CANICULA: AN IMPROVED HYBRID OVERLAY NETWORKS

Yang Chen, Bei-xing Deng, Xing Li
Department of Electronic Engineering, Tsinghua University, Beijing, China
chenyang04@mails.tsinghua.edu.cn

Abstract - Hybrid Overlay Network (HONet) is a locality-aware hybrid overlay architecture, which combines the scalability and simplicity of a structured overlay with the connection flexibility of an unstructured overlay. In this paper, we propose Canicula, an improved HONet. First, using a simple and accurate network coordinate system, Canicula achieves more reliable node clustering. Secondly, Canicula uses an enhanced overlay construction method to reduce the impact of guarded hosts. The experimental results on over 450 PlanetLab nodes show that the ARDP in Canicula is only 45% of that in flat bidirectional Chord.

1. INTRODUCTION

Overlay networks are popular solutions for large-scale distributed network services and applications, such as application layer multicast, peer-to-peer file sharing and wide-area storage. An overlay network consists of a number of plain end hosts (peers) connected via either TCP or UDP. To construct a connected graph with connections (virtual links) among peers is a great challenge for most of the current protocols.

Structured overlays, such as Chord [1], Pastry [2], Tapestry [3] and De Bruijn networks [4], are more scalable comparing to unstructured overlays, but their homogeneous design will result in inefficient group communications when being applied on heterogeneous networks, either overloading servers or wasting resources. On the other hand, unstructured overlays, which is more flexible, often require flooding or gossip to route multicast messages [5] [6], hurting their scalability.

To take advantage of both structures, Tian *et al.* propose Hybrid Overlay Networks (HONet) [7]. HONet integrates the regularity of structured overlays with the flexibility of unstructured overlays by a hierarchical architecture.

Guarded hosts are hosts which can not accept incoming connections. The existence of *guard hosts* challenges overlay construction because not all hosts are capable of receiving and forwarding requests. The design of HONet does not consider the impact of *guarded hosts*. Moreover, the node distance prediction system in HONet is not accurate. In this paper we propose Canicula, an improved Hybrid Overlay Networks. First, we use triangulated heuristic [8] to predict the network distance. Then nodes self-organize into structured clusters based on our locality-aware clustering algorithm. Secondly, we propose an enhanced overlay construction method to reduce the impact of guarded hosts. We use bidirectional Chord [9] as the basic structured overlay in Canicula. We evaluate our design and compare it with flat bidirectional Chord overlay both by simulation and by experiments on over 450 PlanetLab [10] nodes.

The rest of this paper is organized as follows. First we review the related work and the HONet architecture in Section 2. Then we present our design of Canicula in Section 3 and evaluate its performance in Section 4. We conclude the whole paper with Section 5.

2. RELATED WORK

2.1 HONet

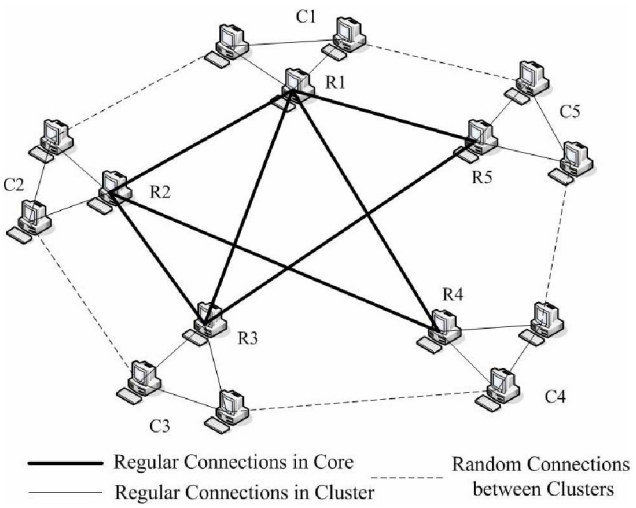

Fig. 1. HONet Architecture

For network locality, nodes form clusters and each cluster provides a root node. All root nodes form a backbone network for inter-cluster traffic. In addition, random connections between members across clusters serve as shortcuts to reduce network delay and bandwidth consumption. As shown in Fig.1, HONet is organized in a two-layer hierarchy. The lower layer consists of many clusters with a root node for each cluster. The cluster root nodes form the upper layer, the backbone network. The backbone network and each cluster are constructed as structured overlays with independent naming spaces.

In addition to inheriting the scalable routing of structured overlays and the flexibility of unstructured overlays, the clustered structure of HONet provides fault-isolation: faults in

0-7803-9746-0/06/$20.00©2006 IEEE

a cluster will mostly affect its local cluster nodes, with limited impact on nodes in other clusters.

2.2 Chord

Chord is a popular topology for routing in peer-to-peer networks. It is an undirected graph having 2^b nodes arranged in a circle, with edges connecting pairs of nodes which are 2^k positions away for any $k \geq 0$. The standard Chord routing algorithm uses unidirectional edges. And the average path length of unidirectional Chord is $b/2$ [1].

TCP connections are widely used as edges between nodes in Chord. Since TCP connections are bidirectional, Chord is actually undirected. In [9], Ganesan and Manku propose bidirectional Chord and exploit the bidirectionality of edges to minimize the path length. In bidirectional Chord, the average all-pairs shortest-path length is only $b/3 + O(1)$ [9].

3. CANICULA ARCHITECTURE

3.1 Network Distance Prediction

Just like in HONet, an important step in Canicula is node clustering, which requires the knowledge of nodes location. We employ a simple coordinate system to identify a node's network location, which uses a set of round-trip time from each node to a group of well-known landmark nodes. Any stable nodes which are able to response ICMP ping message can be chosen as landmark nodes.

After getting the network coordinate, we can predict the distance between two nodes. There are many distance predicting methods based on network coordinates, such as IDMaps [11], triangulated heuristic [8], GNP [8] and Vivaldi [12]. In HONet, Hilbert curves [13] are used to map the coordinates into numbers, called locality number (L number), and then the distance between two nodes is defined as the difference between two locality numbers. Because there is loss of information in mapping from n-dimensional space to one-dimensional space, two nodes which are close to each other in n-dimensional space may be far from each other after being mapped to one-dimensional space.

Canicula uses triangulated heuristic to predict the network distance for both simplicity and accuracy. The triangulated heuristic predicts network distance by assuming shortest path routing. First N nodes in a network are selected to be landmark nodes Bi, i=0, 1, …, N. Then, node H is assigned coordinates which are simply the N-tuple of distances between H and the N base nodes, i.e. $(d_{HB_1}, d_{HB_2}, \ldots, d_{HB_N})$. Given two nodes H1 and H2, assuming the triangular inequality holds, the lower bound of the distance between H1 and H2 is

$$L = max_{i \in \{1,2,\ldots,N\}} \left(\left| d_{H_1 B_i} - d_{H_2 B_i} \right| \right) \qquad (1)$$

And the upper bound is

$$U = min_{i \in \{1,2,\ldots,N\}} \left(\left| d_{H_1 B_i} + d_{H_2 B_i} \right| \right) \qquad (2)$$

Various weighted averages can then be used as distance functions to estimate the distance between H1 and H2.

To measure how well a predicted distance matches the actual measured distance, we use *relative error*, which is defined as follows.

$$relative \quad error = \frac{|predicted - measured|}{min(predicted, measured)} \qquad (3)$$

Thus, a zero relative error implies a prefect prediction. The smaller the relative error is, the better the triangulated heuristic predicts.

In [8], T. S. Eugene Ng and Hui Zhang show by large-scale network measurements that the upper bound U heuristic actually achieves good accuracy and performs far better than the lower bound heuristic L or the $(L+U)/2$ metric in the Internet. They found that with 15 globally distributed landmark nodes, triangulated heuristic can predict 90% of all paths with relative error below 0.59. Therefore we use U to predict the network distance in Canicula.

3.2 Node Clustering

After getting the network distance between nodes, we cluster the nodes. In real network applications, nodes can join or leave the network at any time, which will cause a heavy burden on centralized system. Therefore, we use a distributed clustering algorithm.

To join the overlay, a node first identifies its coordinates in the network. Then it uses triangulated heuristic to find the closest cluster root. In our design, if the new node cannot locate nearby cluster roots, or its distance to the nearest cluster root is larger than a threshold T, this node joins the backbone network as a new cluster root and announces its coordinates in the backbone network. Otherwise, the node joins the cluster led by its nearest cluster root. More details are discussed in Section 3.3.

There may be some outliers with every dimension of their coordinates bigger than T. We call these nodes "island". Instead of joining in the backbone, these islands join the cluster lead by its nearest cluster root.

Because we design our algorithm in a distributed way, no end-to-end measurement is performed between all pairs of nodes. This design reduces the overhead for large scale network and provides Canicula with high scalability.

In triangulated heuristic, shortest path routing is assumed. But due to the inefficient routing behavior in the Internet, this assumption is not always true. For example, in [14], for all the triangular closed loop paths (a,b), (b,c) and (a,c) that they measured, 7% of the $(a,c)/((a,b)+(b,c))$ ratios

are greater than 1. This error hurts the accuracy of the network distance prediction.

Base on our experiment on more than 150 PlanetLab nodes [15], the average distance between each pair of nodes in the same cluster is 23ms and the average distance between each pair of nodes in different clusters is 193ms. So the result of our node clustering is reliable, keeping the intra-cluster distance small and inter-cluster distance large.

3.3 Overlay Construction under Limited End-to-End Reachability

3.3.1. Existence of Guarded Hosts in Internet

Most of current network-overlay construction assumes two-way communication capability: each host can initiate outgoing connections as well as accepting incoming connections. But this assumption is not always true especially due to the use of Network Address Translation (NAT) and firewalls. Here we define guarded host as hosts which can not accept incoming connections. In [16], experiments on eDonkey and Gnutella file-sharing systems reveal that as many as 36% of the hosts may be guarded, i.e. not accepting incoming connections. Chu et al. reported even higher percentages in their Internet overlay multicast experiments [17]. With the presence of guarded hosts, network reachability is no longer symmetric for every pair of overlay hosts, which challenges the overlay construction.

3.3.2. Overlay Construction in Canicula

Wang *et al.* proposes an overlay optimization called e* to help existing overlay protocols overcome the reachability problem [18]. In our design, we use similar idea as e* to construct overlay under limited end-to-end reachability.

A natural approach to integrate guarded hosts into an overlay is to group them into clusters by locality, and then assign an open host as the root of each cluster. To achieve reasonable performance, cluster roots must be carefully selected to meet several criteria, such as the speed of network connection or unicast latencies to other nodes in the cluster.

First of all, we must check whether a host is guarded or not. A host sends query messages to selected overlay nodes with a callback bit set in each message. Upon receiving such messages, an overlay node attempts to connect to the caller. If any of these callbacks succeeds, the callback bit will be cleared in the following queries, and this host considers itself as open. If all of the requested callbacks fail to return, the host recognizes itself as guarded. Also, whenever one host is connected by another host, it recognizes itself as open.

Then, we employ a Root Election Protocol. Root election in a cluster is based on a Root Rank Vector (RRV). Each overlay node has its own RRV and each element in the RRV is a test condition for root election. In Canicula, a typical RRV is as follows.

$$RRV =< open, lifetime, cluster\ \ dist > \qquad (4)$$

In RRV, *open* represents whether one host is guarded; *lifetime* represents how long the host has stayed in the overlay; *cluster dist* represents the summation of latencies to all the other members in the cluster.

Each overlay node is responsible for keeping its RRV up-to-date. A node periodically updates its RRV by active and passive probing. The computation of cluster dist requires latencies to other cluster members, which can be measured by the periodically exchanged "heartbeat" messages without extra measurement overhead. Each cluster member includes RRVs as part of its "heartbeat" messages to others within the same cluster. The received RRVs are then sorted in the order of the three elements with *open* having the highest priority and *cluster dist* the lowest. The node with the top ranked RRV is elected as the root.

3.4 Basic Structured Overlay and Message Routing

In Canicula, we use bidirectional Chord as our basic structured overlay. An important reason for choosing bidirectional overlay is the ubiquity of the guarded hosts. Without using bidirectional overlay, the guarded hosts will be unreachable. The overlay routing of bidirectional Chord is presented in [8].

If a message is destined for a local cluster node, normal structured overlay routing presented in [8] is used. Otherwise, we will use two approaches, hierarchical routing and fast routing, which are proposed in HONet [7]. In either case, DHT routing is used in the backbone network and inside local clusters.

In hierarchical routing, messages are delivered from one cluster to another through the backbone network. Fast routing utilizes the random connections between clusters as inter-cluster routing shortcuts. To implement fast routing, each cluster nodes announces information about its inter-cluster links in the local cluster DHT.

3.5 Summary of Canicula

Fig.2 shows the procedure that a new member A joins a Canicula overlay. Host A first contacts the rendezvous service provided by a Login Server. After obtaining a list of cluster roots from the Login Server, Host A joins a nearest cluster. After joining the overlay, Host A participates in the root election in its cluster. Though nodes can join and leave the overlay at any time, cluster will make corresponding adjustment as soon as possible, not only when the root node departs unexpectedly but also when there is a more appropriate node to serve as root node. The root election, random connections creation and cluster improvement tasks may be done in parallel. To detect node failure, root nodes exchange "heartbeat" messages, so do members in each cluster.

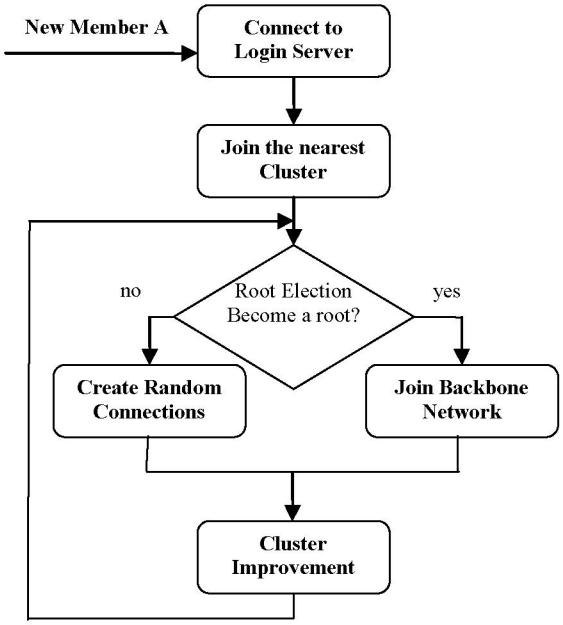

Fig. 2. Workflow of Canicula

4. SIMULATION AND EXPERIMENT

The main performance metric we adopt is the Average Relative Delay Penalty (ARDP). Here we distinguish the latencies in an overlay, which we call overlay latencies, from the unicast latencies, which are the latencies in the underlying physical network. Relative Delay Penalty (RDP) is the ratio of the overlay latency between nodes i and j to their unicast latency. ARDP is the average RDP among all node pairs:

$$ARDP = \frac{1}{N}\sum_{i,j(i\neq j)}^{N}\frac{D'_{i,j}}{D_{i,j}} \qquad (5)$$

where N is the number of node pairs in the overlay. Smaller ARDP indicates that most overlay latencies are close to the respective unicast latencies.

4.1 Simulation Setup

A transit-stub network topology with 287 transition nodes and 3,000 stub nodes is generated for our simulation. 300 PlanetLab nodes are used to help topology construction. We first collect the 300×300 network distances using ping (100 RTT samples per path), then apply clustering to select 13 well distributed nodes as landmarks. The remaining 287 nodes serve as transition nodes. We assign different distances to the edges in the topologies: the distance of intra-stub edges is 1; the distance of the edges between transition node and stub node is a random integer within [2; 15]; and the distance between transition nodes is measured from the distance matrix. T is set as 40 ms in our simulation. To evaluate the impact of guarded hosts, we set some hosts as guarded and increase the percentage of guarded hosts from 0% to 50%. We compare the performance of Canicula with a flat bidirectional Chord overlay using this topology. Ten runs are performed on the network topology and the average value is reported.

4.2 Performance Evaluation on the Generated Topology

Since the random connections are an import factor affecting the performance of Canicula, we compare the performance of Canicula by varying the number of random connections on each node (RC = 1 or 2).

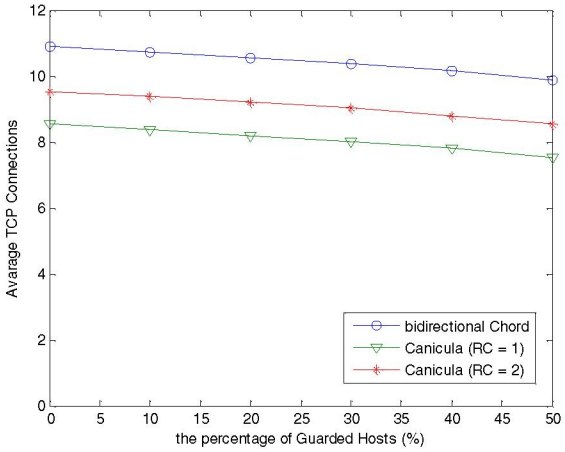

Fig. 3. Average TCP Connections

Fig.3 shows the average number of the TCP connections in each node with the increasing guarded hosts. Generally, an overlay with a larger number of links produces lower ARDP but consumes more network resources. On the other hand, as shown later, with the same number of links, the ARDP of an overlay depends on its path optimization algorithm. From Fig.3 we find that in our simulation, when the number of random connections is 1 or 2, the average number of the TCP connections of each node in the Canicula overlay is less than that in the flat bidirectional Chord.

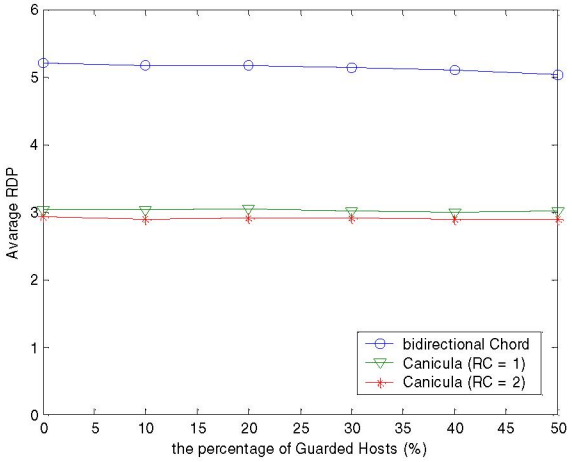

Fig. 4. Average RDP

Fig.4 shows the comparison of ARDP between Canicula and flat bidirectional Chord. The ARDP in Canicula is much smaller than that in flat bidirectional Chord because Chord does not consider the network locality, though Canicula uses less TCP connections as shown in Fig.3. When we set RC to 2 in Canicula, the ARDP is only 55% of that in the flat bidirectional Chord. Our simulation results show that hierarchical structured overlays perform better than flat structures.

4.3 Experiment Setup

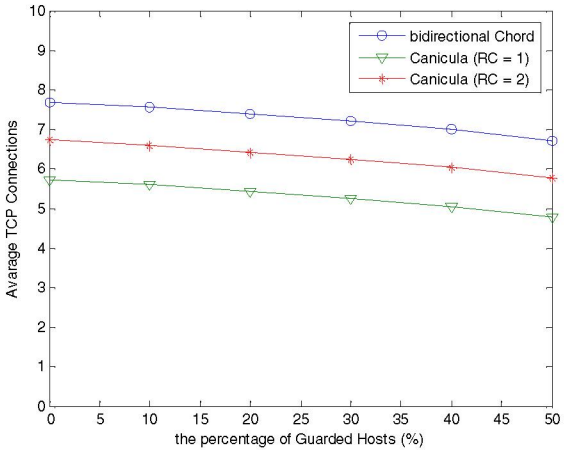

Fig. 5. Average TCP Connections

Fig.5 shows the average number of TCP connections in each node with the increasing guarded hosts. When the number of random connections is 1 or 2, the average number of TCP connections of each node in the Canicula overlay is less than that in the flat bidirectional Chord.

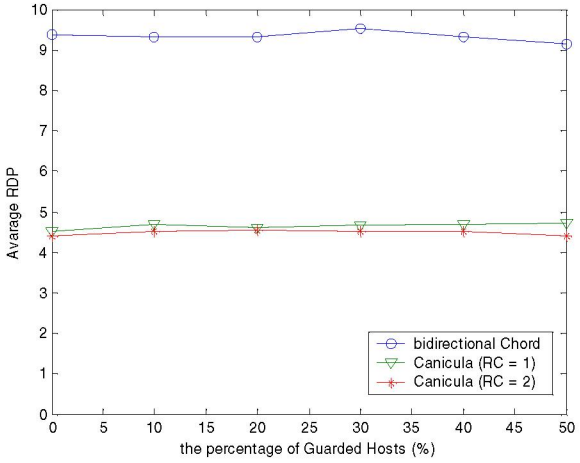

Fig. 6. Average RDP

Fig.6 shows the comparison of ARDP between Canicula and flat bidirectional Chord. When RC is set to 2 in Canicula, the ARDP is only 45% of that in the flat bidirectional Chord. Our experiment results show that hierarchical structured overlays perform better than flat structures.

Another important factor to evaluate our overlay design is the end-to-end reachability through the overlay routing. Due to the Internet heterogeneity and the ubiquity of the guarded hosts, some of the node pairs in the overlay can not connect with each other directly. So there may be some node pairs in our overlay can not reach each other through the overlay routing.

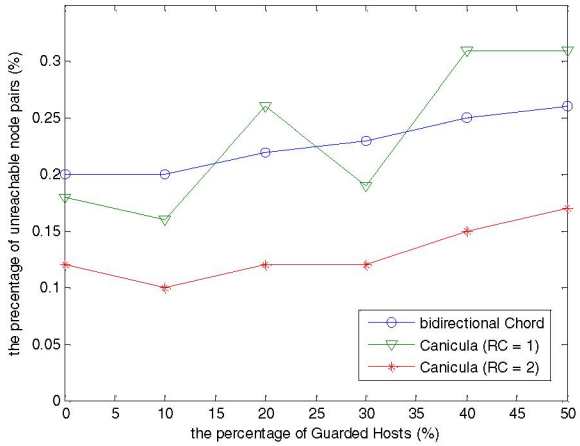

Fig. 7. Unreachable node pairs

Fig.7 shows that given the percentage of guarded hosts below 50%, the percentage of the unreachable node pairs is less than 0.32%. It means that most node pairs are reachable between each other through the overlay. And Canicula (RC = 2) achieves better reachability than flat bidirectional Chord.

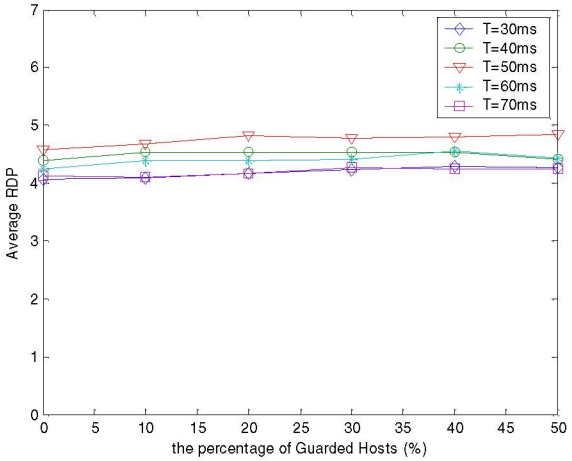

Fig. 8. ARDP of different T

Finally we analysis the impact of T. We compare the ARDP of Canicula (RC = 2) by varying the T from 30ms to 70ms. Fig.8 shows that different T in this range does not hurt ARDP much. So the overlay construction in Canicula does not require much effort to tune T for clustering performance.

5. CONCLUSION AND FUTURE WORK

In this paper, we propose Canicula, an improved Hybrid Overlay Network. According to our simulation and our experiment on PlanetLab, Canicula performs well in Internet, even when up to 50% of the hosts are guarded. It achieves much lower ARDP than flat bidirectional Chord, without consuming more network resources. Caniclua also has better overall reachability with less than 0.32% node pairs unreachable between each other through the overlay.

We currently focus on the node clustering effect and the end-to-end reachability of Canicula, and are preparing to deploy Canicula in a wide range area. We are also developing some exciting applications on this hybrid overlay, such as peer-to-peer streaming, application level multicasting.

6. ACKNOWLEDGMENT

This work is supported by the National Science Foundation of China (No.60473087).

7. REFERENCES

[1] Ion Stoica, Robert Morris, David Karger and M. Frans Kaashoek, Hari Balakrishnan. "Chord: A scalable peer-to-peer lookup service for internet applications". In *Proc of ACM SIGCOMM'01*, 2001

[2] Antony Rowstron and Peter Druschel. "Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems". In *Proc of ACM Middleware'01*, 2001.

[3] Ben Y. Zhao, Ling Huang, Jeremy Stribling, *et al.* "Tapestry: A resilient global-scale overlay for service deployment". *IEEE Journal on Selected Areas in Communications*, v22n1, Pages 41-53, 2004.

[4] Dmitri Loguinov, Anuj Kumar, Vivek Rai, and Sai Ganesh. "Graph-theoretic analysis of structured peer-to-peer systems: Routing distances and fault resilience". In *Proc of ACM SIGCOMM '03*, 2003.

[5] Xin Yan Zhang, Qian Zhang, Zhensheng Zhang, *et al.* "A Construction of Locality-Aware Overlay Network: mOverlay and Its Performance". *IEEE Journal on Selected Areas in Communications*, Vol.22, No.1, Pages 18-28, 2004.

[6] Anne-Marie Kermarrec, Laurent Massoulie and Ayalvadi J. Ganesh. "Probabilistic Reliable Dissemination in Large-Scale Systems". *IEEE Transactions on Parallel and Distributed systems*, Vol.14, No.3, Pages 248-258, 2003.

[7] Tian Ruixiong, Xiong Yongqiang, Zhang Qian, Li Bo, Zhao Ben Y., Li Xing, "Hybrid Overlay Structure Based on Random Walk". In *Proc of the 4th International Workshop on Peer-To-Peer Systems (IPTPS'05)*, 2005.

[8] T. S. Eugene Ng and Hui Zhang. "Predicting Internet network distance with coordinates-based approaches". In *Proc of IEEE INFOCOMM'02*, 2002.

[9] Prasanna Ganesan, Gurmeet Singh Manku. "Optimal Routing in Chord". In *Proc of 15th Annual ACM-SIAM Symposium on Discrete Algorithms(SODA'04)*, 2004.

[10] PlanetLab. http://www.planetlab.org/.

[11] Paul Francis, Sugih Jamin, Cheng Jin, *et al.* "IDMaps: A Global Internet Host Distance Estimation Service". *IEEE/ACM Transactions on Networking,* v9n5, Pages 525-540, 2001.

[12] Frank Dabek, Russ Cox, Frans Kaashoek, *et al.* "Vivaldi: A Decentralized Network Coordinate System". In *Proc of ACM SIGCOMM'04,* 2004.

[13] Tetsuo Asano, Desh Ranjan, Thomas Roos, *et al.* "Space-filling curves and their use in the design of geometric data structures". *Theoretical Computer Science*, v181n1, Pages 3-15, 1997.

[14] Paul Francis, Sugih Jamin, Vern Paxson, *et al.* "An architecture for a global Internet host distance estimation service". In *Proc of IEEE INFOCOMM'99*, 1999.

[15] Yang Chen, Bei-xing Deng and Xing Li, "Experimental study on network coordinate based node clustering". *Journal of Dalian University of Technology*, Vol.45, Supp.1, Pages 41-43, 2005.

[16] Wenjie Wang, Hyunseok Chang, Amgad Zeitoun and Sugih Jamin. "Characterizing Guarded Hosts in Peer-to-Peer File Sharing Systems". In *Proc of IEEE GLOBECOM'04*, 2004.

[17] Yang-hua Chu, Aditya Ganjam, T.S. Eugene Ng, *et al.* "Early Experience with an Internet Broadcast System Based on Overlay Multicast". In *Proc of USENIX'04*, 2004.

[18] Wenjie Wang, Cheng Jin and Sugih Jamin. "Network Overlay Construction under Limited End-to-End Reachability". In *Proc of IEEE INFOCOMM'05*, 2005.