NetForecast: A Delay Prediction Scheme for Provider Controlled Networks

Ahmed Elmokashfi
Simula Research Laboratory
Oslo, Norway
Email: ahmed@simula.no

Michael Kleis
Fraunhofer Institut FOKUS
Competence Center for Autonomic
Networking Technologies (ANTS)
Berlin, Germany
Email: michael.kleis@fokus.fraunhofer.de

Adrian Popescu
Blekinge Institute of Technology
School of Engineering
Dept. of Telecommunication Systems
Karlskrona, Sweden
Email: adrian.popescu@bth.se

***Abstract*— Over the last years, the Internet has evolved towards becoming the dominant platform for deploying real time and multimedia services. This evolution has had as a consequence that the selection of an appropriate server, proxy or super node with reference to some specific QoS parameter becomes of paramount importance. We consider in our paper the specific case of delay estimation. An investigation of existing approaches for the estimation and prediction of network delay is provided. Based on that, we further suggest NetForecast as a way to overcome limitations of existing prediction methods. NetForecast is an algorithm for delay prediction in provider controlled networks. The algorithm is based on a combination of landmark-based distance estimation, clustering and a triangulation principle. The paper reports on preliminary performance of NetForecast, as provided by a simulation study. Our results show the feasibility of the suggested method.**

I. Introduction

The increasing focus on applications like VoIP, online gaming and IPTV has raised the demand for tools able to select, for a specific service, a server, proxy, relay or super node, which meets the required end-to-end QoS constraints from the client perspective. For a service provider this is a challenging task since, e.g., building a knowledge base about the underlying network characteristics is not possible without carrying (on demand) active measurements, which increases the overhead and may affect the stability of the network. Thus, the task of estimating network characteristics with low probing overhead is essential.

Active measurements represent today some of the most accurate means for verifying delay constraints between different pairs of nodes in the network. This method, however, shows high complexity, i.e., $O(n^2)$, where n denotes the number of nodes in the network. The problem of designing an efficient and scalable methodology for predicting network distances has therefore evolved to become a research topic in its own right.

For instance, IDMaps [6] is one of the first suggested solutions for network distance estimation with relatively low measurements overhead. IDMaps involves the deployment of a set of hosts, called tracers, to play the role of a knowledge base assisting end hosts in estimating network distances between themselves. On the other hand, Global Network Positioning (GNP) [13] is a pioneering approach in the area of network embedding. GNP associates network nodes with points in a low dimensional Euclidean space. A network node in GNP measures the distances to a finite set of nodes called landmarks and uses these measurements to further compute corresponding coordinates in the targeted Euclidean space. After finding the coordinates in the targeted Euclidean space, the network node can predict the network distances to other nodes in the network, based on their coordinates and without any additional measurements. This is possible since the calculation of the nodes coordinates aims at minimizing the discrepancy between the measured and the predicted network distances. Even though the GNP approach is efficient in reducing the active measurement overhead, it lacks however accuracy because of the policy-based routing used today in the Internet [11]. Many alternative solutions have followed GNP with the goal to improve the accuracy of the actual network embedding, such as [2], [4], [9], [10], [12], [14], [17]. On the other hand, a new family of distance estimation techniques emerged to overcome the limitations of the network embedding approach. These techniques are based on using proactive active measurements for building a knowledge base about the network. Meridian [18] and Netvigator [19] are members of this family. However, as reported in the paper, these schemes have important drawbacks like, e.g., high measurement or inter-system communication overhead or the fact that they rely on network infrastructure support.

The main contribution of the paper is NetForecast, a novel technique developed for predicting network delay (more specifically, round trip time (RTT)) in provider controlled environments. We motivate the design of NetForecast and compare its accuracy and overhead with other state of the art prediction techniques.

The remainder of the paper is as follows. A detailed analysis of related work is presented in section II together with comparison of the selected schemes. Section II also highlights the most promising candidates selected for further investigation. Section III reports on the comparative study framework and findings. NetForecast is presented in section IV, together with performance study and important observations obtained from comparing NetForecast with other similar work. Finally, section V concludes the paper.

II. State of The Art

We highlight the existing techniques and classify them into different groups. Furthermore, we provide a comparison of their characteristics followed by identification of the most suitable candidates for the selected application scenario.

A. Classification

Based on the underlying principles used by the studied schemes we suggest a classification into three classes.

I- *Landmarks based distance estimation:* the fundamental principle is to associate each node in the network with a point in a metric space and to predict the communication delay between any two nodes in the network by just measuring their corresponding distance in the metric space [2], [9], [10], [12], [13], [14], [17]. This concept is therefore built around the notion of landmarks, which are usually represented by a set of selected nodes used by others as a measurement reference for calculating own coordinates.

II- *Multidimensional scaling based distance estimation:* as an alternative approach it is possible to address the minimization of the discrepancy between the measured network distances and the computed ones without explicitly involving landmark nodes. Schemes based on a so-called Multidimensional Scaling (MDS) approach, e.g., Vivaldi [4] and BBS [16] are belonging to this class.

III- *Creation and maintenance of a distributed network distance database:* schemes belonging to this class, such as Meridian [18] and Netvigator [19], do not use embedding techniques. Instead, they are proactively using active measurements for building a knowledge base about the underlying network.

B. Comparison of Characteristics

Table I shows a comparison of the most relevant schemes used for network delay estimation. To compare these schemes we use the following metrics:

- *Measurement overhead:* this shows the order of measurements needed to maintain the system without including query overhead.
- *Prerequisites:* this is about any special requirement for implementing and/or using the specific scheme.
- *Churn recovery:* this gives information whether churn recovery is supported or not.
- *Infrastructure dependability:* this gives information whether the scheme requires support from the underlying network infrastructure or not.

The following abbreviations are used in Table I: L corresponds to the number of landmarks used by a scheme and H to the total number of participating hosts. In the case of Meridian, N denotes the number of nodes, m the number of rings per node, GI is the overhead per gossip interval, and RI is the overhead per ring management interval. For GCP, C_s denotes the cluster size. For Netvigator, R denotes the number of milestones (routers) in use. No information is provided about the measurement overhead for Vivaldi since it is not using active measurements, but only piggyback latency information to the application traffic.

C. Candidates

Based on above mentioned criteria, we selected Vivaldi [4], PIC [2] and GCP [9] as appropriate schemes to further investigation. The reasons for this are as follows:

- None of the three selected schemes requires explicit support from the underlying network infrastructure.
- There is no need for fixed landmarks, thus the three candidates can work in a fully decentralized manner.
- The candidates exhibit low overhead when compared with other approaches like Meridian.
- The candidates are suitable to be used directly or with less customization in a P2P context, e.g., DHT [15], compared for instance with Meridian, which is completely orthogonal to DHT-like structures.
- The selected schemes also show less vulnerability to landmarks replacement than in the case of schemes depending on matrix factorization.

It is important to mention that the procedure of evaluation is simplified in this case, given that all selected schemes are based on network embedding. This means that a node in the network is identified with a point in a metric space and the quality of the embedding determines the quality of the actual prediction. For a comparative study of the selected schemes, a simulation based study was done as presented below.

III. Comparative Study

In the following we describe the simulation and evaluation framework used to compare the selected schemes. For a detailed description of this framework we refer to [5].

A. Simulation Environment and Data Sets

A modular packet level simulator has been developed to examine the selected distance predication candidates. We have also used two data sets for this study.

The first data set is the publicly available P2PSim King [1] data set, which represents real Internet measurements. This data set is advantageous because of the network and geographical diversity. We have also found that this data set has a very low number (i.e., 0.84%) of missing measurement pairs and a low number (i.e., 4.1%) of the triples violating the triangle inequality [5].

Furthermore, the second data set is based on a synthetic transit-stub topology generated by using the popular GT-ITM topology generator [20]. The transit-stub topology model focuses on reproducing the hierarchical structure of the topology of the Internet.

B. Evaluation Framework

We define a set of five metrics to be used for the evaluation of the selected schemes. The metrics are as follows:

- *Directional Relative Error (DRE):* this metric is an overall performance measure for the quality of embedding,

Scheme	Measurements overhead	Prerequisites	Churn recovery	Infrastructure dependability
Global Network Positioning (GNP) [13]	$O(L^2 + L \cdot H)$	A set of landmarks	No	No
Practical Internet Coordinates (PIC) [2]	$O(L^2 + L \cdot H)$	P2P substrate	No	No
Geometric Cluster Placement (GCP) [9]	$O(L^2 + L \cdot H + C_s \cdot H)$	P2P substrate	No	No
Internet Coordinates System (ICS) [10]	$O(L^2 + L \cdot H)$	A set of landmarks	No	No
Lighthouses [14]	$O(L^2 + L \cdot H)$	Frame of reference	No	No
Vivaldi [4]	-	Inter-nodes traffic	Yes	No
Meridian [18]	$O(N \cdot m^2)(GI) + O(log^2 N)(RI)$	-	Yes	No
Netvigator [19]	$O(L \cdot H + R \cdot H)$	Routers support for traceroute	No	Yes

TABLE I
COMPARISON OF DELAY PREDICTION SCHEMES

which was introduced in [13]. It measures the magnitude of the deviation between the network distance before and after embedding for each pair of measurements. These values are then aggregated to characterize the system.

- *Stress:* the stress is another overall performance metric to measure the quality of embedding [3]. It measures the magnitude of the deviation between the distance before and after the embedding over all existing pairs in the system. The stress is inherently an aggregate measure.
- *Relative Rank Loss (RRL):* metric first introduced in [11] to measure the embedding from the perspective of relative ranking preservation. If node A has two neighbors B and C where B is closer to A than C, RRL verifies if the relative closeness remains unaffected when mapping the three nodes to points in a low dimensional metric space.
- *Closest Neighbor Loss Significance (CNLS):* this metric was initially introduced in [11] to check whether the closest neighbor in the real network is still the same after embedding or not. We use an improved version of CNLS to make it more expressive, where we incorporate the loss percentage. This gives us more information about the significance of the loss, as defined below.

  Definition Let X denote the set of nodes in a network. For an arbitrary node $x \in X$ let $y \in X$ denote the closest node to x in the network and $z \in X$ the closest node to x after embedding. The closest neighbor loss significance at x is given by:

$$CNLS(x) = \frac{|d(x,y) - d(x,z)|}{d(x,y)} \quad (1)$$

  where is $d(x,y)$ is the measured distance between x and y and $d(x,z)$ is the measured distance between x and z.
- *Recall [7]:* this metric was introduced to measure the correctness of similarity queries after an embedding. We use it to infer the suitability of the schemes for QoS range queries.

  To illustrate the usage of recall in the context of QoS range queries, let X denote the set of nodes in a network. If $z \in X$ is an arbitrary node, let us consider that z executes a query before and after the embedding to identify the other nodes in the network satisfying the criteria $\{y \in X | d(z,y) \leq d_{max}\}$, where $d(z,y)$ denotes the network distance between z and y and d_{max} denotes the maximum accepted network distance. We say that a prediction scheme is suitable for QoS prediction if the query attains high recall, i.e., close to 100%. This means that, after embedding, a query returns all correct hits.

C. Main Observations

The results of the comparative study are reported in [5]. In the following, we report a set of concluding remarks and recommendations regarding the suitability of the selected schemes for accurate delay prediction.

I- A general observation is that the embedding is more accurate for short distances than for medium and long distances. Inaccurate prediction of medium and long distances has consequences in the form of degradation of the overall mapping performance as well as specific performance measures. The consequence therefore is that finding a solution for this problem is expected to improve the overall mapping performance and the specific performance measures.

II- It has been further observed that the landmark based distance estimation techniques PIC and GCP perform better than Vivaldi.

III- GCP has been observed to be more suitable than PIC and Vivaldi in environments with triangle inequality violations when using the P2PSim King data set.

IV- Based on the results obtained by using the GT-ITM topology we observe that the performance of PIC is more sensitive to the triangle inequality violation.

V- The selected schemes have been observed to perform poorly with regards to metrics like RRL and CNLS, especially when using the P2PSim King data set.

VI- We have observed from the recall results that in general, the landmark based schemes have better recall when considering short range queries. This observation implies, from a viewpoint of a network node, that the mapping of nearby nodes shows less error than the mapping of nodes far away.

IV. NETFORECAST

Based on the results reported above, we have developed NetForecast, a new scheme for delay prediction intended to be implemented in provider controlled networks. The goal

was to enhance the prediction accuracy for medium and long network distances as well as to conserve the accurate prediction for short distances of landmark based distance estimation techniques. Furthermore, another important goal was to improve the performance with regard to the specific evaluation metrics.

A. Architecture

NetForecast has a hierarchical architecture, where each node is characterized by two different coordinates, one for predicting short distances and the other one for predicting medium and long distances. The basic concept is to group the network nodes into circular clusters with equal radii, based on their actual network position.

The motivation for using the dual coordinate approach is to avoid using landmarks based distance estimation for predicting medium and long network distances. In contrast, for predicting short distances, NetForecast applies a suitable landmark based distance estimation technique independently by creating a local non-exportable coordinate system for each cluster. To predict the distances between nodes belonging to different clusters, a different technique is used as described below.

B. Operation

Details of the NetForecast operation are as follows.

1) System Initialization: Given that NetForecast has been developed for predicting network delay in provider controlled environments, we assume the availability of information about the underlying network, e.g., network map, node distribution characteristics. The network administration entity is supposed to initially select a set of nodes to be used as cluster heads. This is a procedure that considers the delay distribution in the network and the distribution of the network nodes. Furthermore, the selected cluster heads are expected to be highly reliable and connected to the network via links with high bandwidth.

2) Cluster Characteristics: NetForecast assumes clusters with the following characteristics:

- An unique $Cluster\_ID$ is assigned to the cluster head, either administratively or by using hashing.
- Cluster membership is mutually exclusive.
- All clusters are circles with the same radius in terms of network delay.
- A cluster head allows an arbitrary node to join its cluster if the delay between itself and the node is less than or equal the cluster radius.

3) Node Clustering: A newly joining node receives upon bootstrapping a sorted list of cluster heads as well as the value for the cluster radius in the network. It measures then the RTT to each cluster head and arranges them in a specific tuple characterizing its location in the network with respect to the cluster heads. For example, this tuple can be expressed as $(d_{L_1}, d_{L_2}, ..., d_{L_n})$, where each element reports the measured RTT with the respective cluster head. Furthermore, as a final step the node seeks to join one of the existing clusters by following the steps listed below:

- Identification of the closest cluster head.
- It then checks if the distance between itself and the closest cluster head is less than or equal to the system clusters radius value.
- If the previous check result is true, it sends a cluster join request to the closest cluster head or, alternatively, it sets the $Cluster\_ID$ to -1 denoting so no cluster membership.
- Upon receiving a cluster join request, any cluster head may further check to assure the node eligibility to join its cluster. This is done by measuring its distance with the node. After verifying the eligibility of the requesting node, the respective cluster head sends back a join acceptance along with its $Cluster\_ID$ (which is used by the node as its new $Cluster\_ID$).

4) Network Distance Estimation: NetForecast estimates inter-clusters and intra-clusters distances by using different approaches. By intra-cluster distances we mean the distances between the nodes within the same cluster. By inter-clusters distances we mean the distances between nodes that are members of different clusters.

I- *Intra-Clusters Distance Estimation:* NetForecast applies a suitable landmarks based distance estimation technique locally and independently at each cluster. Furthermore, different clusters have different non-related coordinates systems.

II- *Inter-Clusters Distance Estimation:* the concept of triangulated heuristics [8] is used for the estimation of the network distance between any two nodes belonging to different clusters. The main reason for this is that we assume that at least one of the cluster heads is in the shortest path between any two nodes A and B. Assume for instance that we want to estimate the distance between any two nodes A and B, which are members in different clusters. A is characterized by the tuple $(d_{AL_1}, d_{AL_2}, ..., d_{AL_n})$, which represent its measurements to the cluster heads. B is characterized by the tuple $(d_{BL_1}, d_{BL_2}, ..., d_{BL_n})$, which represent its measurements to the cluster heads. The distance between A and B is then estimated by

$$min_{k \in 1,2,..,n}(d_{AL_k} + d_{BL_k}).$$

C. Performance

We have simulated NetForecast by using the same simulation environment and the same data sets as used in the comparative study. For each simulation run we selected ten cluster heads following the requirements mentioned above. Based on the results obtained from the comparative study we set the cluster radius to 150 ms in the P2PSim King data set and to 250 ms in the GT-ITM topology. These values correspond to distances where landmarks based distance estimation techniques have been observed to achieve better recall. GCP is used for the intra-clusters coordinates system while experimenting with P2PSim King data set. PIC is used for the GT-ITM topology experiments. We used the evaluation metrics presented above for the evaluation of NetForecast and

the comparison with the other three candidates. As a general observation, all schemes achieved better performance when using the GT-ITM topology [5]. In the following we therefore report the results obtained with the P2PSim King data set only.

NetForecast has been simulated and the performance compared with Vivaldi, PIC and GCP with regard to the following parameters:

- Stress, reported in the form of the average stress measured over the whole data set (table II).
- Directional Relative Error (DRE), reported in the form of Cumulative Distribution Function (CDF) (figure 1).
- Local Relative Rank Loss (RRL), reported in the form of CDF (figure 2).
- Closest Neighbor Loss Significance (CNLS), reported in the form of CDF (figure 3).
- Recall, reported in the form of CDF. For each node in the system we calculated the recall in three delay ranges by partitioning the network cloud around each node into three ranges. The first range contains the nodes with a network distance, to the originating node, which is less than or equal to R_1. The second range contains the nodes with a network distance, to the originating node, which is inbetween R_1 and R_2. Finally the third range contains the rest of the nodes, i.e., nodes considered to be far away with regard to delay, larger than R_2. In the P2PSim King data set we set R_1 equal to 150 ms and R_2 equal to 300 ms. On the other hand, we set R_1 equal to 200 delay units and R_2 equal to 400 delay units in the GT-ITM topology. These values are selected based on the distribution of the delay ranges measured in each data set. Figure 4 shows the recall CDF in the first range, figure 5 shows the recall CDF in the second range and figure 6 shows the recall CDF in the third range.

It is observed that NetForecast performs better than the other three candidates with reference to stress, DRE, RRL and CNLS. Furthermore, we also observe that NetForecast shows better recall performance than other schemes in the first range at the beginning for approximately 0.1 of the cumulative fraction of nodes. This is followed by a comparable performance with PIC and Vivaldi but less than GCP. On the other hand, it is observed that NetForecast outperforms the other three schemes with a considerable recall margin in the second and third range.

Scheme	Vivaldi	Pic	GCP	NetForecast
Stress	0.111645004	0.139883934	0.138243647	0.049862

TABLE II
AVERAGE STRESS

Based on these results, we can therefore state that:

I- NetForecast definitely improves the delay predictability at medium and long network distances. This is an observation supported by the fact that, in most of cases, all evaluation metrics show better values for NetForecast than for the other schemes Vivaldi, PIC and GCP.

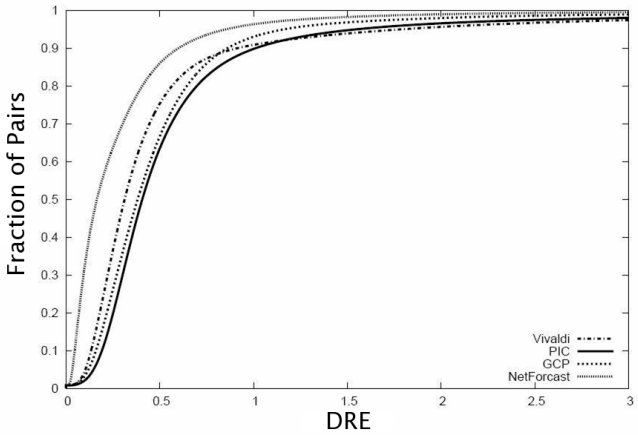

Fig. 1. DRE distribution

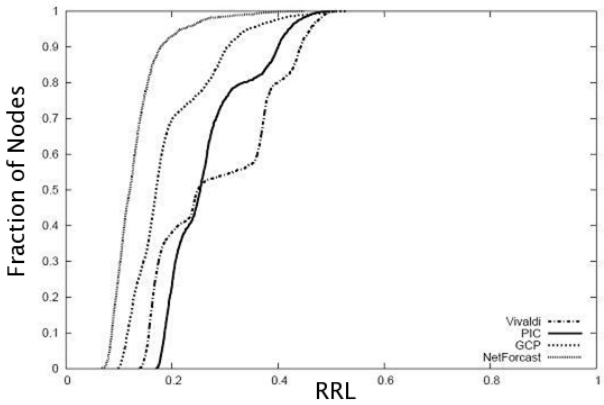

Fig. 2. Local RRL distribution

II- NetForecast depends however on several heuristics, with the consequence that we expect differences in performance from one network to another, depending upon the specific conditions.

III- NetForecast has better immunity to triangle inequality violations than other schemes and it performs best if shortest path routing is used with regard to delay.

A set of guidelines for the implementation of NetForecast are presented in [5].

V. CONCLUSIONS

A novel delay prediction scheme called NetForecast has been introduced, which is intended for controlled network environments. NetForecast is a scalable decentralized delay verification scheme based on using network distance prediction techniques and active measurements. Measurement and simulation studies have shown that NetForecast outperforms other similar schemes like Vivaldi, PIC and GCP.

Planned future work is to develop an analytical framework for specifying the configuration parameters and to validate our results by testing NetForecast in real networks. Another important issue is to develop schemes for predicting other QoS-related parameters, e.g., jitter, bandwidth and packet loss.

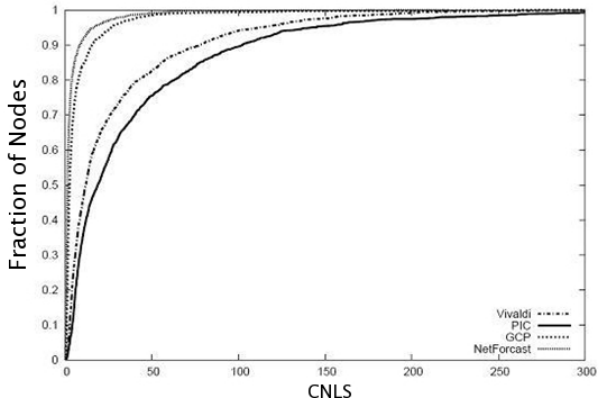

Fig. 3. CNLS distribution

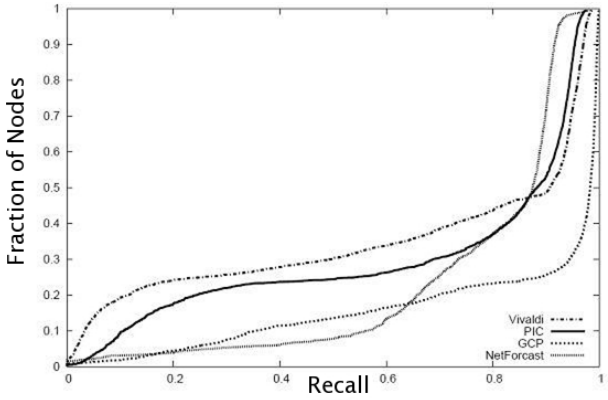

Fig. 4. Recall distribution for the first range

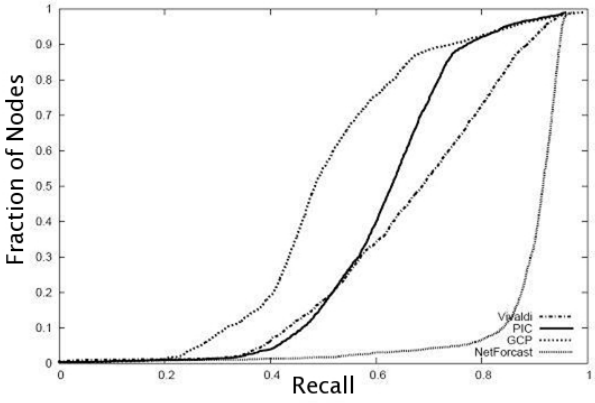

Fig. 5. Recall distribution for the second range

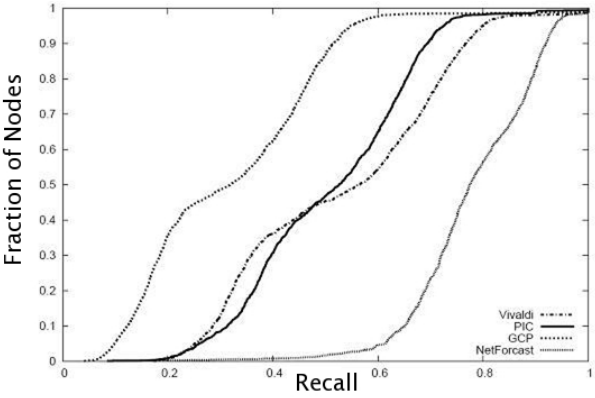

Fig. 6. Recall distribution for the third range

REFERENCES

[1] P2psim king data set. http://pdos.csail.mit.edu/p2psim/, 2006.

[2] M. Costa, M. Castro, A. Rowstron, and P. Key. Pic: Practical internet coordinates for distance estimation. In *Proceedings of the 24th International Conference on Distributed Computing Systems(ICDCS)*, Tokyo, Japan, March 2004.

[3] T. F. Cox and M. A. A. Cox. *Multidimensional Scaling*. Chapman and Hall, London, second edition, 2001.

[4] F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi: A decentralized network coordinate system. In *Proceedings of ACM SIGCOMM Conference*, August 2004.

[5] A. Elmokashfi. Scalable, decentral qos verification based on prediction techniques and acmeasurements. Technical report, Department of Telecommunication Systems, Blekinge Institute of Technology, 2007.

[6] P. Francis, S. Jamin, V. Paxson, L. Zhang, D. F. Gryniewiczand, and Y. Jin. An architecture for a global internet host distance estimation service. In *Proceedings of IEEE INFOCOM*, New York, March 1999.

[7] G. Hjaltason and H. Samet. Contractive embedding methods for similarity searching in metric spaces. Technical Report TR-4102, Computer Science Department, University of Maryland, 2000.

[8] S. Hotz. *Routing Information Organization to Support Scalable Interdomain Routing with Heterogeneous Path Requirements*. PhD thesis, University of Southern California, 2004.

[9] M. Kleis and X. Zhou. A placement scheme for peer-to-peer networks based on principles from geometry. In *Proceedings of the IEEE Fourth International Conference on Peer-to-Peer Computing (P2P'04)*, pages 134–141, 2004.

[10] H. Lim, J. C. Hou, and C.-H. Choi. Constructing internet coordinate system based on delay measurement. In *Proceedings of the ACM IMC*, 2003.

[11] E. K. Lua, T. Griffin, M. Pias, H. Zheng, and J. Crowcroft. On the accuracy of embeddings for internet coordinate systems. In *Proceedings of the ACM IMC*, Berkeley,CA, October 2005.

[12] Y. Mao and L. K. Saul. Modeling distance in large-scale networks by matrix factorization. In *Proceedings of ACM IMC*, October 2004.

[13] S. Ng and H. Zhang. Predicting internet network distance with coordinates-based approaches. In *Proceedings of the IEEE INFOCOM*, 2002.

[14] M. Pias, J. Crowcroft, S. Wilbur, T. Harris, and S. Bhatti. Lighthouses for scalable distributed location. In *Proceedings of the IPTPS*, 2003.

[15] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content addressable network. In *Proceedings of ACM SIGCOMM*, 2001.

[16] Y. Shavitt and T. Tankel. Big-bang simulation for embedding network distances in euclidean space. In *Proceedings of the IEEE INFOCOM*, San Francisco, California, April 2003.

[17] L. Tang and M. Crovella. Virtual landmarks for the internet. In *Proceedings of the ACM IMC*, Miami Beach, FL, October 2003.

[18] B. Wong, A. Slivkins, and E. Sirer. Meridian: A lightweight network location service without virtual coordinates. In *Proceedings of the ACM SIGCOMM*, Philadelphia, PA, August 2005.

[19] Z. Xu, P. Sharma, S.-J. Lee, and S. Banerjee. Netvigator: Scalable network proximity estimation. Technical Report HPL-2004-28R1, HP Laboratories, March 2005.

[20] E. W. Zegura, K. Calvert, and S. Bhattacharjee. How to model an internetwork. In *Proceedings of IEEE Infocom*, 1996.