Experimental Study on Neighbor Selection Policy for Phoenix Network Coordinate System

Gang Wang, Shining Wu, Guodong Wang, Beixing Deng, Xing Li
Tsinghua National Laboratory for Information Science and Technology
Department of Electronic Engineering, Tsinghua University, Beijing 100084, P.R.China
g-wang06@mails.tsinghua.edu.cn, wsn06@mails.tsinghua.edu.cn

***Abstract*—Network Coordinate (NC) which provides efficient distance prediction with scalable measurements, brings benefits to various network applications. Phoenix Network Coordinate system is a recently proposed dot product based NC system with high prediction accuracy and better robustness. In this paper, we discuss the neighbor selection policies for Phoenix and propose a modified Phoenix with Hybrid neighbor selection policy. Extensive experiments have been carried out to evaluate the performance of Phoenix NC systems with different neighbor selection policies. The results show that the system with Hybrid policy achieves lower relative errors (REs) and obtains a higher accuracy in selecting the nearest neighbor host. Moreover, the Hybrid policy based Phoenix NC system has a better performance in the application of Overlay Multicast. It reduces the tree cost of the ESM and MST multicast tree construction by at least 20%.**

I. Introduction

Network Coordinate which is designed to predict distances (Round Trip Time, RTT) among hosts in the network, has been proved to be efficient and practical. For instance, the performance of peer-to-peer network will be greatly improved as the nearest host is selected by computing the coordinates instead of probing messages through the network [12]. Some large scale distributed systems, such as DHT, also benefit from the distance prediction of NC system. In NC systems, we can predict $O(N^2)$ RTTs by measuring $O(N)$ RTTs among N nodes. The overhead of the distance prediction can be significantly reduced and the accuracy, on the other hand, has been proved to be relatively high.

A new NC system named Phoenix is proposed recently. Contrast to other Euclidean space based NC systems such as PIC [3] and Vivaldi [7], Phoenix is a dot product based NC system and avoids the problem of Triangle Inequality Violation (TIV) [9]–[11]. It thus achieves higher distance prediction accuracy. Moreover, a weight calculation algorithm has been adopted in Phoenix to reduce the bad effects of error propagation. Phoenix also obtains a relatively high convergence rate and better robustness over measurement anomalies, which make Phoenix an accurate and practical NC system. [2]

In Phoenix NC mechanism, a list of reference hosts is required when a new node joins the overlay. The currently proposed Phoenix NC system employs a random policy to select reference neighbors. However, it suffers disadvantages in certain applications, such as overlay multicast [8]. In addition, how different neighbor selection policies affect the performance of Phoenix NC system has not been discussed in [2]. In this paper, we will focus on neighbor selection policy for Phoenix NC system. Besides the Random selection policy, two other neighbor selection policies: Hybrid neighbor selection and Closet neighbor selection will be studied. The accuracy of Phoenix with different neighbor selection policies will be evaluated using the most famous metric called relative prediction error.

Considering Phoenix is a dot product based NC system, the evaluation of neighbor selection policy should be quite different from that for PIC which is base on the Euclidean space [3]. Moreover, we will evaluate the performance of each policy in overlay multicast, a representative application in the Internet. The accuracy of the predicted distances will be evaluated by reporting the cost of constructing overly multicast trees such as Minimum Spanning Tree (MST) and End System Multicast Tree (ESM). [6]

Finally, based on various experiments, we propose a modified Phoenix NC system with Hybrid neighbor selection policy. The results of the experiments show that the new system achieves lower relative prediction errors over different data sets and has better performance in overlay multicast. It reduces the tree cost of multicast tree construction by at least 20% compared with Phoenix with random neighbor selection policy. The results present an improvement in the ability to select the shortest links.

The rest of the paper is organized as follows: In Section II, we describe the Phoenix NC systems with three different neighbor selection policies. In Section III and Section IV the accuracy of distance prediction by the three NC systems will be evaluated. We present some related work in Section V and the paper will be concluded in Section VI.

II. System Design

Phoenix maps each node to a d-dimensional incoming vector and a d-dimensional outgoing vector. The predicted distance from node A to node B is the dot product of the outgoing vector of A and the incoming vector of B. When a new node, for example, m, joins the system, it will pick up M nodes as reference hosts from a set of N nodes whose NC have already been calculated ($M < N$). Then the NC of m will be computed and updated per round. When $N < M$, all the N nodes gain a $N \times N$ distance Matrix by probing each other and their coordination will be computed by Non-negative Matrix Factorization (NMF) [1] method.

Neighbor Selection policy has great impact on the accuracy and overall overhead of the NC system. Here we propose three neighbor selection policies. The first one is the Random policy (Fig.1): Randomly select M nodes as reference hosts. Secondly, we propose the Hybrid policy (Fig.2): Pick up some nodes randomly while others are the closest nodes. Thus, according to Hybrid policy, M will be divided into two parts: Mc is a set of nodes in N which are closest to m and the rest part, Mr, will be randomly selected in the network. The last one is the Closest policy (Fig.3): Select M closest nodes to m as its neighbors to compute the NC; The neighbor selection policy in the current version of Phoenix NC system [2] is the Random.

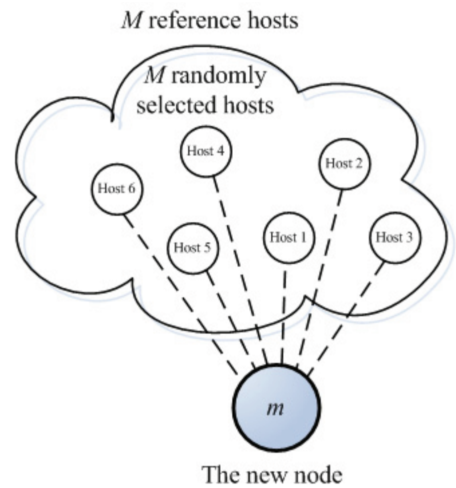

Fig. 1. The Random neighbor selection policy

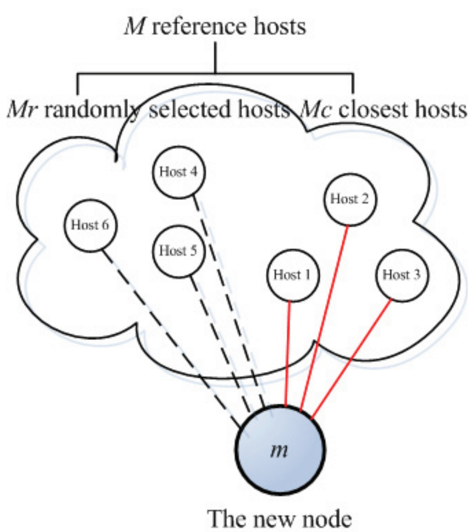

Fig. 2. The Hybrid neighbor selection policy

In the following subsection, we will apply these three neighbor selection policies to Phoenix system and evaluate their

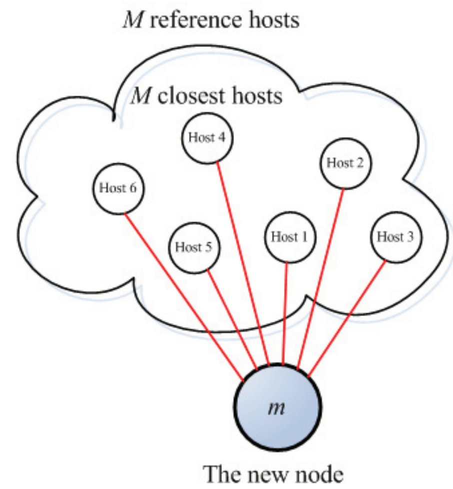

Fig. 3. The Closest neighbor selection policy

performance. In the Closest and Hybrid policies, a mechanism to find the closest node is required. To focus on the evaluation of neighbor selection policy and ensure validity and equality, we introduce an Oracle which is supposed to have a general knowledge of the distances among all the nodes in the whole network to find the closest neighbors. Practically, this Oracle can be implemented by several algorithms [3], [13], [14] which have been proved to be quite efficient with their accuracy and overhead considered.

III. Experimental Setup

In the following experiment, we will evaluate the performance of Phoenix NC systems with three different neighbor selection policies. PhoenixSimNs [17] is our simulator.

The experiment is based on both the PlanetLab [4] data set and the King [5] data set. The PlanetLab data set contains the RTTs among 226 hosts all over the earth and the King data set includes the RTTs among 1740 Internet DNS servers.

The three Phoenix systems will be set equally with 10-dimension coordinates and each node has 32 reference neighbors. The coordinates of newly joined node are supposed to converge within 10 rounds [2] and in our experiment, the number of rounds will be set 30 to ensure convergence. The test runs 10 rounds on each data set and an average result will be reported.

IV. Performance Evaluation

A. Prediction Accuracy

In this section, we evaluate the accuracy of Phoenix System with three different neighbor selection policies. The prediction accuracy is measured by relative prediction error.

Table.I shows the results of the three NC systems over the PlanetLab data set and the King data set. The Hybrid policy uses 6 closest neighbors and 26 randomly selected neighbors. And the mean relative error is reported. The result illustrates that Phoenix System with the Closest policy performs significantly worse than those with Hybrid and Random.

So we just move on with the other two polices. Hybrid, according to Table.I, achieves a lower mean relative error than Random by 41.7% over the PlanetLab data set. The experiment conducted over the King data set has similar results. The Hybrid policy reduces the mean relative error by 35%. So the Hybrid achieves better accuracy on distances prediction than the Random one.

TABLE I
MEAN RE OF THREE NEIGHBOR SELECTION POLICIES

Policy	PlanetLab	King
Random	0.2363	0.2416
Hybrid	0.1377	0.1567
Closest	1.6548	0.8791

Based on this result, we have an intuition. Phoenix NC system is able to predict most of the distances among the hosts with high accuracy. But there are a small amount of predicted distances suffering serious errors which bring confusion to NC system when selecting the closet hosts. Hybrid policy, with lower mean relative error, should improve the ability of NC system to select the nearest host.

To confirm this intuition, we introduce another metric: Nearest Neighbor Loss (NNL) to evaluate the performance of the three Phoenix NC systems. NNL is defended to be the difference between the latency to the node estimated to be the nearest neighbor by NC and the latency to the true one [8]. In our experiment, an average result is reported in Table.II. The result shows that Phoenix, with Hybrid neighbor selection policy achieves lower NNL than those with Random and Closest over both the PlanetLab and the King. This reflects a higher accuracy of Hybrid based Phoenix in selecting the nearest neighbor host within the whole network.

TABLE II
NLL OF THREE NEIGHBOR SELECTION POLICIES

Policy	PlanetLab	King
Random	21.085	20.8871
Hybrid	13.4995	14.8103
Closest	112.3941	53.5009

B. Application on Overlay Multicast

In this section, we evaluate the performance of three neighbor selection policies by applying Phoenix to Overly multicasting [15], [16]. The multicast tree will be constructed under the direction of predicted distances. The prediction accuracy will be evaluated by measuring the cost of building a tree.

To focus on the impact of the three neighbor selection policies on distance prediction, here we introduce two kinds of abstract overlay multicast trees: MST and ESM. MST is constructed based on the Prime Algorithm. The key point of building a good multicast tree is to select the shortest links. As for ESM, for instance, if a new node m is to join the ESM tree, it will get a list of nodes which have been already on

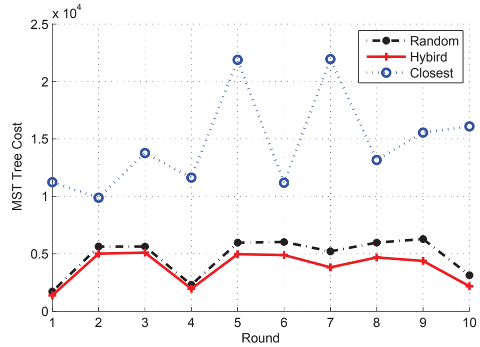

Fig. 4. The Tree cost (MST) based on the PlanetLab data set

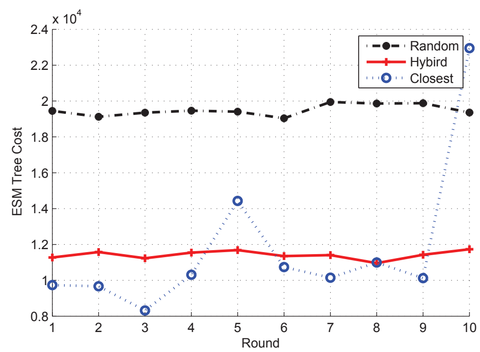

Fig. 5. The Tree cost (ESM) based on the PlanetLab data set

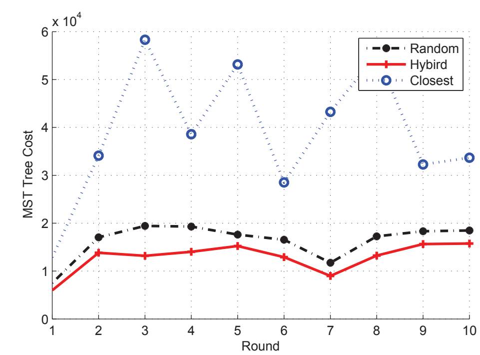

Fig. 6. The Tree cost (MST) based on the King data set

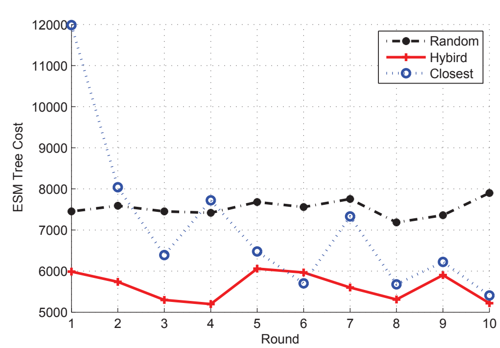

Fig. 7. The Tree cost (ESM) based on the King data set

the tree. Then m will choose the closest node in the list as its parent and this process will be repeated. In our experiment the list includes 30 randomly chosen nodes. [6]

Instead of studying specific overlay multicast protocols and algorithms; we will operate our experiment on these two abstract models. We use tree cost to measure how well the overlay multicast tree is constructed, which is defined as the sum of latencies of all tree links. The tree cost reflects the ability to select the shortest link which is tightly related to the accuracy of distances prediction by Phoenix NC system.

In the experiment, we build the overlay trees according to the predicted distances provided by Phoenix NC systems with different neighbor selection policies: Random, Hybrid and Closest. The setup for Phoenix NC system remains the same as IV.A. The tree construction process is repeated for 10 times and an average tree cost is reported. And we will follow the results of 10 runs.

TABLE III
TREE COST SUMMARIES

Tree type	Policy	PlanetLab	King
MST	Random	4792.1	16319
MST	Hybrid	3842.4	12874
MST	Closest	14635.0	38984
ESM	Random	7779.6	19487
ESM	Hybrid	5742.8	11419
ESM	Closest	7095.2	11741

Table.III shows the summary of average tree cost over the PlanetLab and the King. The Hybrid neighbor selection policy leads to the lowest tree cost in both MST algorithm and ESM algorithm. In ESM, the Hybrid version reduces the tree cost by 26.18% over the PlanetLab data set and 41.4% over the King data set compared with the current Phoenix system with Random neighbor selection policy; In MST, the Hybrid version reduces the tree cost by 20% over the PlanetLab and 21.11% over the King. On the other hand, the Hybrid policy performs better than the Closest strategy as well. Details of each run are displayed in the following figures Fig.4 to Fig.7.

Fig.4 and Fig.5 shows the MST and ESM tree cost of each run based on the PlanetLab data set. The tree size is 266 (including all the nodes). The figures illustrate that the hybrid policy obtains a lower tree cost in almost each run compared to the others. Similarly the result of the experiment over the King data set is shown in Fig.6 and Fig.7. The tree size is 500.

In addition, we study how the tree cost change as the tree size increases over the King data set. Fig.8 depicts the impact that different tree size has on the average MST tree cost and Fig.9 is the result of ESM algorithm. The two figures illustrate that for both MST and ESM algorithm, the Hybrid policy obtains a relatively lower growth rate of tree cost as the tree size increases. This result indicates that Hybrid policy has an advantage over Random and Closest in overlay multicast tree construction and its advantage will not receive serious damage as the tree size becomes large.

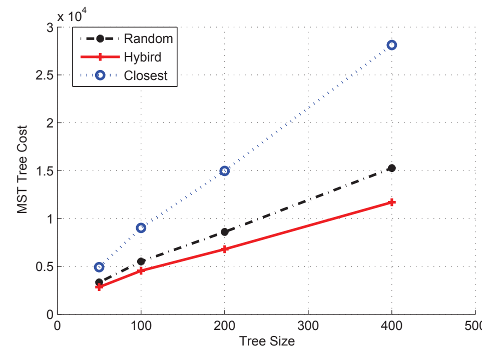

Fig. 8. The Tree cost (MST) over King

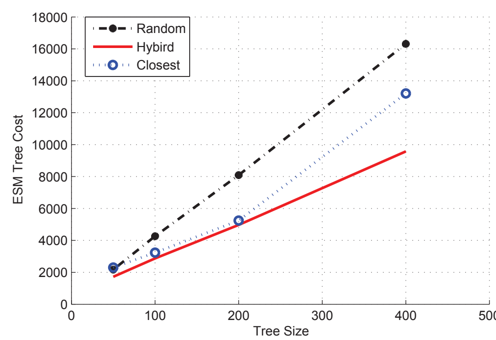

Fig. 9. The Tree cost (ESM) over King

Based on the experiment, the Hybrid based Phoenix NC system is quite efficient in multicast tree construction. This result will in turn confirm the conclusion that the Hybrid policy achieves higher accuracy in distances prediction.

V. RELATED WORK

A. *Network Coordinate (NC) system*

Network Coordinate system is an efficient and scalable tool to predict distances among hosts in the network. In most of the NC systems, such as Vivaldi [5] and PIC [3], the internet is mapped to an Euclidean space and each host has a coordinate to represent its location in the geometric space. The distances between two hosts thus could be estimated by computing their coordinates using Euclidean distance. The overhead of distances prediction will be significantly reduced compared with full mesh probing in the real network.

NC systems bring great benefits to various distributed network applications. However, the Triangle Inequality Violations (TIVs) has been found to exist in most of the Euclidean space based NC systems. Distances in Euclidean space have to satisfy the triangle inequality but in real network, RTTs among three hosts often go against it. This constrains the accuracy of these NC systems.

Dot product based NC systems, such as IDES [4] and Phoenix [2], have been proposed in recent years. In these NC systems, a large distance matrix could be factorized into two smaller matrixes by methods like Singular Value

Decomposition (SVD) or Non-negative Matrix Factorization (NMF) [1]. The distances then could be defined as the dot product of two vectors, which are simply the row or the column of the smaller matrixes. Since the distances predicted by this system don't have to obey the triangle inequality, the dot product based NC systems can avoid the problem of TIV and get higher accuracy.

Our work focus on dot product based NC system, and the evaluation of neighbor selection policy is different from that of Euclidean space based NC system, such as PIC.

B. Dot product based NC and relative error

Suppose there are N hosts in an overlay network. Define $\boldsymbol{D}$ as the $N \times N$ distance matrix among the hosts. $\boldsymbol{D}(i,j)$ indicates the measured RTT between host i and host j. Matrix $\boldsymbol{D}$ can be factorized into two matrixes. $\boldsymbol{D} \approx \boldsymbol{A}\boldsymbol{B}^{\boldsymbol{T}}$ where $\boldsymbol{A}$ and $\boldsymbol{B}$ are $N \times d$ matrixes($d \ll N$).

In dot product based NC system, let $\boldsymbol{A}_i$ be the outgoing vector and $\boldsymbol{B}_i$ be the incoming vector corresponding to host i. The predicted distance from host i to host j is simply the dot product of the outgoing vector of host i and the incoming vector of host j. Just as follows: [2]

$$\boldsymbol{D}^{\boldsymbol{P}}(i,j) = \boldsymbol{A}_{\boldsymbol{i}} \cdot \boldsymbol{B}_{\boldsymbol{j}} = \sum_{n=1}^{d} \boldsymbol{A}(i,n) \cdot \boldsymbol{B}(j,n)$$

The accuracy of estimation of an NC system is often indicated by relative error which is a rate between the predicted distance and the real measured RTT. The definition of relative error [2], [8] of the distance between host i and host j is as follows:

$$\boldsymbol{RE} = \frac{|\boldsymbol{D}^{\boldsymbol{P}}(i,j) - \boldsymbol{D}(i,j)|}{\boldsymbol{D}(i,j)}$$

An NC system demonstrates higher estimation accuracy by showing smaller relative error. If the predicted distance is of absolute accuracy, i.e., the predicted distance agrees with the real measured RTT completely, the relative error would be zero.

C. Phoenix

Phoenix is a promising NC system based on dot product. Compared with IDES, it has several improvements [2]. First, the negative predicted distance can be totally avoided so that the performance can correspond with RTT's physical meaning and be beneficial to practical applications. Moreover, the propagation of predicted errors can be reduced by applying a weigh calculation algorithm. In addition, it is practical due to its fast convergence and small relative error.

VI. Conclusion

This paper discusses the neighbor selection policies for Phoenix NC system. The three policies including Random, Hybrid and Closest have been evaluated by various experiments. Based on the results of these experiments we propose a modified Phoenix NC system with Hybrid policy. Our work demonstrates Hybrid neighbor selection policy is useful in Phoenix Network Coordinate system. The new version achieves lower relative prediction error and has a better performance in selecting nearest host. Moreover, the Hybrid based Phoenix NC system has a better performance in the application of Overlay Multicast. It significantly reduces the tree cost of the ESM and MST multicast tree construction.

Acknowledgment

This work is supported by the *National Basic Research Program of China* (No.2007CB310806) and the *National Science Foundation of China* (No.60473087, No.60703052, No.60850003).

References

[1] Lee, D.D., Seung, H.S., *Learning the Parts of Objects by Non-negative Matrix Factorization*. *Nature* 401 (6755): 788-791 (1999).

[2] Y. Chen, X. Wang, X.X. Song, E.K. Lua, C. Shi, X.H. Zhao, B.X. Deng, X. Li, *Phoenix: Towards an Accurate, Practical and Decentralized Network Coordinate System*. In *Proc. of 8th International IFIP-TC6 Networking Conference (Networking'09)*, Aachen, Germany, May.2009.

[3] M. Costa, M. Castro, A. Rowstron, and P. Key., *PIC: Practical Internet Coordinates for Distance Estimation*. In *Proceedings of IEEE ICDCS*, March 2004.

[4] Mao, Y., Saul, L., Smith, J.M., *IDES: An Internet Distance Estimation Service for Large Network*. *IEEE Journal on Selected Areas in Communications (JSAC)* (2006).

[5] Dabek, F., Cox, R., Kaashoek, F., Morris, R., *Vivaldi: A Decentralized Network Coordinate System*. In *Proc. of ACM SIGCOMM* (2004).

[6] 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.

[7] Pietzuch P, Ledlie J, Mitzenmacher M, et al. *Network-Aware Overlays with Network Coordinates*. In *Proceedings of the 26th IEEE International Conference Workshops on Distributed Computing Systems, Washington, DC, USA: IEEE Computer Society*, 2006.12.

[8] Zhang, R.M., Tang, C.Q., Hu, Y.C., et al. *Impact of the Inaccuracy of Distance Prediction Algorithms on Internet Applications: an Analytical and Comparative Study*. In *Proc. of IEEE INFOCOM* (2006).

[9] Zheng, H., Lua, E.K., Pias, M., et al. *Internet Routing Policies and Round-Trip-Times*. In *Proc. of the Passive Active Measurement Workshop* (2005).

[10] Lua, E.K., Griffin, T., *Embeddable Overlay Networks*. *In Proc. of IEEE ISCC* (2007).

[11] Wang, G., Zhang, B., Ng, T.S.E., *Towards Network Triangle Inequality Violation Aware Distributed Systems*. In *Proc. of ACM IMC* (2007).

[12] A. Rowstron and P. Druschel. Pastry, *Scalable, distributed object location and routing for large-scale peer-to-peer systems*. In *Proc. IFIP/ACM Middleware 2001, Heidelberg, Germany*, Nov. 2001.

[13] D. R. Karger and M. Ruhl, *Finding nearest neighbors in growth-restricted metrics*. In *ACM Symposium on Theory of Computing (STOC '02)*, May 2002.

[14] M. Castro, P. Druschel, Y. C. Hu, and A. Rowstron, *Exploiting network proximity in peer-to-peer overlay networks*. *Technical Report MSR-TR-2002-82, Microsoft Research*, May 2002.

[15] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, *Scalable Application Layer Multicast*. In *Proceedings of ACM SIGCOMM*, August 2002.

[16] K. Chen and K. Nahrstedt, *Effective Location-Guided Tree Construction Algorithms for Small Group Multicast in MANET*. In *Proceedings of IEEE INFOCOM*, June 2002.

[17] PhoenixSimNs. http://www.net-glyph.org/~wanggang/Phoenix_NCS_sim.zip.