A Distributed Topology-aware Overlays Construction Algorithm

Xiaoming Zhang
School of Computer Science and Engineering, Beihang University, Beijing, China,100083 Phn: +86-01082338247
yolixs@163.com

Zhoujun Li
School of Computer Science and Engineering, Beihang University,Beijing, China 100083
Zhoujun.li@263.net

Yijie Wang
National Laboratory for Parallel and Distributed Processing, NUDT, Changsha, China,410073
wwyyjj1971@vip.sina.com

ABSTRACT
Peer-to-Peer (P2P) computing systems are rapidly growing in importance as the medium of choice for the mass storage. Peers in the most P2P systems randomly choose logical neighbors without any knowledge about underlying physical topology. This mechanism can cause a serious topology mismatch between the P2P overlay network and the underlying network. It greatly limits the performance gain from various search or routing techniques. In this paper, a distributed topology-aware overlays construction algorithm Taonet is proposed, in which the overlay network construction is based on the location information of underlying topology. The nodes that are close in the logical network are also close in terms of network latency. This leads to low latency stretch of the overlay network. This algorithm is scalable and distributed. Simulation results indicate that the latency of data location in these topology-aware P2P systems that are constructed with Taonet can be significantly decreased.

1. INTRODUCTION
In the past few years peer-to-peer (P2P) systems [1,2,3,4,5,6,7,8] have increased in importance as the medium for mass storage. Unlike traditional client-server systems that essentially rely on a few servers with high reliability and availability, P2P systems decentralize the management of data storage and harness unused resources on computers distributed across the world. Within a P2P system, failure of a few nodes may not result in a sharp decrease in data availability because all nodes are peers and the remaining peers may provide or reproduce the required data. The potential of storing Petabytes of data accessible whenever and wherever required has spurred great enthusiasm. This has lead to a renewed interest in techniques for routing queries to data using names efficiently.

In some P2P systems [4,5,6,7] that can serve as medium for mass storage a data can be found with polylogarithmic number of hops. These systems make use of application-level or overlay networks, and a routing path on the overlay network then consists of a series of application-level, not IP-level, hops between the source and destination nodes. However, in these systems, little effort is made to ensure that this application-level connectivity is congruent with the underlying IP-level network topology. This in turn can lead to inefficient routing where, for example, a node in Changsha has its neighbor nodes in Beijing and hence its path to a node in Wuhan may traverse distant nodes in Beijing. Researches [9,10] show that P2P traffic contributes the largest portion of the Internet traffic on some popular P2P systems. Thus a fundamental challenge in using large-scale overlay networks is to incorporate IP-level topologies information in routing process.

The rest of this paper is organized as follows. In section 2 we introduct the related works. Section 3 contains the algorithm of predicting network distance, and section 4 introducts the construction of topologt-aware overlay network. The constraction of decentralized P2P overlay is described in section 5. Section 6 gives the performance evaluation. We conclude in section 7 with a brief summary

2. RELATED WORKS
Liu,Y et al aim at alleviating the mismatching problem of unstructured P2P overlay and reducing the unnecessary traffic [11]. It builds an efficient overlay by disconnecting slow connections and choosing physically closer nodes as logical neighbors while still retaining the search scope and reducing response time for queries. In structured overlay there are mainly three approaches to topology-aware routing [12], namely proximity routing, proximity neighbor selection and topologybased nodeId assignment.

Proximity routing**:** When a message is routed, the idea is to select, among the possible next hops, the one that is closest in the underlying network or one that represents a good compromise between progress in the id space and proximity. With k alternative hops in each step, the approach can reduce the expected delay in each hop from the average delay between two nodes to the expected delay of the nearest among k nodes with random locations in the network. The main limitation is that the benefits depend on the magnitude of k; with practical protocols, k is small. Moreover, depending on the overlay protocol, greedily choosing the closest hop may lead to an increase in the total number of hops taken.

Proximity neighbor selection**:** The idea is to choose routing table entries to refer to the topologically nearest among all nodes with nodeId in the desired portion of the id space. The success of this technique depends on the degree of freedom an overlay protocol has in choosing routing table entries without affecting the expected number of routing hops. A limitation of this technique is that it does not work for overlay protocols like CAN and Chord, which require that routing table entries refer to specific points in the id space. It has been successfully used in Tapestry [13, 14]. They choose routing table entries to referto the topologically nearest among all nodes with nodeId in the desired portion of the id space when construct Tapestry routing table.

Topology-based nodeId assignment**:** It attempts to map the overlay's logical id space onto the phisical network such that neighboring nodes in the id space are close in the physical network. It has been successfully used [15,16].

MG '08, January 29-February 3 2008, Baton Rouge, LA, USA

Topologically-Aware CAN [15] uses landmark ordering to cluster nodes that are physically close into logical vicinity during overlay construction. However, a fixed set of landmarks renders this approach unsuitable for dynamic networks, such as ad-hoc networks and the resulting stretch remains high in their simulation results.

Z. Xu et al introduce maps containing information on close-by nodes in a specific region to allow nodes to join the overlay network with a more accurate reflection of its own position in the physical network [16]. A map is stored as global soft state among the nodes of a region. This approach, however, comes with a significant overhead. Potentially, there could be a very large number of regions, all of which have to maintain their map. Moreover, to achieve a finer granularity, additional inner-bin measurements are required.

In this paper, we chose network latency as metric for measuring distance in underlying network, and an algorithm named Taonet whereby the overlay network construction is based on the location information of underlying network is presented. Each node is connected to a small subset of the other nodes that are close in underlying network. The nodes that are close in the logical network are also close in underlying network. Comparing with above algorithms, our approach doesn't rely on any centralized server or global knowledge, and don't increase the overhead of underlying network greatly. We apply this algorithm to the construction of topology-aware P2P systems. Simulation results indicate that the latency of data location in these topology-aware P2P systems can be significantly reduced.

3. PREDICTING NETWORK DISTANCE

Ng et al. proposed a Cartesian coordinate-based approach, called Global Networking Positioning (GNP) [17], to predict Internet network distance. GNP represents the location of each host in a N-dimensional Cartesian coordinate system. The coordinate of a host is the distances from itself to the landmark nodes, and the distance between two hosts is calculated as the Euclidean distance in the Cartesian coordinate.

However, GNP has its own drawbacks. First, the landmarks must be globally distributed to the entire Internet to accurately measure the distances between two arbitrary distant hosts. But the distance estimation between two local hosts is biased with the coordinates of the widespread landmarks, which are not local to the given two hosts. Second, these landmarks are central points of failure and may become target of attacks.

To address these shortcomings presented above, a coordinate-based Pure Peer-to-Peer Network Positioning (Triple-P NP or TPNP) architecture is proposed in [18]. The idea of TPNP is similar to the basic principle of GNP. A host that is interested in participating the TPNP system first discovers the nearby hosts that are already in the system to use them as reference hosts, which replace the landmarks for the host, in question. Then, the host contacts these references to compute own coordinates. A selection heuristics is used to select reference nodes from the candidates discovered by host that is interest to participate TPNP.

a) Pick the hosts, which reside in outside domains or use greatest number of hosts in different outside domains as their own reference hosts.

b) Pick the hosts, which are closest to origin in Euclidean space (i.e. the ones with absolute smaller coordinate values).

c) Pick the closest hosts (i.e. quick responding hosts).

(a) tries to ensure that the coordinates converge not only locally but also globally. (b) tries to ensure that the system/coordinates converges towards a fixed point to prevent chaos. Coordinates may go out of control over time due to absence of fixed hosts (e.g., landmarks), accumulated error terms and periodic recalculations. In TPNP a single point that never changes its coordinates is used to solve the similar problem. For example, this point can be set to *O* (0, 0) as the origin, assuming a two-dimensional space. (c) tries to minimize overhead into the network due to distance measurements. It is applied when the first two cannot make a decision. After the participating host measure the distances to reference hosts TPNP follows the coordinate computation process of GNP.

4. CONSTRUCTION OF TOPOLOGY-AWARE OVERLAY NETWORK

4.1 Overlay design

The intention of constructing topology-aware P2P overlay network is to ensure that the overlay topology congruent with the underlying IP-level network topology. The construction algorithm is scalable and distributed for that P2P is a fully distributed and cooperative overlay network. The main idea of Taonet is:

1. Underlying network is modeled as an n-dimension coordinate space. The entire coordinate space is dynamically partitioned among all the nodes in the system such that every node "owns" its individual, distinct zone within the overall space. Distinct zones don't intersect with each other.
2. The coordinate space is used to store node's location in underlying network. Each node's coordinate reflects its location in the network and lies within its zone.
3. Each node has a neighbor table that is used to store informations of neighbor nodes. The neighbor table is used as a routing table when a node joins. Nodes in this system self-organize into an overlay network that represents this coordinate space.

The definition of neighbor is: in a d-dimensional coordinate space, two nodes are neighbors if their zone spans overlap along d-1 dimensions and abut along one dimension. For example, Fig.1 (a) shows a 2-dimensional [0,1]×[0,1] coordinate space partitioned between 5 nodes, and the neighbors of node C are node D and E.

4.2 Construction algorithm Taonet

As described above, the entire coordinate space is divided amongst the nodes currently in the system. To allow the system to grow incrementally, a new node that want to join the system must get its own coordinate and be allocated its own portion of the coordinate space or zone. The coordinate of is calculated through the algorithm TPNP. The zone of a new node is received from an existing node splitting its allocated zone into two portions, retaining one portion and handing the other portion to the new node. Each node has a cache that cache IP address and coordinate of a set of fix nodes or nodes that are close to origin in Euclidean space. The node-joining procedure is shown in pseudo code form in Fig.2. A, B are nodes and A is the node want to join. Set is a set of reference nodes.

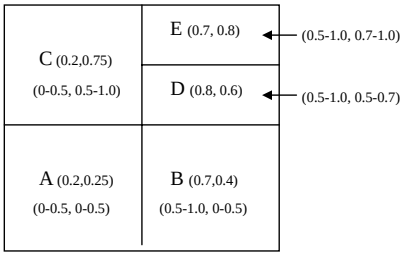

Figure.1. (a) Example 2-d space overlay with 5 nodes.

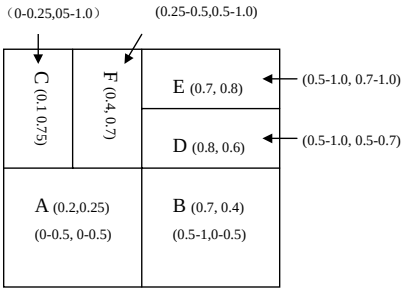

Figure.1. (b) Example 2-d space after node F joining

1.	A discovers a existing node B use IP-multicast;
2.	Set = Select k reference nodes form B;
3.	A. Coordinate = Calculated by TPNP with Set;
4.	while (! B.zone contains A.Coordiante)
5.	route to B∈ B.neighborList that is closet to A;
6.	A.zone_c = portion of B.zone_c;
7.	A.neighborList_c = B.neighborList_c ∪ {B};
8.	A and neighbor nodes adjust neighborList_c;

Figure. 2. procedure of node joining

The new node A first discovers the IP address of a close node B currently in the system use IP-multicast (line 1). It get k reference nodes in which k/2 nodes are the closest ones selected from B's neighbor table and k/2 nodes are the ones that are closest to origin in Euclidean space selected from B's cache (line 2). New node's coordinate is calculated through TPNP[18] (line 3). Then node A sends a join request destined for node B whose zone contain coordinate of node A (line 4-5). This message is sent into the system via an existing node. Each existing node then uses the routing mechanism to forward the message, until it reaches the destination node. The routing mechanism is: Using its neighbor table, a node routes a message towards its destination by simple greedy forwarding it to the neighbor with coordinate closest to the destination coordinate. The zone of destination node is splinted into two portions; Destination node retains one portion and the new node handles the other one (line 6). The splitting process is described as:

If the destination node owns a zone of $(a_1\text{-}b_1, a_2\text{-}b_2, \ldots, a_n\text{-}b_n)$, and the coordinate is $(x_1,x_2,\ldots,x_n)$, and coordinate of the new node is $(y_1,y_2,\ldots\ldots,y_n)$. Then the zone is splinted along the dimension by which the value $|x_i\text{-}y_i|(1<=i<=n)$ is the biggest. If $|x_h\text{-}y_h|$ is the biggest one, and the portion of zone handled to new node is $(c_1\text{-}d_1, c_2\text{-}d_2, \ldots.., c_n\text{-}d_n)$. Then:

If $y_h>x_h$ $c_h=(x_h+y_h)/2$, $d_h=b_h$, $b_h=c_h$;

Else $c_h=a_h$, $a_h=(x_h+y_h)/2$, $d_h=a_h$;

$c_k=a_k$, $d_k=b_k$, (0<k<=n && k!=h)。

After receive a portion of zone splinted from destination node, node A copies the neighbor table from destination node (line 7). Finally node A and nodes in neighbor table adjust their neighbor table, and nodes in the neighbor table also adjust their neighbor table (line 8).

For example if the coordinate space of a overlay network is a 2-dimension space as the space in Fig.1. (a). A node F whose coordinate is (0.4, 0.7) tries to join the system. It first finds the destination node C whose zone is then splinted into two portions. The new node handles the portion of (0.25-0.5,0.5-1.0) and receives the neighbor table (A, E, D) from node C, and then it connects to neighbor nodes and adjust its neighbor table to (C, E, D). The neighbor table of node C is adjusted to (A, F), and the neighbor tables of node E and D are also adjusted. The result is described if Fig.1.(b).

When a node leaves the system, if its zone can merge with some node's zone, then its zone merge with the closest node's zone. If not, then its zone is handled to the closest neighbor. Under normal condition a node sends period update message to each of its neighbors giving its zone coordinate, its own coordinate, a table of its neighbors and their zone coordinate. The prolonged absence of an update message from a neighbor signals its failure.

Once a node has decided that its neighbor has died it initiates the takeover mechanism and starts a takeover timer running. Each neighbor of the failed node will do this independently, with the timer initialized in proportion to the distance to the failed node. When the timer expires, a node sends a TAKEOVER message conveying its own coordinate and zone volume to all of the failed node's neighbors.On receipt of a TAKEOVER message, a node cancels its own timer if the zone in the message can and its own zone can't merge with the zone of the failed node, or the two zone all can merge with the zone of failed node but the zone in the message is smaller, or the two zone all can't merge with the zone of failed node but the distance in the message is shorter. Otherwise it replies with its own TAKEOVER message.

5. CONSTRUCTION OF DECENTRALIZED P2P OVERLAY

We will show how to construct topology-aware P2P system use algorithm Taonet. The construction of topology-aware decentralized structured system is illustrated by the example of construction of topology-aware CAN, and the construction of topology-aware decentralized unstructured system is illustrated by the example of construction of topology-aware Gnutella.

5.1 Topology-aware CAN

A Content-Addressable Network is an application level network whose constituent nodes can be thought of as forming a virtual d-dimensional Cartesian coordinate space [5]. This coordinate space is completely logical and bears no relation to any physical coordinate system. At any point in time, the entire coordinate space is dynamically partitioned among all the nodes in the system such that every node owns its individual, distinct zone within the overall space. This virtual coordinate space is used to store (key, value) pairs as follows: to store a pair (K, V), key K is deterministically mapped onto a point P in the coordinate space using a uniform hash function. The corresponding key-value pair is then stored at the node that owns the zone within which the point P lies. How-

ever, the CAN construction mechanisms allocate nodes to zones at random. Thus, a node's neighbors on the CAN need not be nearby on the underlying IP network. This can lead to inefficient routing because every application-level hop on the CAN could potentially be between two geographically distant nodes. Some works on CANs [5] explores a number of heuristics to lower the latency of CAN routing. These schemes however try to improve the selection of paths on an existing overlay. Our work in this paper tries to improve the structure of the overlay itself.

We construct a topology-aware CAN which we call TCAN using algorithm Taonet. Two coordinate spaces is needed. One space that called geo space is used to store the physical location information of nodes, and the other space that called id space is used to store (key, value) pairs. The two coordinate spaces is dynamically partitioned among all the nodes in the system such that every node owns its individual, distinct geo zone and id zone. Each node's coordinate lies within its geo zone and the hashing values of keys that the node own lie within its id zone. Each node owns two neighbor tables. One neighbor table which we call it geo table is used to store information of neighbor nodes in underlying network. The other neighbor table which we call it id table is used to store information of neighbor nodes in id space and it is the same as routing table in CAN. The construction algorithm of TCAN is shown in pseudo code form in Fig. 3.

```
1.   A discovers a existing node B use IP-multicast;
2.   Set = Select k reference nodes form B;
3.   A. Coordinate = Calculated by TPNP with Set;
4.   while (! B.zone contains A.Coordiante){
5.      route to B∈B.neighborList_c that is closest to A;
6.   A.zone_c = portion of B.zone_c;
7.   A.zone_v = half of B.zone_v;
8.   A.neighborList_c = B.neighborList_c ∪ {B};
9.   A.neighborList_v = B.neighborList_v ∪ {B}
10.  A and physical neighbors adjust neighborList_c;
11.  A and logical neighbors adjust neighborList_v;
```

Fig. Figure 3. Construction of TCAN

The actions taken in lines 1-6 in Fig. 3 are the same as in lines 1-6 in Fig. 2. The id zone of node B within whose geo zone the new node's coordinate lies is splinted into two half, and one half retained and the other half is handled by new node (line 7). Then new node A copies two neighbor tables from node B, and neighborList_c is the geo table and neighborList_v is the id table (lines 8-9). Finally node A, nodes in geo neighbor table and id neighbor table adjust their neighbor tables (lines 10-11). It can be concluded that nodes that are close in virtual coordinate space are also close in underlying network. When a message is routed, each next hop is the hop that is closest in underlying network.

For example: in a system the geo space is as figure 1 and the 2-dimension virtual space is (0-2.0, 0-2.0). Then the virtual space is splinted among 5 nodes as show in Fig.4 (a) After joining of node F, the virtual space is splinted as show in Fig.4 (b).

5.2 Topology-aware Gnutella

In decentralized unstructured system, each node randomly chooses logical neighbors without any knowledge about underlying topology. If each node has knowledge about physical location of other nodes, then it can chooses neighbors that are closest to itself in underlying network. As a result, the expected delay of each hop in the routing path can be low. This leads to low latency stretch of the overlay network. But even under the assumption of global knowledge of the IP-level latencies between every possible pair of nodes, the problem of constructing an optimal overlay is know to be NP-hard [19].

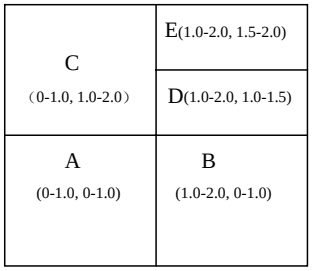

Figure 4. (a)Example of 2-d virtual space with 5 nodes

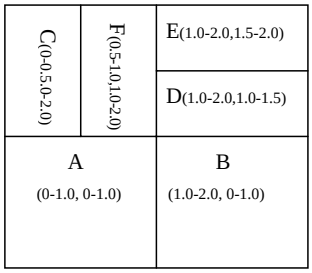

Figure 4. (b)Example of 2-d virtual space after node F joining

We will use a heuristic algorithm [15] to construct a topology-aware Gnutella which we call TGnutella: a node picks its k neighbors by picking the k/2 nodes in the system closest to itself and then picks another k/2 nodes at random. The intuition in devising the above algorithm was that the k/2 connections to closeby nodes will result in well-connected pockets of nearby nodes and reduce the average latency between two routing hops, while the random links serve to keep the graph connected and to interconnect these different pockets of nodes and improve the succeed rate of data location.

Each node in this system owns a table that maintained with LRU mechanism to store information of remote nodes. When nodes receive a message it check the table and replace the latest used recently node with the node in message. The k/2 random nodes are selected from the table. The k/2 closest nodes are select from neighbor tables constructed using algorithm Taonet. The construction of TGnutella is shown in pseudo code form in Fig.5.

New node A initializes the table that store information of remote nodes by copying it from a node B currently in the system (line 1). Then new node searches k/2 closest nodes from neighbor tables constructed with algorithm Taonet (lines 2-8), and the search is breadth first. Finally new node picks k/2 random nodes from the table that maintain a list of remote nodes (line 9).

```
1. A.remoteList = remoteList of a node currently in the system
2. insert A to Queue;
3. while (number of A.neighbor_close < k/2){
4.     num = k/2 – number of A.neighborList_v;
5.     C = select the closest node form Queue and delete it;
6.     set = select no more than num closest nodes form
       C.neighborList_c;
7.     A.neighbor_close = A.neighbor_close ∪ set;
8.     insert set to Queue;}
9. A.neighbor_random = select no more than k/2 nodes form
   remoteList;
```

Figure 5. Construction of Tgnutella

6. PERFORMANCE EVALUATION

We test the algorithm Taonet on both simulated topologies and Internet measurement data. The test topologies we use are as follows:

P2PSim: The P2Psim project [20] measured a distance matrix of RTTs among about 2000 Internet DNS servers based on the King method [21]. The DNS servers were obtained from an Internet-scale Gnutella network trace.

BRITE: BRITE [22] is a more realistic topology generator, which generates truly representative Internet topologies. Here we use a two-level hierarchical topology generated by BRITE. The hierarchical topology was generated by the bottom-up approach. The router-level topology used *Waxman* models, and router nodes were assigned to autonomous systems in the way of *heavy-tailed*. We constructed Chord networks of size 4096, and marked 100 as the size of AS.

We construct topology-aware P2P overlay based on both data measured in P2PSim and topology generated by BRITE, and use two metrics to evaluate algorithm Taonet. One metric is the average ratio of the latency of the shortest-path between two nodes on the overlay network to the latency of the shortest-path between two nodes on the underlying, and we call this metric overlay latency stretch. The other metric is the ratio of the average ratio of data location latency in P2P system constructed with algorithm Taonet to data location latency in original P2P system, and we call this metric data location stretch. The simulation in paper [17] shows that a 7-dimensional space has a good result, and when the dimension is increased the result is unchanged. So in our simulations the network is modeled to a 7-dimensional coordinate space.

1. Measuring overlay latency stretch

In this simulation the underlying network is modeled to a 7-dimensional Cartesian coordinate space and is constructed to a topology-aware overlay with algorithm Taonet. Fig. 6 shows the cumulative distribution function of overlay latency stretch. It can be concluded that the latency stretch of topology-aware overlay is very low. This is because that the nodes that are close on the overlay network are also close on the underlying.

2. Comparison of latency stretch:

In this simulation we compare the latency stretch of Taonet and the binning strategy proposed in paper [15]. In Taonet the Cartesian coordinate space is 7-dimensiona and the number of landmarks in binning strategy is 12. Fig. 7 plots the latency stretch for increasing overlay size. It can be concluded that the average latency stretch of Taonet is smaller than the binning strategy. The latency stretch of Taonet is not bigger than 2, however the average latency stretch of binning strategy is bigger than 4.

3. Measuring data location latency stretch of TCAN

In this simulation both the physical coordinate space and virtual coordinate space are 7-dimensional, and the number of nodes generated by BRITE is 10k. We generate 1000 queries from random nodes for random data. Fig.8 shows the cumulative distribution function of data location stretch in TCAN. It indicates that most of the data location latencies in the TCAN decrease to less than 30 percent compare with it in CAN. This is because that in TCAN nodes that are close in logical are also close in underlying network, and when a message is routed every node in the routing path routes the message by forwarding it to the neighbor which are the closest node in the underlying network. As a result the total latency of data location is very low.

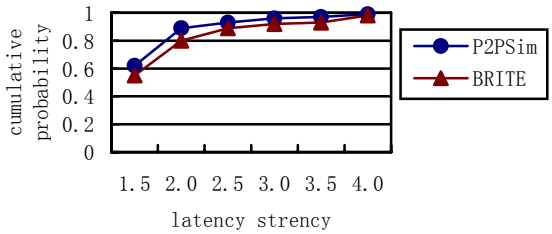

Figure.6. distribution of overlay latency stretch

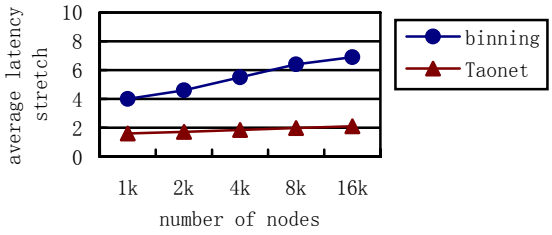

Figure 7. distribution of overlay latency stretch

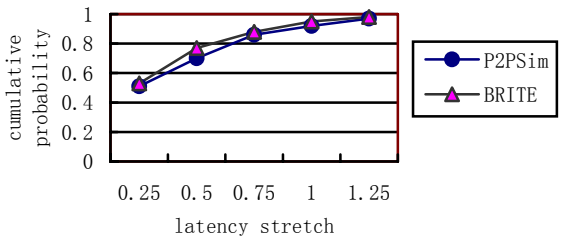

Figure 8. distribution of data location stretch in TCAN

4. Measuring data location latency stretch of TGnutella

In this simulation the underlying network is modeled to a 7-dimensional coordinate space and the number of nodes generated by BRITE is 10k.. The number of requests for data location is 1000 and k=4. Fig.9 shows the cumulative distribution function of data location latency stretch in TGnutella. This figure indicate the data location stretch in TGnutella is very low compare with that in Gnutella. This is because that every node in TGnutella has k/2 neighbors which are close in underlying network, and in the path of routing messages some latencies between two routing hops is

low because them are close in underlying network. These short routing hops result low data location latency.

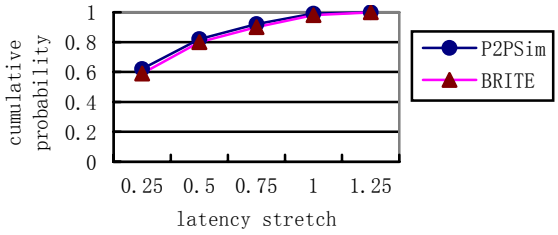

Figure 9. distribution of data location stretch in Tgnutella

7. CONCLUSION

In mass storage systems with P2P infrastructure it is expect to route a request for data more efficiently. This has aroused a great interest in the research of routing requests for data base on location information of underlying network. In this paper, a distributed topology-aware overlays construction algorithm Taonet is proposed, it can be used to construct topology-aware overlay. Nodes that are close in overlay network are also close in underlying network. We apply this algorithm to the construction of topology-aware P2P systems CAN and Gnutella. The simulation results indicate that even rather coarse-grained topological information can significantly improve data location performance in these P2P systems, and the result is independent on the nature of the underlying network topologies.

8. REFERENCES

[1] Napster, "http://www.napster.com,".

[2] Gnutella, "http://gnutella.wego.com," .

[3] FreeNet, "http://freenet.sourceforge.net," .

[4] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in Proceedings of SIGCOMM 2001, Aug. 2001.

[5] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A Scalable Content-Addressable Network," in Proceedings of SIGCOMM 2001, Aug. 2001.

[6] A. Rowstron and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems," available at http://research.microsoft.com/ antr/PAST/, 2001.

[7] J. Kubiatowicz et al., "Oceanstore: An Architecture for Global-scale Persistent Storage," in Proceedings of ASPLOS 2000, Cambridge, Massachusetts, Nov. 2000.

[8] Zhuge, H., Liu, J., Feng, L., Sun, X., He, C.: Queryrouting in a peer-to-peer semantic link network. Computational Intelligenec 21(2) (2005) 197-216.

[9] Saroui, S., Gummadi, K.P.,Dunn, R.J., Gribble, S.D., Levy, H.M.:An analysis of intener content delivery systems. In: Proc. IEEE Fifth Symp. Operating Systems Design and Implementation.(2002).

[10] Sen, S., Wang, J.:Analyzing peer-to-peer traffic across large networks. In: Proc . ACM SIGCOMM Internet Measurement Workshop (2002).

[11] Liu,Y.,Xiao,L.,Liu,X.,Ni,L.M.,Zhang,X.:Location Awareness in unstructured peer-to-peer systems. IEEE Transaction on Parallel and Distributed Systems 16 (2005) 163-174.

[12] S. Ratnasamy, S. Shenker, and I. Stoica. Routing algorithms for dhts: Some open questions. In *Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS'02)*, Boston, MA, Mar. 2002.

[13] M. Castro, P. Druschel, Y. C. Hu, and A. Rowstron, "Topology aware routing in structured peer-to-peer overlay networks," Tech. Rep. MSR-TR-2002-82, Microsoft Research, One Microsoft Way, Redmond, WA 98052, 2002.

[14] Winter, R., Zahn, T., Schiller, J.: Topology-aware overlay construction in dynamic network. (2004).

[15] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker. Topologically-aware overlay construction and server selection. In Proc. 21st IEEE INFOCOM, New York, NY, June 2002.

[16] Z. Xu, C. Tang, and Z. Zhang. Building Topology-Aware Overlays using Global Soft-State. In ICDSC'2003, May 2003.

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

[18] Oguz Kaan Onbilger, Shigang Chen, Randy Chow :A Peer-to-Peer Network Position Architecture. Gainesville, FL 32611, USA, 2004.

[19] M.R.Garrey and D.S.Johnson, Computers and Intractability: A Guide to the Theory of NP-completeness, W.H.Freeman and Company, San Francisco, CA, 1979.

[20] The p2psim project. http://www.pdos.lcs.mit.edu/p2psi-m.

[21] Krishna P. Gummadi, Stefan Saroiu, and Steven D. Gribble. King: Estimating latency between arbitrary internet end hosts. In Proceedings of the SIGCOMM Internet Measurement Workshop (IMW 2002), Marseille, France, November 2002.

[22] A. Medina. A. Lakhina. I. Matta. And J. Byers. BRITE: An Approach to Universal Topology Generation. In Proceedings of MASCOTS 2001. Cincinnati. OH. August 2001.