasets

We use three Internet latency datasets based on real-world measurements. The first dataset (“King” dataset) is from the P2PSim [2] project. It contains the pair-wise RTTs between 1143 Internet DNS servers measured using the King method [10]. (The original dataset contains more than 1143 servers but we exclude servers with incomplete measurements in order to have a complete distance matrix.) The DNS servers were obtained from an Internet-scale Gnutella network trace. The average RTT between a node pair is about 144 ms and the maximum is about 971 ms. The second dataset (“PlanetLab” dataset) contains the pair-wise RTTs between 169 PlanetLab nodes [23]. The average RTT is about 106 ms and the maximum is about 1005 ms. The third dataset (“AMP” dataset) is from the NLANR Active Measurement Project [1]. The 110 nodes in this dataset are connected to the high performance connection (HPC) networks. The average RTT is about 55 ms and the maximum is 373 ms.

Due to space limitation, we only show representative evaluation results from the three prediction mechanisms and the three latency datasets. Interested readers should refer to an extended version of this paper [30] for complete results.

IV. EVALUATION OF DISTANCE PREDICTION ACCURACY

In this section, we evaluate the performance of GNP, IDES, and the triangulated heuristic in predicting network distances. We measure the prediction accuracy using the **relative prediction error**, which is defined as:

$$\frac{\text{abs}(\text{predicted distance} - \text{measured distance})}{\text{measured distance}}$$

We first investigate the results over the King dataset. Fig. 1 shows the cumulative distribution function (CDF) of the relative prediction error. The three prediction algorithms achieve similar prediction accuracy, although the IDES optimal coordinates are more accurate. Fig. 2 shows the average prediction error over links of various latencies. We can see that the shortest links have the highest prediction error. Although the IDES optimal coordinates generate more accurate predictions for longer links, the predictions for links below 20 ms are equally inaccurate.

The relative prediction error reflects the magnitude of the prediction error. We want to take a closer look at the degree of underestimation or overestimation by the prediction algorithms. Therefore we measure the **directional relative prediction error**, which is defined as:

$$\frac{\text{predicted distance} - \text{measured distance}}{\text{measured distance}}$$

Fig. 4 and 5 plot the directional prediction error of IDES and IDES optimal respectively. The results from the triangulated heuristic and GNP are similar to those from IDES (Fig. 4). In this experiment, the links are first grouped based on their distances; the i th group covers the distance range $[50i, 50(i+1))$. For links within each group, we measure the 10th, 25th, 50th, 75th, and 90th percentile of the directional prediction error. We observe that short links are likely to be overestimated and long links are likely to be underestimated. Moreover, the shortest links are overestimated to similar degrees by IDES and IDES optimal.

Prediction inaccuracy, especially the high prediction inaccuracy for short links, makes it difficult to correctly select the closest node. Fig. 3 shows for each node the distance to the closest node selected based on IDES prediction, versus the distance to the actual closest node. The results are sorted in ascending order of the real distance to the actual closest node. The real distance to the IDES selected closest node can be substantially higher. On average, the real distance to the actual nearest node is 3 ms; the real distance to the closest node selected based on IDES, however, is 117 ms. The results for GNP and the triangulated heuristic are similar.

The prediction accuracy of the three prediction algorithms over the PlanetLab dataset is similar to that over the King dataset. Fig. 6 measures the prediction accuracy of the three prediction algorithms over the NLANR AMP dataset. In contrast to the King and PlanetLab datasets, the AMP dataset

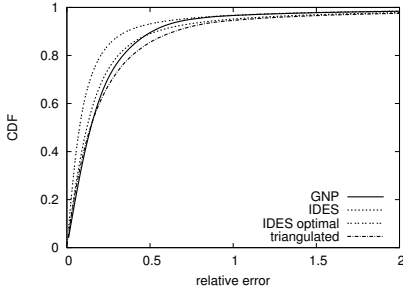


Fig. 1. CDF of relative prediction error (King)

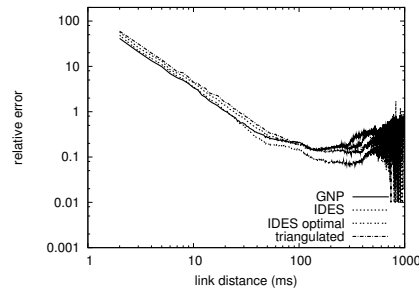


Fig. 2. Average relative prediction error (King)

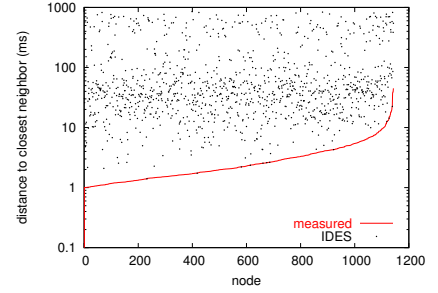


Fig. 3. Distance to closest neighbor selected based on IDES prediction (King)

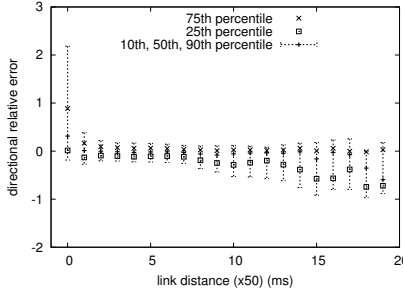


Fig. 4. Directional relative prediction error of IDES (King)

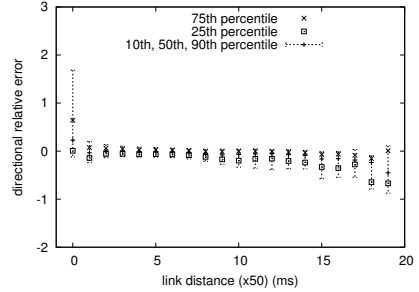


Fig. 5. Directional relative prediction error of IDES optimal (King)

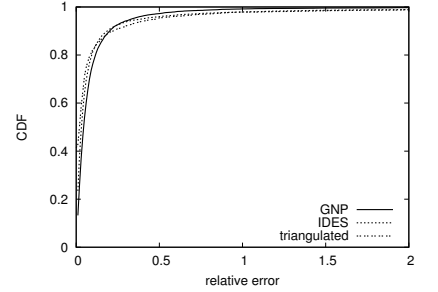


Fig. 6. CDF of relative prediction error (AMP)

can be predicted with high accuracy, even for short links (see [30]). One possible explanation is that the distances in the AMP dataset lie inside a small range with a maximum distance of 373 ms, as opposed to around 1000 ms for the King and PlanetLab datasets. Note that the three prediction algorithms perform equally well for the AMP dataset.

In summary, the prediction accuracy of a distance prediction algorithm can vary widely for different datasets. In addition, for each of the three latency datasets, the differences between the three prediction algorithms (GNP, the default IDES, and the triangulated heuristic) are minor in terms of the overall prediction accuracy. In the rest of the paper, we study the effects of the prediction inaccuracy on the performance of Internet applications.

V. EVALUATION OF THE IMPACT OF PREDICTION INACCURACY ON OVERLAY MULTICAST

The first application we investigate is overlay multicast. Our goal is to answer the following question: how good is an overlay multicast tree constructed based on predicted distances, compared to the tree constructed based on measurements?

There exists a rich body of work on overlay multicast (e.g., [3], [4], [6], [12]). An overlay multicast algorithm usually seeks to optimize some performance metric or a combination of metrics. Since our problem domain is distance prediction, we only consider the latency metric here. Compared to other applications, the performance of overlay multicast is potentially more sensitive to the prediction inaccuracy. For instance, a mistake made earlier in building the tree can potentially alter the entire tree topology.

To focus on the impact of distance prediction rather than the artifacts of any particular multicast algorithm, we study

three simple, abstract algorithms for overlay tree construction: minimum spanning tree (MST), modified ESM [12], and LGK [4]. The MST is constructed using Prim's algorithm. We selected MST for our study because it reflects the ability to correctly select the shortest links in the network by the distance prediction mechanism.

The modified ESM algorithm is a variant of the broadcast ESM protocol [12]. Specifically, a new node to join a multicast tree obtains a partial list of on-tree nodes, and selects one of them as its parent. The parent selection algorithm in [12] chooses the shortest widest path to an on-tree node. We do not consider link capacity or node degree here, and hence a new node selects the closest node in the partial list as its parent. In our evaluation, each new node uses a random sampling of 30 on-tree nodes [12].

The third algorithm, LGK [4], constructs a k -ary tree by exploring node location information. First, the root of the multicast tree selects the closest k nodes as its direct children on the tree. Next, the rest of the nodes are grouped with the k children according to proximity: each remaining node is assigned to the closest of the k children. Ties are broken by load balancing: the node is assigned to the smallest group. In this way, each of the k children is the root of a sub-tree consisting of those nodes close to it. The multicast tree is formed as each subtree repeats the two steps of child selection and clustering. It has been shown that $k = 2$ gives the best tradeoff between the delivery delay and overhead of the multicast tree.

We believe that the above three algorithms capture the two essential building blocks of most overlay multicast protocols, namely, shortest link selection and proximity-based clustering.

For example, LGK shares several features with the NICE protocol [3]: both build the multicast tree through recursive clustering of closest nodes.

A. Evaluation Metrics

We use the **tree cost** to measure the overlay tree quality for the MST and modified ESM algorithms. The tree cost is defined as the sum of the latencies over *all* tree links. This metric measures the error in identifying the shortest links based on predicted distances, either from all the candidate links (for the MST algorithm) or from a random candidate pool (for the modified ESM algorithm). The LGK algorithm aims to optimize the delivery delay instead of the overall cost of the multicast tree. Therefore, we use the **delay stretch** to measure the latency property as perceived by the on-tree nodes. **Delay stretch** is defined as the ratio of the delay on the overlay multicast tree and the delay of the direct unicast path between the root and a tree node. Note that the delay stretch metric is highly sensitive to the overlay tree topology, including the depth of the tree. With all other settings equal, using a different distance prediction mechanism is most likely to generate a completely different tree topology. Although it is difficult to quantify this impact of distance prediction, delay stretch allows us to measure the overall impact on the performance of the overlay multicast tree that might be experienced by the users.

B. Evaluation Results

We first report the results of building overlay trees using distance prediction based on the King dataset. For each tree size n , the tree construction experiment is repeated 100 times, each time with n nodes randomly selected from the dataset. The reported results are averaged over the 100 runs.

Fig. 7 shows the cost of MSTs built using real distances and GNP predicted distances (labeled “measured” and “GNP” respectively). The cost of GNP-based MSTs grows at a much higher rate as the tree size increases from 50 to 400. Fig. 8 plots the cost of overlay trees built by the modified ESM algorithm (labeled “measured” and “GNP”). The cost of GNP-based trees again grows at a high rate. Note that the MST algorithm produces trees with comparable qualities to the modified ESM algorithm when GNP predicted distances are used. Fig. 9 shows the average on-tree delay stretch for the LGK algorithm (labeled “measured” and “GNP”). The delay stretch of LGK trees built based on prediction is more than twice that of LGK trees built based on measurement.

Overall, these results demonstrate that multicast trees constructed under the guidance of predicted distances can be dramatically worse than trees constructed under the guidance of measured distances. In section IV, we have shown that short links tend to be overestimated. We are now interested in quantifying the impact of the high prediction error for short links on overlay tree construction.

We enhance each multicast algorithm with an oracle that can tell the exact latencies for short links below a certain threshold. The results are also reported in Fig. 7, 8, and 9. In these figures, “shortest10” and “shortest50” indicate that the latencies are precisely predicted by the oracle for all links

under 10 ms and 50 ms respectively. The links with latencies under 10 ms account for 7% of all links; the links with latencies under 50 ms account for 17% of all links. For the MST and LGK algorithms, eliminating the prediction error for links under 50 ms makes prediction-based trees almost as good as measurement-based trees. The same effect is not as notable for the ESM algorithm. In ESM, each node selects the closest out of at most 30 candidates to be its parent. The small candidate pool makes the oracle less effective since many of the candidate links might be above 10 ms or 50 ms.

Fig. 10-12 report the results of applying IDES prediction and the triangulated heuristic with the three tree construction algorithms. As before, we observe severe degradation of the tree quality, even with the optimal IDES coordinates. Experiments from adding the oracle produce results (not shown here) that are similar to the “shortest10” and “shortest50” curves in Fig. 7-Fig. 9.

Experiments with the PlanetLab dataset yield results similar to the King dataset, i.e., using prediction alone generates overlay trees that are significantly worse. Fig. 13-15 show the results for the AMP dataset. Although prediction-based overlay trees also experience certain deterioration in terms of the total tree cost or delay stretch, the differences from measurement-based trees are small. For instance, the increase in the cost of MSTs is below 50% using the three prediction algorithms, which is probably acceptable in practice.

In summary, experiments over the three Internet latency datasets suggest that the performance of overlay multicast trees built based on predicted distances is dependent upon the distance prediction accuracy, especially the prediction accuracy of short links. Our study also indicates that, from the application’s perspective, there is no clear winner among the three prediction mechanisms; in some cases, the simple triangulated heuristic is almost as good as the more sophisticated GNP and IDES algorithms.

VI. ANALYSIS OF THE IMPACT OF PREDICTION INACCURACY

In this section, we analyze the impact of the inaccuracy of distance prediction on applications.

A. Analysis of the Impact of Prediction Inaccuracy on Nearest Neighbor Selection

First, we analyze the quality of the nearest neighbor selected by distance prediction algorithms. Suppose node N wants to choose the nearest node out of a set \mathcal{M} of nodes, and $|\mathcal{M}| = k$. For each node $M \in \mathcal{M}$, let X_M be the random variable that denotes the actual distance between node M and node N in the Internet, and let Y_M be the random variable that denotes the predicted distance between node M and node N . Among the k nodes in \mathcal{M} , let A denote the actual closest node to N , and B denote the closest node selected by the distance prediction algorithm. The error introduced by the distance prediction algorithm is then

$$Z(k) \triangleq X_B - X_A, \quad (1)$$

which is again a random variable and $Z(k) \geq 0$. Let $h^k(\cdot)$ denote the probability density function (PDF) of $Z(k)$. As we

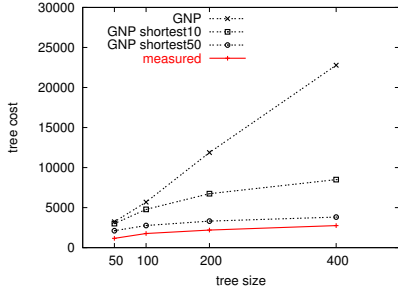


Fig. 7. Tree cost (MST) based on GNP prediction (King)

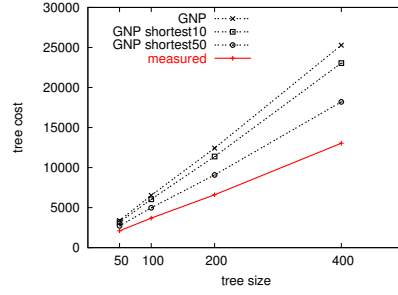


Fig. 8. Tree cost (ESM) based on GNP prediction (King)

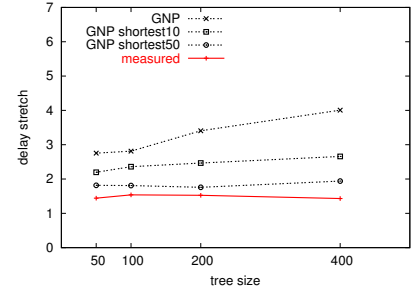


Fig. 9. Delay Stretch (LGK) based on GNP prediction (King)

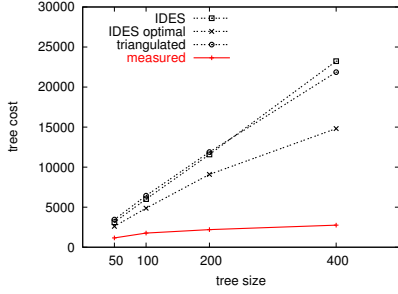


Fig. 10. Tree cost (MST) based on IDES and triangulated heuristic (King)

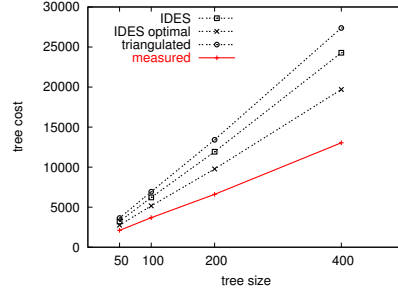


Fig. 11. Tree cost (ESM) based on IDES and triangulated heuristic (King)

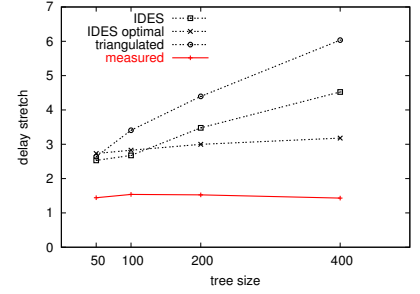


Fig. 12. Delay stretch (LGK) based on IDES and triangulated heuristic (King)

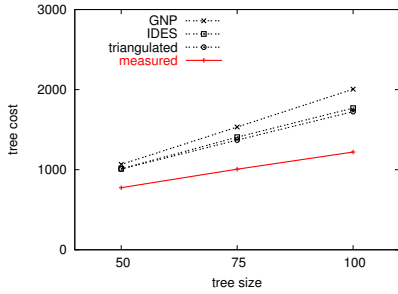


Fig. 13. Tree cost (MST) based on prediction (AMP)

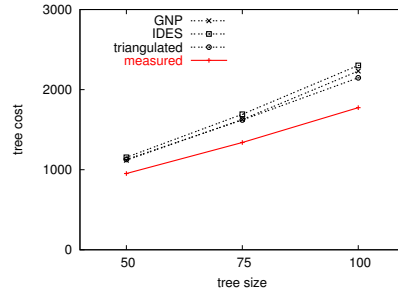


Fig. 14. Tree cost (ESM) based on prediction (AMP)

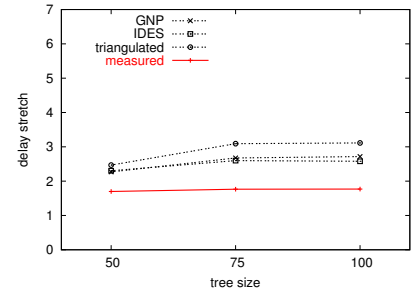


Fig. 15. Delay stretch (LGK) based on prediction (AMP)

will soon see, the function $h^k(\cdot)$ depends on the size k of the candidate set, the distribution of real distances, and the distribution of predicted distances.

We next describe our models for the distribution of real distances and the distribution of predicted distances. (We will specify particular forms for these distributions in section VI-C based on real measurement data, but our analysis also applies to the general model.) We assume that each node is independently and identically distributed in the space. For an arbitrary node in \mathcal{M} , random variable X denotes the actual distance to node N . Let $f(x), x \geq 0$ denote the PDF of X . Random variable Y denotes the predicted distance from this node to node N . We assume that prediction errors are independent. Let $g(y|d)$ denote the PDF of Y given that $X = d$. In Fig. 16, we draw $g(y|d)$ according to a Gaussian distribution, but again our analysis is generic and applies to other distributions as well.

Recall that node A is the actual closest neighbor to node N among the k nodes in \mathcal{M} . Since we assume that the actual

distances from these k nodes to node N are independent from each other, we obtain the PDF of the real distance between N and its closest neighbor A as

$$p_A^k(x_A) = k \cdot f(x_A) \cdot (\Pr[X > x_A])^{k-1}, \quad (2)$$

where

$$\Pr[X > s] = \int_s^\infty f(x) dx \quad (3)$$

is the probability that an arbitrary node in \mathcal{M} is at more than s distance away from N . In Equation (2), we ignore the probability that some nodes in \mathcal{M} are at the same distance to N . For any practical $f(x)$ and constant k , this probability is a high-order term that has no impact on the results of our analysis.

Assume that the real distance between A and N is $X_A = s$, $s \geq 0$, and there is another node $B \in \mathcal{M}, B \neq A$ that is at a distance $X_B = s + e, e > 0$ away from N . Thus, closest neighbor selection introduces an error of e if it mistakenly

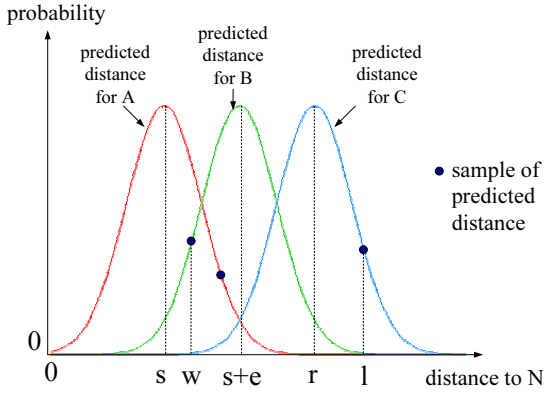


Fig. 16. The distribution of predicted distances between node pairs (N, A), (N, B), and (N, C) respectively. The peaks of the bell shapes are the real distances between (N, A), (N, B), and (N, C) respectively. Node B will be mistakenly predicted as the closest node to N if the sample of the predicted distance for B is the smallest among the three samples (see the dots in the figure).

chooses B as the closest node to N . Next, we calculate the probability for this to happen given $X_A = s$ and $X_B = s + e$. Fig. 16 is an illustration of the analysis process.

Given $X_B = s + e$, the PDF of Y_B , i.e., the predicted distance between N and B , is given by

$$g(w|(s + e)), \quad (4)$$

where w denotes the predicted distance between N and B . The neighbor selection algorithm chooses B as the closest node to N only if w is smaller than Y_A , the predicted distance between N and A . Hence, given $X_A = s$ and $Y_B = w$, the probability that B appears to be a closer neighbor than A based on predicted distances, is given by

$$\Pr[Y_A > w|X_A = s] = \int_w^\infty g(y_A|s) dy_A. \quad (5)$$

For any other node $C \in \mathcal{M}, C \neq A$ and $C \neq B$, the real distance between N and C must be larger than the real distance between A and N . Hence, given $X_A = s$, the conditional PDF of X_C , i.e., the real distance between N and C , is given by

$$f_C(x_C|X_C > s) = \frac{f(x_C)}{\Pr[X > s]}, \quad x_C > s. \quad (6)$$

The prediction algorithm chooses B as the closest node to N only if w is smaller than Y_C , the predicted distance between N and C . Given $X_C = r$ and $Y_B = w$, the probability that C appears to be further away than B in the prediction space is,

$$\Pr[Y_C > w|X_C = r, Y_B = w] = \int_w^\infty g(y_C|r) dy_C.$$

Let $P_{farB}(s, w)$ denote the probability that C appears to be further away than B in the prediction space conditioned on $Y_B = w$ and $X_A = s$. Then,

$$\begin{aligned} P_{farB}(s, w) &= \Pr[Y_C > w|Y_B = w, X_A = s] \\ &= \int_s^\infty \frac{f(r)}{\Pr[X > s]} \int_w^\infty g(y_C|r) dy_C dr. \end{aligned} \quad (7)$$

Combining Equations (4), (5), and (7), and noting that there are $k - 2$ nodes in \mathcal{M} other than A and B , we obtain the probability that B is chosen by the prediction algorithm as the closest node to N conditioned on $X_A = s$ and $X_B = s + e$:

$$\begin{aligned} P_{selB}(s, e) &= \Pr[\text{select } B|X_A = s, X_B = s + e] \\ &= \int_{-\infty}^\infty g(w|s + e) \cdot \Pr[Y_A > w|X_A = s] \\ &\quad \cdot \{P_{farB}(s, w)\}^{k-2} dw. \end{aligned} \quad (8)$$

We next remove the conditioning on X_B . Given $X_A = s$, the real distance between N and B must be larger than s since A is the closest node to N . The conditional PDF of X_B , i.e., the real distance between N and B , is thus given by

$$f_B(x_B|X_B > s) = \frac{f(x_B)}{\Pr[X > s]}, \quad x_B > s. \quad (9)$$

Noting that any of the $k - 1$ nodes in \mathcal{M} other than A can take the role of B in the above analysis, and combining Equations (8) and (9), we have the PDF of introducing error e when selecting the nearest neighbor from a candidate pool of size k conditioned on $X_A = s$ as:

$$h^k(e|X_A = s) = (k - 1)P_{selB}(s, e)f_B(s + e|X_B > s).$$

Hence, using Equation (2),

$$h^k(e) = \int_0^\infty h^k(e|X_A = s)p_A^k(s) ds. \quad (10)$$

Finally, the expectation of the error $Z(k)$ in nearest neighbor selection is then,

$$\begin{aligned} E[Z(k)] &= \int_0^\infty e \cdot h^k(e) \cdot de \\ &= \int_0^\infty de \left\{ e \int_0^\infty ds \left\{ k(k-1)f(s)f(s+e) \right. \right. \\ &\quad \cdot \int_{-\infty}^\infty dw \left\{ g(w|s+e) \left\{ \int_w^\infty dl \cdot g(l|s) \right\} \right. \\ &\quad \cdot \left. \left. \int_s^\infty dr \left\{ f(r) \left\{ \int_w^\infty dl \cdot g(l|r) \right\} \right\}^{k-2} \right\} \right\} \right\}, \end{aligned} \quad (11)$$

which is a function of the size k of the candidate pool, the distribution $f(x)$ of real distances, and the distribution $g(y|d)$ of predicted distances.

B. Analysis of the Impact of Prediction Inaccuracy on Overlay Multicast

In this section, we analyze the quality of multicast trees created under the guidance of distance prediction algorithms. As a baseline for comparison, we first analyze the tree cost for the modified ESM protocol with two assumptions: (i) the real distance between any two nodes is known; (ii) a new node has full knowledge of all the on-tree nodes. Let random variable $T(n)$ denote the cost of an n -node tree built by this protocol. Below we calculate the expectation of $T(n)$.

Let random variable $D(k)$ denote the real distance between a node N and its nearest neighbor selected out of k random nodes. According to Equation (2), the expectation of $D(k)$ is

$$E[D(k)] = \int_0^\infty s \cdot p_A^k(s) ds. \quad (12)$$

During the process of building an n -node tree, nodes are added into the tree one by one. On average, each step increases the tree cost by $E[D(k)]$. Therefore, the expectation of the tree cost $T(n)$ is

$$T(n) = \sum_{k=1}^{n-1} E[D(k)], \quad (13)$$

where n is the number of nodes in the tree.

In a prediction-based version of the baseline protocol, a new node N has no knowledge of the real distances between itself and the on-tree nodes. Instead, node N predicts the distances to the on-tree nodes and selects the node that has the shortest predicted distance as its parent. Suppose the current tree consists of k nodes. When N joins, it adds a link that is on average $E[Z(k)]$ (see Equation (11)) longer than the actual shortest link between N and an on-tree node. Therefore, compared to the baseline protocol where real distances are known, this prediction-based protocol introduces an additional tree cost of $\sum_{k=2}^{n-1} E[Z(k)]$, and results in a total tree cost of

$$T'(n) = T(n) + \sum_{k=2}^{n-1} E[Z(k)]. \quad (14)$$

C. Models for Network Distances

The analysis in section VI-A represents real distances and predicted distances as random variables X and Y with PDF $f(x)$ and conditional PDF $g(y|d)$, respectively. The estimation of $f(x)$ and $g(y|d)$, however, can be challenging. On the one hand, sophisticated models with a large number of parameters may approximate measured and predicted distances more accurately. On the other hand, overly sophisticated models can be a barrier for understanding the analysis results. Our goal here is to develop simple models that approximate the real data reasonably well.

Fig. 17(a)-(c) plot the CDF of the measured latencies for the three datasets. To make the figures readable, we cut off the latency on the x -axis at a certain level, i.e., excluding extremely long latencies from the figures. Despite the fact that the distribution of Internet latency is sophisticated, these figures surprisingly suggest that the latency distribution is close to a uniform distribution (note that an ideal uniform distribution should be a straight line across the diagonal of the figure). We therefore assume a uniform distribution for $f(x)$.

$$f(x) = \begin{cases} 1/H & \text{if } 0 \leq x \leq H \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

The uniform distribution cannot model outliers with latencies longer than H , but this error is tolerable since the fraction of outliers is small and the applications we consider—server selection, application-level multicast, and overlay construction—tend not to use those outliers with long latencies.

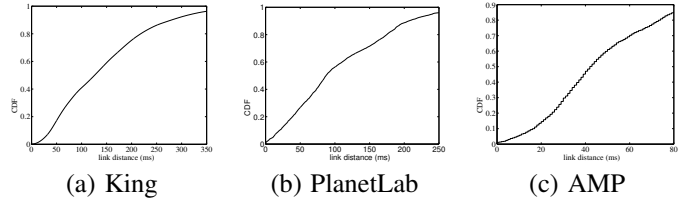


Fig. 17. CDF of measured network distances

Substituting Equations (15), (2), and (3) into Equation (12), we obtain the expectation of the distance to the nearest neighbor in a candidate pool of size k .

$$E[D(k)] = \frac{H}{1+k} \quad (16)$$

Equation (16) indicates that, for even a medium k , the distance to the nearest neighbor is short. Given the fact that existing prediction algorithms are not good at estimating short distances, prediction alone cannot choose the nearest neighbor with a high accuracy.

Substituting Equation (16) into Equation (13), we obtain the expectation of the cost of a tree with n nodes.

$$T(n) = \sum_{k=1}^{n-1} E[D(k)] = \sum_{k=1}^{n-1} \frac{H}{1+k} \quad (17)$$

Next, we develop a model for predicted distances. One natural model for prediction error is the Gaussian distribution

$$g(y|d) = \frac{1}{\sigma_d \sqrt{2\pi}} \exp\left\{-\frac{(y - \mu_d)^2}{2\sigma_d^2}\right\}, \quad (18)$$

where μ_d and σ_d are functions of d . To further simplify the model, we assume

$$\mu_d = d + c \quad (19)$$

$$\sigma_d = \sigma, \quad (20)$$

where both c and σ are constants. We introduce the bias c because we observe that existing prediction algorithms tend to overestimate the distances over the range being considered here ($c > 0$).

Fig. 18 shows the matching between the Gaussian model and the predicted distances. The y axis is the CDF of distance prediction error. The “real distribution” curve plots the difference between real distances and predicted distances by IDES (using the King dataset). The Gaussian distribution in this figure corresponds to the function $g(z) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left\{-\frac{(z-c)^2}{2\sigma^2}\right\}$, where $\sigma = 22$ and $c=2.29$ (see Equation (18) and note that $z = y - d$ is the prediction error). This figure suggests that the Gaussian model accurately captures the distribution of prediction error.

D. Numerical Solutions

As Equations (11), (13), and (14) cannot be calculated directly, we use the Monte Carlo method to develop an approximate solution for a given configuration. For each configuration, our program generates 1,000,000 different instances for node N and the candidate neighbor pool, and then computes the

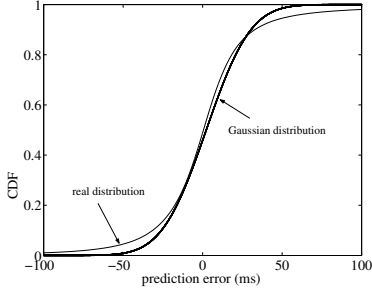


Fig. 18. CDF of prediction error using IDES prediction (King)

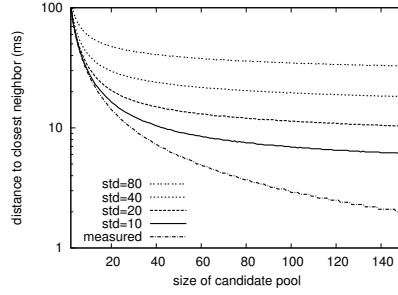


Fig. 19. Distance to closest neighbor based on real distances (bottom curve) and predicted distances (the other curves)

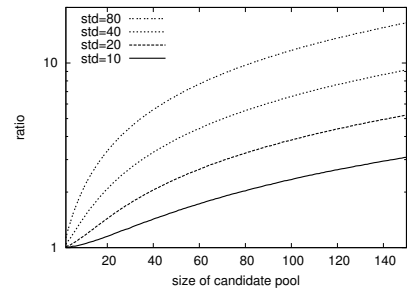


Fig. 20. Ratio between distance to predicted closest neighbor and distance to actual closest neighbor

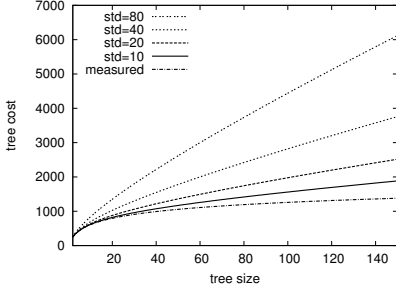


Fig. 21. Tree cost based on real distances (bottom curve) and predicted distances (the other curves)

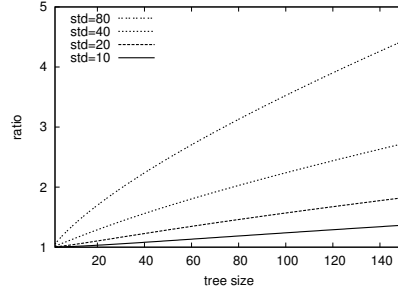


Fig. 22. Ratio between tree cost based on real distances and tree cost based on predicted distances

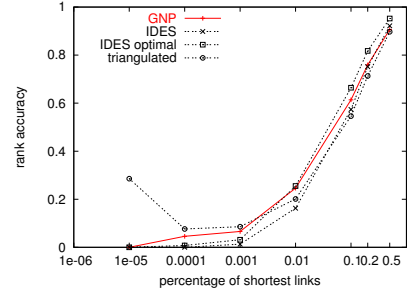


Fig. 23. Ranking accuracy (King)

sampled values for real distance X and predicted distance Y . Our program varies the standard deviation σ and the size k of the candidate pool, while fixing $H = 300$ ms (Equation 15) and $c = 2.29$ ms (Equation 19). Fig. 19-22 show the impact of the inaccuracy of distance prediction on nearest neighbor selection and overlay multicast. The curves correspond to results extracted from Equations (11), (13), and (14).

The standard deviation σ indicates the accuracy of distance prediction, which varies with different datasets. We have measured the value of σ for the three prediction algorithms over the three distance datasets. Using the IDES prediction, $30 < \sigma < 50$ for the King dataset and the PlanetLab dataset, and $8 < \sigma < 15$ for the AMP dataset. σ also varies with different choices of landmark nodes. The tree cost curves in Fig. 21 under the corresponding σ values for the various datasets show similar trends as the simulation results in the previous section (e.g., Fig. 10-11 and Fig. 13-14). In addition, these analysis results confirm that, independent of the datasets used in our evaluations, given the accuracy of existing prediction algorithms, the performance gap between the prediction-based versions of the applications and the measurement-based versions is significant.

VII. SELECTIVE MEASUREMENTS

In the previous sections, we have evaluated and analyzed the impact of network distance prediction on overlay multicast. Our study suggests that it is not advisable to solely rely on network distance prediction in building an overlay multicast tree, especially due to the high prediction error for short distances and thus the high error in selecting the shortest links. We have

studied several enhancements to existing distance prediction algorithms, including smart landmark selection [7], varying the distance function and error function, and alternative techniques to extract information from the measured distances (e.g., classical multidimensional scaling, neural networks, and linear regression). (Detailed results are omitted here due to limited space.) Although some of these enhancements improve the prediction accuracy in certain cases, our experiments indicate that they cannot fundamentally shrink the performance gap between the prediction-based applications and the measurement-based applications.

One natural solution is to combine measurement with distance prediction. Then the question becomes how many and which links to measure. In this section, we present a selective measurement scheme that selectively performs a small number of measurements to help choose the shortest links.

We first evaluate the **ranking accuracy** [16] of all three prediction mechanisms to demonstrate that prediction alone cannot rank the short links accurately. Assume that $PredictedLinks_p$ and $MeasuredLinks_p$ denote the shortest p links (p stands for a percentage) selected based on predicted distances and measured distances respectively. **Ranking accuracy** is defined as $\frac{|PredictedLinks_p \cap MeasuredLinks_p|}{|MeasuredLinks_p|}$, where $|X|$ denotes set X cardinality. Fig. 23 plots the ranking accuracy as p varies from 0.01% to 50%. We observe that the ranking accuracy for the shortest 5-10% links is fairly high (around 50%).

Based on this observation, we devise a selective measurement scheme as follows: each time the shortest link is to be selected, we can first narrow down the candidates by selecting

a small number of links with the lowest predicted latencies, and then measure the actual latencies for those links to identify the best one. In the simplest form, the number of measured links can either be a constant m or a constant fraction p (e.g., 5-10%) of the total candidate links.

A. Evaluation of Selective Measurement in Overlay Multicast

In this section, we evaluate the impact of selective measurement on the quality of multicast trees constructed based on predicted distances. For the MST algorithm, the original candidate pool contains all links from the on-tree nodes to the remaining nodes. The amount of measurements is bounded by $m(n-1)$ if a constant number m of links are selected each time. We also experiment with measuring a uniform fraction p of the candidate links, which requires $p \sum_{i=1}^{n-1} i(n-i)$ total measurements. For the ESM algorithm, the candidate pool for a new node is defined by the random subset of on-tree nodes. The total number of measurements is $m(n-1)$, if each new node is allowed m measurements. For the LGK algorithm, during the k closest nodes selection step, the candidate pool includes all the nodes in the subtree rooted at the current node (subtree root). The total number of measurements varies with the height of the tree ($O(\log_k n)$).

Fig. 24 depicts the impact of applying selective measurement with GNP prediction on the MST algorithm. We evaluate two selective measurement schemes: in the “uniform” scheme, the 5% predicted shortest links from the candidate pool are selected for measurement; in the “constant” scheme, the 20 predicted shortest links are measured. We can see that selectively measuring 5% shortest links can achieve tree costs close to using pure measurement, and this scheme works well for all tree sizes. On the other hand, the “constant” scheme is not as effective for larger trees as the “uniform” scheme. This can be explained by the larger candidate pools as the tree size increases, since it is more difficult to make the right selection for a smaller selection ratio (Fig. 23). Nevertheless, even the “constant” scheme can reduce the tree cost growth rate by about 50%.

For the ESM algorithm, the selective measurement scheme selects the predicted closest 10 out of the 30 random subset of on-tree nodes for measurement. Fig. 25 shows that this configuration can bring the tree cost to near that of measurement-based trees. Fig. 26 gives the results of using selective measurement with the LGK algorithm. During the k -children selection, the 20% closest nodes based on predicted distances are chosen for measurement.

Fig. 27-29 show the effectiveness of applying selective measurement with the triangulated heuristic. The results from applying selective measurement with the IDES prediction are similar (see [30]). Our experiments indicate that, when combined with selective measurement, there exist no significant performance differences between the three prediction mechanisms from the perspective of the application.

VIII. USING DISTANCE PREDICTION IN SERVER SELECTION AND OVERLAY CONSTRUCTION

Thus far, we have studied the impact of using network distance prediction on overlay multicast. In this section, we

investigate the impact of distance prediction on server selection and overlay construction.

A. Server Selection

First, we consider the problem of server selection and measure the selection accuracy using predicted distances against using measured distances. Specifically, we randomly select a number of nodes from each trace as the servers, and measure the stretch of using prediction to select the closest server. The stretch is defined as the distance to the closest server selected based on prediction, divided by the distance to the actual closest server.

Fig. 30 shows the average stretch using GNP, IDES, and the triangulated heuristic over the King dataset. Although using prediction can reduce the stretch considerably compared to random selection, the results are far from optimal. For instance, the average stretch is above 3.65 when selecting from 32 servers and above 5.71 from 64 servers using IDES prediction. The stretch is even higher for the PlanetLab dataset (details in [30]). In contrast, for the AMP dataset (Fig. 32), the stretch is below 2.6 for up to 40 servers. These results closely match the analysis results on nearest neighbor selection (curves with corresponding values of σ in Fig. 20). We can see that the benefit from using distance prediction for server selection is also highly sensitive to the prediction accuracy. Similar to overlay multicast tree construction, server selection can also benefit from selective measurement. For example, if we allow 10 measurements based on the prediction (refer to Fig. 31), we find that the stretch is closest server selection can be reduced by more than 50%.

B. Overlay Construction

In this subsection, we study the impact of using distance prediction on building overlay networks, including both unstructured and structured overlay networks. We use the results from the King dataset in our discussion.

We first investigate building unstructured overlays using predicted distances. Our overlay construction protocol is similar to the protocol proposed in [24]. Specifically, each node maintains a number of random links to other nodes, and a number of links to nearby nodes. In our experiment, each node is connected to one randomly selected node. A previous study [24] has shown that this configuration is sufficient to guarantee the connectivity of the overlay network with a very high probability. In addition, each node maintains up to 6 neighbors that are close in terms of network distance. We measure the cost of the overlay, which is defined as the sum of the cost over all overlay links. The results are reported in Fig. 33. From the figure, prediction-based overlays have much higher costs than measurement-based overlays. For overlays of 1000 nodes, the total cost is increased by approximately 3 times using GNP or IDES prediction, and by about 4 times using the triangulated heuristic. Fig. 34 illustrates the effects of applying selective measurement with the prediction, in which the 20 predicted closest nodes are measured each time the close neighbors are selected. A comparison of Fig. 33 and 34 shows that selective measurement dramatically reduces the overlay cost.

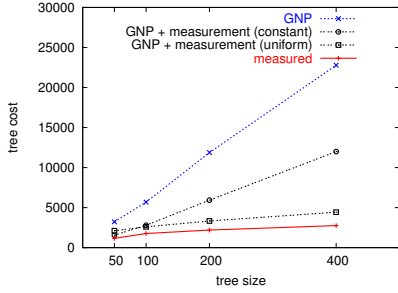


Fig. 24. Tree cost (MST) based on GNP with selective measurement (King)

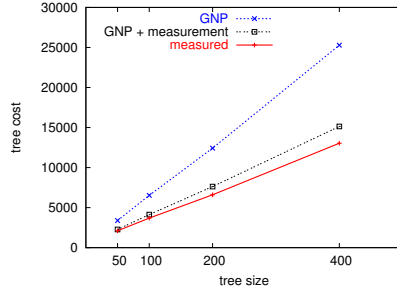


Fig. 25. Tree cost (ESM) based on GNP with selective measurement (King)

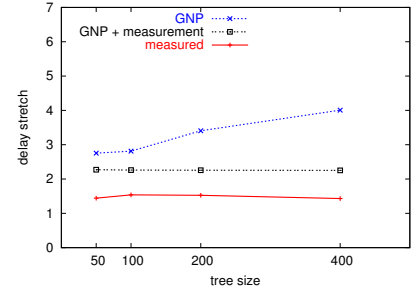


Fig. 26. Delay stretch (LGK) based on GNP with selective measurement (King)

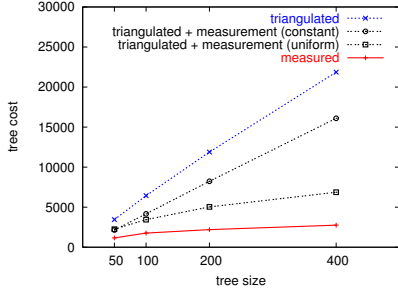


Fig. 27. Tree cost (MST) based on triangulated heuristic with selective measurement (King)

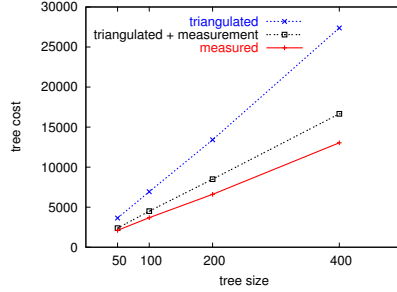


Fig. 28. Tree cost (ESM) based on triangulated heuristic with selective measurement (King)

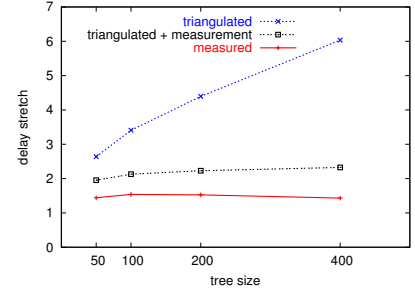


Fig. 29. Delay stretch (LGK) based on triangulated heuristic with selective measurement (King)

In order to study the impact of distance prediction on structured overlays, we construct Pastry [20] networks based on predicted distances. In Pastry, each node maintains a routing table based on node identifier prefixes. When there is more than one node satisfying the identifier prefix constraint, the closest in terms of network distance can be selected (referred to as “proximity-awareness”). For each Pastry overlay of size n , n Ping messages are sent from randomly selected source nodes towards randomly selected destinations. Fig. 35 reports the average routing delay stretch when the Pastry overlay is constructed using measured distances and predicted distances respectively. The routing delay stretch is defined as the delay along the overlay routing path divided by the delay of the direct path from the source to the destination. The results suggest that distance prediction works well in Pastry-like structured overlay construction, as the increase in the routing delay stretch is below 20%. The intuition behind this is as follows. In Pastry, the latency of the last routing hop dominates and the choices for the last hop routing are only a few at best. Therefore, the prediction algorithms can do well in selecting the shortest one from them. Our experiments with unstructured and structured overlays further confirm that the outcome of selecting the shortest links based on prediction varies with the size of the candidate pool. All the three prediction mechanisms work well when the candidate pool is small (as in structured overlay construction).

IX. CONCLUSION

In this paper, we have considered Internet distance prediction from an application’s perspective. We have studied the impact of the inaccuracy of distance prediction algorithms on Internet applications by systematically experimenting with three types of representative applications (overlay multicast,

server selection, and overlay construction), three distance prediction algorithms (GNP, IDES, and the triangulated heuristic), and three Internet distance traces (King, PlanetLab, and AMP). We have also developed an analytic framework to aid in understanding the impact of the distance prediction error on the application’s performance.

Our major findings can be summarized as follows.

- Existing distance prediction algorithms are inadequate for the applications in terms of the prediction accuracy. The performance of the prediction-based versions of the applications can be significantly worse than the measurement-based versions.
- Both our analytical and experimental results suggest that the prediction accuracy for short links has a major impact on application performance. Unfortunately, existing prediction algorithms are found to be inaccurate in predicting these short links.
- Combining selective measurement with distance prediction is very effective in improving application performance. When selective measurement is used, we have observed no major performance differences between the selected distance prediction algorithms, and the choice of the prediction algorithm itself becomes less important.

One possible direction for our future work is to study more advanced distance prediction algorithms for improving the prediction accuracy. We are also interested in solving practical issues with distance prediction such as those discussed in [17].

ACKNOWLEDGMENT

The authors would like to thank Prof. Ness Shroff, Dr. Rong Chang, and the anonymous reviewers for their helpful comments. This work was supported in part by NSF CAREER award grant ACI-0238379 and CNS-0238294.

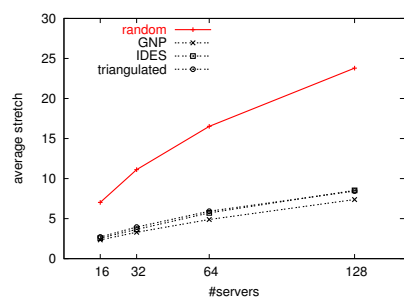


Fig. 30. Stretch in server selection using prediction (King)

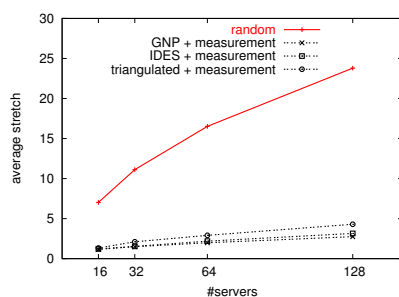


Fig. 31. Stretch in server selection using prediction and selective measurement (King)

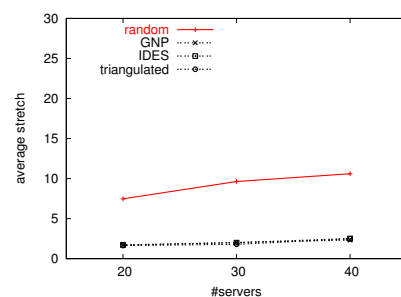


Fig. 32. Stretch in server selection using prediction (AMP)

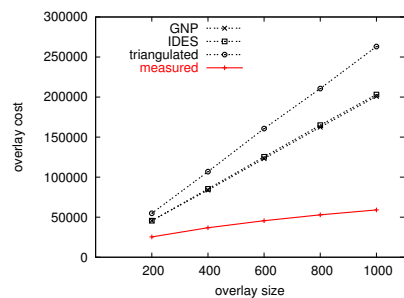


Fig. 33. Unstructured overlay cost based on prediction (King)

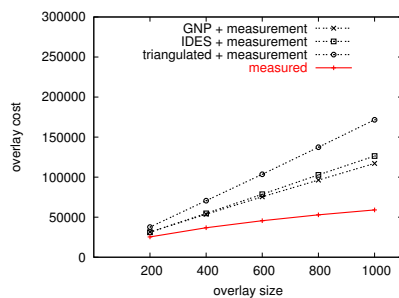


Fig. 34. Unstructured overlay cost based on prediction with selective measurement (King)

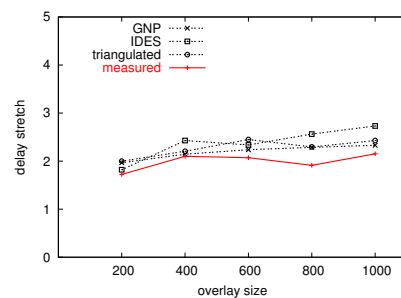


Fig. 35. Routing delay stretch in Pastry based on prediction (King)

REFERENCES

- [1] NLANR Active Measurement Project. <http://amp.nlanr.net/>.
- [2] The P2PSim Project. <http://pdos.csail.mit.edu/p2psim/>.
- [3] S. Banerjee, B. Bhattacharjee, and C. Kommareddy. Scalable Application Layer Multicast. In *Proceedings of ACM SIGCOMM*, August 2002.
- [4] K. Chen and K. Nahrstedt. Effective Location-Guided Tree Construction Algorithms for Small Group Multicast in MANET. In *Proceedings of IEEE INFOCOM*, June 2002.
- [5] Y. Chen, K. H. Lim, R. H. Katz, and C. Overton. On the Stability of Network Distance Estimation. *ACM SIGMETRICS Performance Evaluation Review*, 30, September 2002.
- [6] Y.-H. Chu, S. G. Rao, and H. Zhang. A Case for End System Multicast. In *Proceedings of ACM SIGMETRICS*, 2000.
- [7] M. Costa, M. Castro, A. Rowstron, and P. Key. PIC: Practical Internet Coordinates for Distance Estimation. In *Proceedings of IEEE ICDCS*, March 2004.
- [8] F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi: A Decentralized Network Coordinate System. In *Proceedings of ACM SIGCOMM*, August 2004.
- [9] P. Francis, S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang. IDMaps: A Global Internet Host Distance Estimation Service. *IEEE/ACM Transactions on Networking*, October 2001.
- [10] K. P. Gummadi, S. Saroiu, and S. D. Gribble. King: Estimating Latency between Arbitrary Internet End Hosts. In *Proceedings of SIGCOMM Internet Measurement Workshop (IMW)*, November 2002.
- [11] S. M. Holz. Routing Information Organization to Support Scalable Interdomain Routing with Heterogeneous Path Requirements, 1994. Ph.D. Thesis, University of Southern California.
- [12] Y. hua Chu, A. Ganjam, T. E. Ng, S. G. Rao, K. Sripanidkulchai, J. Zhan, and H. Zhang. Early Experience with an Internet Broadcast System Based on Overlay Multicast. In *Proceedings of USENIX*, June-July 2004.
- [13] H. Lim, J. C. Hou, and C. ho Choi. Constructing an Internet Coordinate System Based on Delay Measurement. In *Proceedings of ACM IMC*, October 2003.
- [14] E. K. Lua, T. Griffin, M. Pias, H. Zheng, and J. Crowcroft. On the Accuracy of Embeddings for Internet Coordinate Systems. In *Proceedings of ACM IMC*, October 2005.
- [15] Y. Mao and L. K. Saul. Modeling Distance in Large-Scale Networks by Matrix Factorization. In *Proceedings of ACM IMC*, October 2004.
- [16] T. S. E. Ng and H. Zhang. Predicting Internet Network Distance with Coordinates-Based Approaches. In *Proceedings of IEEE INFOCOM*, June 2002.
- [17] T. S. E. Ng and H. Zhang. A Network Positioning System for the Internet. In *Proceedings of USENIX*, June 2004.
- [18] M. Pias, J. Crowcroft, S. Wilbur, T. Harris, and S. Bhatti. Lighthouses for Scalable Distributed Location. In *Proceedings of IPTPS*, February 2003.
- [19] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker. Topologically-Aware Overlay Construction and Server Selection. In *Proceedings of IEEE INFOCOM*, June 2002.
- [20] A. Rowstron and P. Druschel. Pastry: Scalable, Distributed Object Location and Routing for Large-Scale Peer-to-Peer Systems. In *Proceedings of ACM/FIP/USENIX Middleware*, November 2001.
- [21] Y. Shavitt and T. Tankel. Big-Bang Simulation for Embedding Network Distances in Euclidean Space. In *Proceedings of IEEE INFOCOM*, April 2003.
- [22] Y. Shavitt and T. Tankel. On the Curvature of the Internet and Its Usage for Overlay Construction and Distance Estimation. In *Proceedings of IEEE INFOCOM*, March 2004.
- [23] J. Stribling. PlanetLab All Pairs Pings. http://www.pdos.lcs.mit.edu/~strib/pl_app.
- [24] C. Tang, R. N. Chang, and C. Ward. GoCast: Gossip-Enhanced Overlay Multicast for Fast and Dependable Group Communication. In *Proceedings of DSN*, June 2005.
- [25] C. Tang and S. Dwarkadas. Hybrid Global-Local Indexing for Efficient Peer-to-Peer Information Retrieval. In *Proceedings of USENIX NSDI*, March 2004.
- [26] L. Tang and M. Crovella. Virtual Landmarks for the Internet. In *Proceedings of ACM IMC*, October 2003.
- [27] W. Theilmann and K. Rothermel. Dynamic Distance Maps of the Internet. In *Proceedings of IEEE INFOCOM*, March 2000.
- [28] M. Waldvogel and R. Rinaldi. Efficient Topology-Aware Overlay Network. In *Proceedings of ACM HotNets*, October 2002.
- [29] B. Wong, A. Slivkins, and E. G. Sirer. Meridian: A Lightweight Network Location Service without Virtual Coordinates. In *Proceedings of ACM SIGCOMM*, August 2005.
- [30] R. Zhang, C. Tang, Y. C. Hu, S. Fahmy, and X. Lin. Impact of the Inaccuracy of Distance Prediction Algorithms on Internet Applications: an Analytical and Comparative Study. Technical Report TR ECE 06-01, Purdue University, January 2006. http://www.ece.purdue.edu/~ychu/publications/infocom06_tr.pdf.