PlanetLab Deployment, Comparison and Analysis of Network Coordinate Systems

Fenglin Liao, Keshava P Subramanya, Veljko Pejovic, Ben Zhao

{fenglin, keshava, veljko, ravenben} @ cs.ucsb.edu

Abstract - In a short time, network coordinate services have advanced to being useful and reliable tools for network-distance aware systems. A number of large scale distributed applications can enhance their performance by using NC systems to replace extensive RTT measurements with a lightweight estimation of network distances.

Traditional criteria used for NC systems comparison, such as deviation of the estimated network distance from the actual one, prove to be insufficient as a deciding parameter for the system selection in many cases. Thus, we introduce the other criteria that would depict the overall NC system performance in three categories defined: accuracy, robustness and stability. Our main goal is to determine a set of measurements that should be performed on every future NC system.

As an illustration, we provide in depth experimental analysis of two essentially different NC systems: GNP and Vivaldi. A comparative study is performed on more than 100 PlanetLab machines running GNP and Vivaldi concurrently. Our experiments show that each approach outperforms the other in some of the measurements, therefore leaving "the most appropriate NC system" decision solely to a specific application and its needs.

Finally, based on derived results and structural analysis of Vivaldi and GNP, we provide guidelines for future improvements of NC systems design and implementation.

Categories and Subject Descriptors

C.2.1 [**Computer Communication Networks**]: Network Architecture and Design—*Distributed networks*
C.4. [**Performance of Systems**]:
Performance attributes

General Terms

Performance, Measurement, Design, Experimentation

Keywords

Network coordinates, Vivaldi, GNP, Comparison criteria

1. Introduction

Network coordinate systems can enhance the performance of many large scale distributed applications relying on the accurate network distance (i.e, round-trip transmission and propagation delay) information. The traditional approach in which participants probe each other in order to collect network distance data, is proved to be unscalable and cumbersome especially for dynamic, peer-to-peer networks. A well designed NC system provides the above-lying service, a possibility to estimate the remoteness to some other point in the network without need for extensive pairwise RTT measurements. Several NC systems were developed so far, the first major one being GNP[1], a landmark based network distance predictor. The following work in this field focused mainly on scalability and distribution, abandoning the landmarks idea and resulting in the following notable achievements: Vivaldi[3], adaptive, distributed system, and PIC[2], which maintains a level of reliability even in malicious surroundings.

As a number of available NC solutions grows, a representative way of selecting the most appropriate one, for a certain use, is needed. Since every NC system results in some form of coordinate system with a defined distance function, conventional forms of evaluation such as relative distance error seem the most appropriate. Comparing NC systems by this manner provides a relatively good accuracy categorization. Still, numerous applications heavily depend on other properties: system convergence time, reliability of estimation when nodes are leaving/joining the system, accuracy in presence of malicious nodes, and others.

In this study we concentrated on defining the set of criteria that should be taken into consideration when comparing NC systems. Requirements imposed by different large-scale applications directed us towards the following classification of important NC system properties:

- Accuracy
- Robustness
- Stability

An exhaustive discussion on selecting these criteria is given in § 3 along with explanation of measurements required to satisfy each of them.

In order to exemplify the relevance of the criteria stated above, we conducted a comprehensive comparative study of two approaches for network coordinate estimation – Vivaldi and GNP. The experimental setup (described in § 4) consists of more than 100 PlanetLab machines concurrently running both NC systems.

The results of the experiments, stated and analyzed in § 5, along with white-box examination of Vivaldi and GNP, supported our conclusions and guidelines for future work introduced in § 6.

2. Related work

Since the network coordinates concept represents a relatively new paradigm, with various and ever changing ways of implementation (even "traditional" approaches like GNP appear in new incarnations [7]) no exhaustive comparative study of existing NC systems has yet been performed.

Authors of the new NC system usually compare its accuracy with one of GNP, therefore [3] compares Vivaldi and GNP in two different environments: PlanetLab and a number of Internet DNS servers. Similarly, [2] provides experimental comparison of PIC and GNP.

Ledlie et. al. [4] show that the anticipated behavior of Vivaldi on Planetlab differs from its behavior in the "wild", real life applications. Nevertheless, a set of simple techniques can be applied in order to enhance the accuracy of "wild" Vivaldi.

Probably the most comprehensive analysis in this field was done by Lua et.al.[6]. It introduces two important new metrics – Relative Rank Loss and Closest Neighbor Loss, which will be explained in § 3 and used in § 5 of this paper.

However, all aforementioned measurements concentrate on a specific attribute of NC systems – accuracy, while we strive to define a broader set of criteria that should be concerned when comparing systems.

3. Methodology

Observing potential users of network coordinates estimation, we found several colliding demands: coordinates should be accurate, system should be lightweight, performing as little probing as possible, false information will not affect behavior of correct nodes, as well as others. Without any doubt, these can be classified in the following categories: accuracy, robustness and stability. This classification, by no means finite and definite, concentrates on requirements established by current applications.

3.1. Accuracy

Simply defined as the difference between RTT gathered between nodes and the data calculated by the given NC system, accuracy is the most common way of representing system quality. Defined this way, accuracy in some usages can be seen as a confidence factor in a whole system, e.g. VoIP applications may require a contact to a node that is located on a certain distance from the request originator.
However, other applications may require only a knowledge of a relation between distances of two nodes (e.g, Which is closer – A, B?) or information on the nearest neighbor. Obviously, slight deviation from the real RTT now does not matter, additionally, providing only this, necessary data, simplifies the system API. Defined in [4], Relative Rank Loss (RRL) and Closest Neighbor Loss (CNL) is measured in order to evaluate these properties.
Another aspect of accuracy, effect of node distance, can affect overall performance. Whether a person residing on one continent can estimate the difference between distances on the other, qualifies the system for continent or world scale usage.

Finally, the accuracy of estimation may change over time and represents another aspect to be considered. Protocols like Vivaldi need a certain time to converge to a level of accuracy which is acceptable for proper application. If the convergence period exceeds application lifetime, designated accuracy will not be achieved.

3.2 Robustness

Under this category we file measurements concentrated on system behavior under heavy exterior impact. We are mostly interested in measurements of the system performance under extreme network traffic and in the presence of malicious nodes, since it is seen as a most common robustness problem in modern peer-to-peer applications – the sudden appearance of an interesting content initiates file transfer bursts, while hacker attacks may aim NC system accuracy.

3.3 Stability

Deployment of centralized systems such as GNP requires a set of reliable nodes – landmarks; Vivaldi's spring like behavior is seriously disrupted when network changes happen. The effect of nodes joining and leaving the system matters in a real world scenario where machines go up and down. Therefore the analysis of performance under these changes is suggested.

4. Experimental Setup

Our experiments were performed on 100+ PlanetLab nodes. In the selection process, we designated one node at UCSB as a coordinator and performed SSH and ping tests to all other available nodes (466 total). We sorted all reachable nodes by the ping times from the coordinator node, and selected a decreasing number of representatives from every group (RTT interval) so that 104 machines started the test. Throughout the experiments, several nodes experienced long-term crashes, leaving us with 99 complete sets of data.

Every node ran Vivaldi in the form of Pyxida[12] - Java based open source network coordinate library and application (the suggested [3] height value was used in addition to standard 4 dimensions); and eight-dimensional GNP based on [13]. Original source codes were modified for certain observations (the effect of malicious nodes)

5. Comparative Study

In the first set of the experiments, accuracy, in all aspects described in § 3.1, is measured and compared for both GNP and Vivaldi. CDF in *Figure 1* shows that GNP outperforms Vivaldi, experiencing more than 90% of nodes with less than 0.5 relative error, while in Vivaldi's case that number is around 85%. Our endeavors with additional dimensionality slightly improved Vivald's accuracy, but not enough to top GNP.

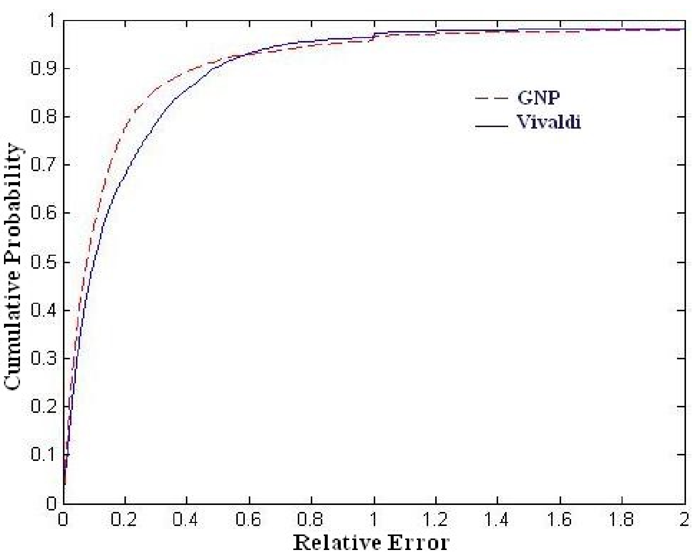

Figure 1: The cumulative distribution of relative estimation errors

Another important observation follows from the above figure: examined NC systems are still very inaccurate: more than 40% of Vivaldi and 30% of GNP measurements resulted in relative error greater than 0.2

As stated earlier, relative rank loss (RRL) and the closest neighbor loss (CNL) are more relevant deciding factors in some applications. RRL is measured as a ratio between the number of incorrectly estimated distance relations and the total number of inter-node relations. CNL represents a percentage of nodes whose estimated closest neighbor differs from the actual one. A median percentage of relative rank loss is around 11% in both systems (*Table1);* still, their behavior reassembles on an individual node basis.

NC system	min RRL	avg.RRL	max RRL	CNL [%]
GNP	0.0429	0.1154	0.7908	50.5
Vivaldi	0.0608	0.1095	0.3232	57

Table 1: Relative Rank Loss (RRL) and Closest Neighbor Loss (CNL) for GNP and Vivaldi

We observe varying RRL throughout the experiment (*Figure 2*). In GNP's context, several nodes demonstrated RRL of 0.5 or more, while no such occurrence was noted in Vivaldi. Moreover, nodes with significant RRL in one system do not necessary display the same fallacy in the other. Although the most erroneous GNP nodes are the ones with a relatively high number, thus being far from coordinator, and isolated in their group, data gathered are not sufficient to conclude that GNP's RRL related performance drops with a network distance.

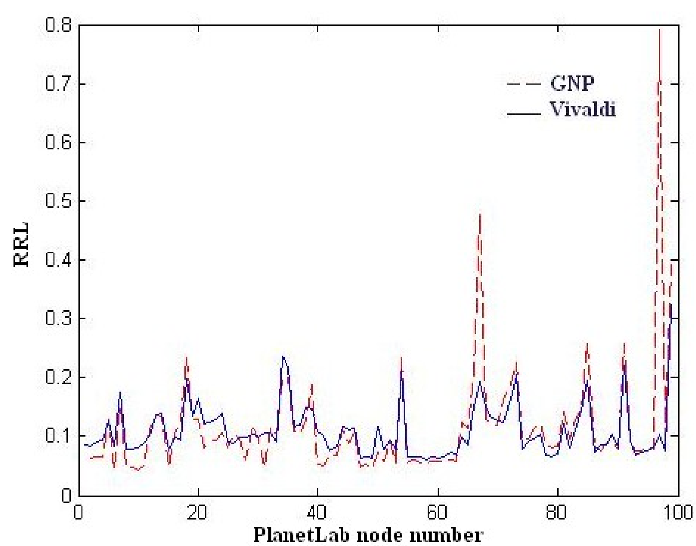

Figure 2: Distribution of RRL(observed per node) on 100 nodes. Nodes are sorted by the distance from the coordinator

Therefore, further analysis is conducted. We fully examine the impact of distance on estimation accuracy: each PlanetLab node is positioned in the origin of the coordinate system once, and the prediction error towards other nodes is plotted. In *Figure 3* Vivaldi's tendency to overestimate closer nodes (<100ms) is shown by high density of blue Xs directly above the zero value of relative directional error, while starting from distance of 200ms GNP makes underestimation errors (red dots below X axis).

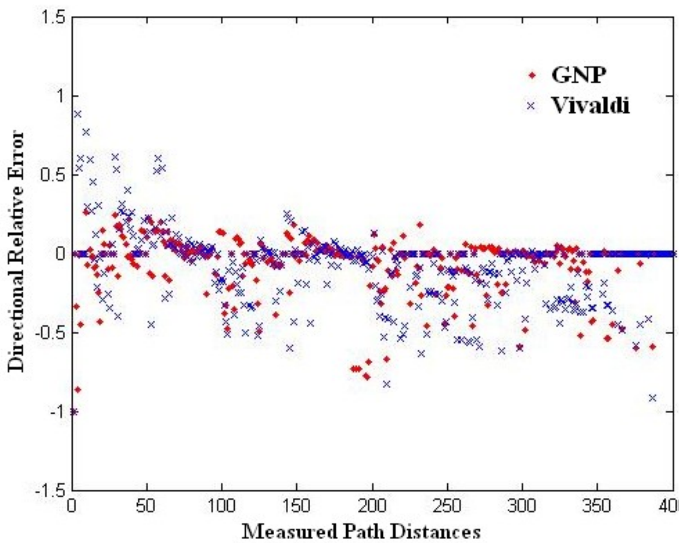

Figure 3: Directional Relative Error

Another important side of accuracy is its time resistance. The static nature of GNP code did not allow us to measure convergence behavior; therefore only Vivaldi data are plotted on *Figure 4*. The moment of convergence is defined as a point of time when a relative estimation error fell under 0.2. It is concluded, directly from the figure, that after 2.5 hours all nodes in Vivaldi coordinates reach significant coordinate reliability.

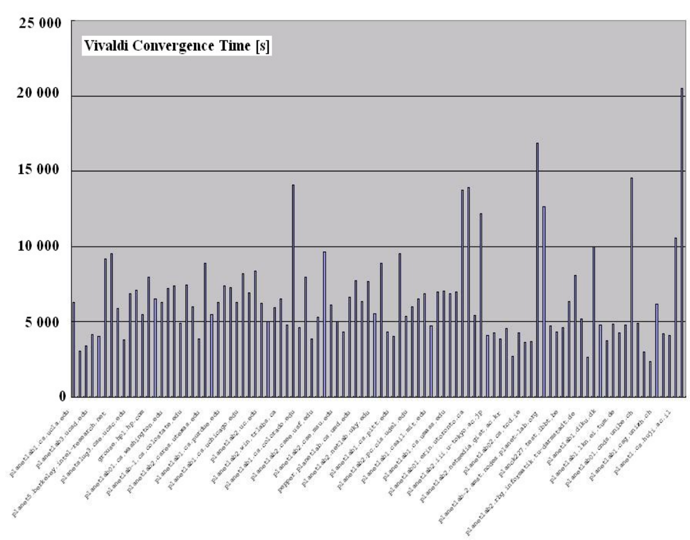

Figure 4: Vivaldi: Convergence time per each PlanetLab node

Our following set of experiments is aimed at systems' robustness. In the first tryout we isolated a set of eight nodes and exposed them to artificial heavy traffic. More than 3 000 varying network connections, each with 3MB payload were generated by specialized software [14] We examined the performance of Vivaldi and GNP and found no significant impact of bursting network traffic. Having in mind that GNP is fed by RTTs gathered through ICMP messages, possible congestion resulted from UDP/TCP flows does not have a significant effect on GNP. Up to this time, we do not know why Vivaldi is not affected by heavy network traffic.

In the second experiment in this section, we again isolated eight stable nodes (a four hour period was given to Vivaldi to converge) and introduced three malicious nodes. These malicious nodes supplied Vivaldi with false coordinates determined to “pull” all systems coordinates as far as possible. Since coordinates estimation in GNP is based on plain landmark pinging, rather then mutual coordinate exchange as in Vivaldi, malicious nodes, designed here, do not affect GNP's behavior. Although, spoofed ICMP messages (e.g. “smurf attack”) may result in GNP's loss of accuracy. This, however, is not within the scope of our research.

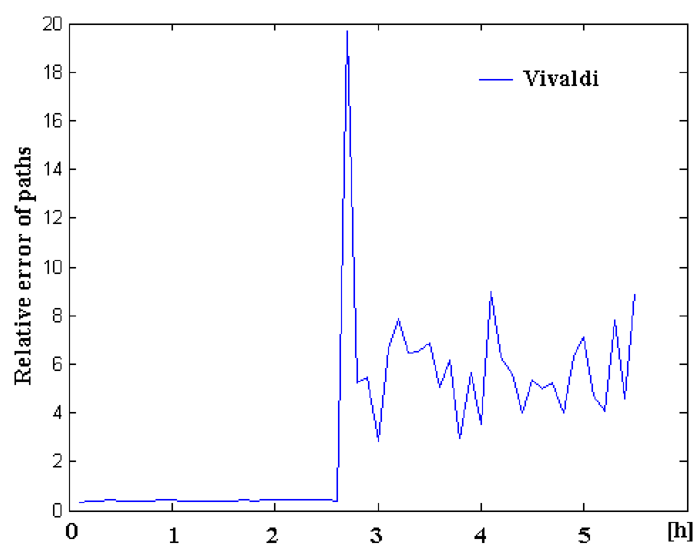

Figure 5: The effect of malicious nodes on Vivaldi. In t=2.5h three malicious nodes joined the system

As can be seen *(Figure 5)*, these malicious nodes largely affected correct ones, creating an average relative error peak to 20, allowing it to stabilize around 6 after some time. Apparently, Vivaldi's spring-like structure was “stretched” by these “pulling” nodes.

The stability of that spring-like architecture is largely influenced by machines that are joining and leaving the network. Further, GNP's distance estimation may be disrupted by failing landmarks. We conducted an experiment where in the above isolated topology three nodes left the system and rejoined it after a period of time (intended to test system stabilization).*Figure 6* illustrates the change in *the NC system confidence factor (CF)*– the percentage of paths that have relative error of 0.2 or less. It is observed that *CF* increases as Vivaldi is starting and suddenly drops when the three nodes leave the system.

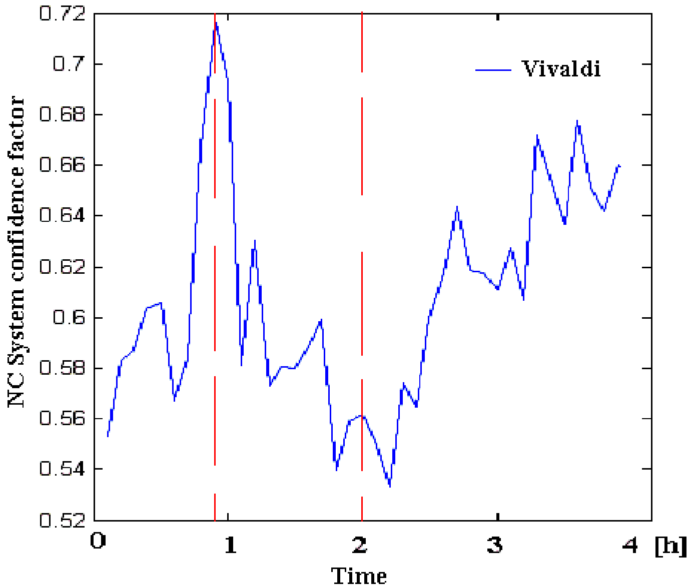

Figure 6: Confidence factor when nodes leave/join the system

Another drop occurs when the three nodes rejoined the system. Afterwards, the confidence factor slowly rises. We conclude that Vivaldi's convergence time prevents it from adopting to network changes; furthermore, nodes leaving the system have a higher impact on stability than nodes joining the system.

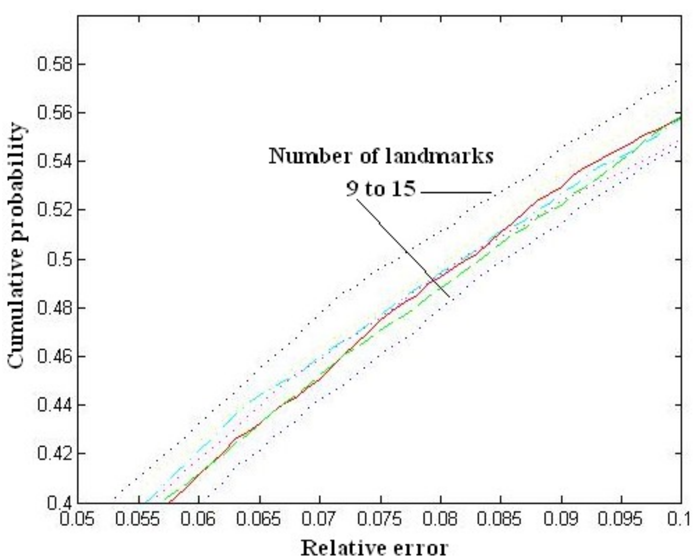

Figure 7: Cumulative probability of relative estimation error; different number of GNP landmarks

When discussing stability, "ordinary" GNP nodes obviously have no influence *(Figure 7)* unless they are landmark nodes. However, even when landmark nodes fail down, estimation accuracy level is almost unchanged. Additionally, a set of nodes *(candidates)* is kept as a "backup landmarks" in case of original landmarks break down. Summarizing, GNP landmarks either give reliable results (when landmarks are leaving/joining the network but a minimum number of landmarks is always present in the system) or they provide no result at all (if a number of landmarks is fewer then a number of dimensions)

6. Conclusions and Guidelines for Future Work

In this work we have represent our view on NC system comparison. We defined three major criteria we believe should be examined when evaluating NC systems: accuracy, robustness and stability. Our experimental review of GNP and Vivaldi was conducted as an example of how these criteria can be measured.

Based on our measurements we can say that GNP is generally more accurate but both protocols are far from perfect: CNL performance of both systems are highly unreliable - more than 50% of closest neighbor estimations were inaccurate in both NC systems. Another accuracy measurement, RRL, revealed that some nodes in GNP tend to make considerably more relative rank estimation errors, making GNP's per-node behavior unpredictable. A deeper research showed that incorrect estimations tend to occur when predicting short distances, in Vivaldi, and long distances, in GNP. Convergence time of Vivaldi took 2.5 hours, while GNP can be set almost immediately.

A bright point of NC protocols was shown when neither one had problems when dealing with bursting network traffic. However malicious nodes, especially in Vivaldi, can make the system almost unusable.

We do not see problems of nodes joining and leaving the network in GNP, unless the number of landmarks falls down under the number of dimensions used; Vivaldi's delivery, on the other hand, is influenced with nodes joining, and even more nodes leaving the system.

One more important, real-world restriction on protocols should be considered – how scalable the NC system is. Surely implementing GNP on 100+ PlanetLab network is not a serious burden on 9 to 16 landmark nodes, but we doubt that a popular peer-to-peer network with hundreds of thousands of users can rely on such a few nodes. Vivaldi on the other side promises easy deployment, and similar characteristics even in a huge network like Azureus [11].

Finally, we can conclude that GNP is more appropriate for small, denser networks with several reliable clients that can be designated as landmarks, while Vivaldi aims towards large scale networks with long lifetime expectancy.

Guidelines for Future Work

Following a path of an ongoing trend toward scalability and distribution, future NC systems will have to adopt to expected large-scale distributed deployment. However, some aspects of centralized systems, may be kept to improve accuracy, producing somewhat hybrid solutions. Either periodic random pinging of certain "landmarks" or running several systems at the same time (relying on the most reliable at a certain moment) would enhance the already distributed system.

Our experiences with current Java based implementations of the two systems showed that the average CPU usage by an NC system was around 4% (on 100+ PlanetLab nodes testbed) Since the real world applications expect an NC system to be available at all times, and to operate under various conditions, even this low percentage can represent a burden. As they get more ubiquitous, systems should be more lightweight, thus realized in some other programing language like C. Together with simple, open source API, this can raise NC system popularity and create new applications for it.

As concerned with stability issues, we propose keeping the log data which would contain the latest coordinates a node was assigned with. Now, even if a node fails down, on recovery it can easy obtain the latest "live" coordinates instead of starting the convergence process from scratch. Malicious nodes represent a serious threat, and an NC system should guard itself by using a sort of reputation mechanism.

In the end, wireless, especially mobile, nodes tend to rapidly change their "distance" from the rest of the network, and may result in overall estimation inaccuracy. Therefore we suggest observing them as read-only participants, who may find out information on the other nodes in the network, but do not affect distance estimation themselves.

References

[1] S. Banerjee, M. Pias, T. Griffin, The Interdomain Connectivity of PlanetLab Nodes, *Passive and Active Measurements (PAM) Workshop*, Sophia-Antipolis, France, April 2004

[2] M. Costa, M. Castro, et al. PIC: Practical Internet Coordinates for Distance Estimation. In *ICDCS*, March 2004.

[3] F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi: A Decentralized Network Coordinate System. In *SIGCOMM*, August 2004.

[4] Jonathan Ledlie, Paul Gardner, and Margo Seltzer, Network Coordinates in the Wild, *Submitted for publication, Harvard University Computer Science Technical Report TR-20-06*, October 2006

[5] J. Ledlie, P. Pietzuch, and M. Seltzer, Stable and Accurate Network Coordinates, In Proceedings of *ICDCS* 2006, Lisbon, Portugal, July 2006

[6] E. K. Lua, T. Griffin, et al. On the Accuracy of Embeddings for Internet Coordinate Systems. In *IMC*, October 2005

[7] T. S. Eugene Ng and Hui Zhang, "A Network Positioning System for the Internet", *USENIX Annual Technical Conference 2004*, Boston, MA, June 2004

[8] T. S. E. Ng and H. Zhang. Predicting Internet network distance with coordinates-based approaches. In *Proceedings of IEEE Infocom,* pages 170–179, 2002.

[9] Y. Shavitt and T. Tankel. Big-Bang Simulation for embedding network distances in Euclidean space. In Proc. of *INFOCOM*, San Francisco, CA, June 2003.

[10] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. In Proc. of *SIGCOMM*, San Diego, CA, Aug. 2001.

[11] Azureus BitTorrent Client http://azureus.sourceforce.net/

[12] Ledlie, Pietzuch, Parker, http://pyxida.sourceforge.net/

[13] T. S. E. Ng, http://www.cs.rice.edu/~eugeneng/research/gnp/

[14] Network Traffic Generator: http://sourceforge.net/projects/traffic