TB_Chord: An Improved Routing Algorithm to Chord Based on Topology-aware and Bi-Dimensional Lookup Method

Wei Lv[1], Qing Liao[2], Jingling Zhao[3], Yonggang Xiao[1]
Beijing University of Posts and Telecommunications
Beijing, P.R. China
Email: [1]{buptlvwei, xiaoyonggang4106}@gmail.com,
[2]liaoqing@bupt.edu.cn,
[3]jing_lingzh@sina.com

***Abstract*—The research on routing efficiency of DHT P2P networks is a key element to promote the development of P2P networks. One reason is that a node's logical ID is independent of its physical location, bringing tremendous delay to network routing. In this paper, we propose a structured P2P system with low network delay, named TB_Chord, which is extended to enable physical topology aware on the base of Chord. In the TB_Chord, nodeID is configured with prefix called domainID, finger tables are set bidirectional. The most important is TB_Chord neither has super node nor any other plus layer, which guarantees no extra maintenance cost or overhead and original chord's advantages remaining. The routing algorithms, node arrival mechanisms are designed and tested. The results of simulation experiments suggest that the TB_Chord's performance is obviously improved in the delay of routing and the hops of overlay network by contrast with the traditional Chord.**

***Keywords*—chord; topology-aware; Bi-Dimensional**

I. INTRODUCTION

Chord[1] has many salient features over other P2P systems: it is simple and robust; it has guaranteed retrieval, low path length and efficient recovery schemes. Although this overlay system can guarantee that any data object can be located in O(logN) overlay hops on the average, where N is the number of peers in the system, the underlying path between two peers can be significantly different from the path on the overlay [9]. This is because the construction of a P2P network in Chord does not take into account underlying network topology. Therefore, the end-to-end latency of the path can be quite large even though the number of application-level hops is small.

As can be seen, although the target node can be located in a logarithmic overlay hops, the physical path traveled during the overlay routing is often less than optimal. Nearly 70 percents overlay network mismatch with its physical network. Hence, combining DHT and network topology infomation is a hot research field in structured P2P system.

This paper proposes a topology-aware structured P2P system — TB_Chord, whose nodes are configured by prefix, and finger tables are set bidirectional to resolve resource object single directional lookup problem. Therefore, routing process of The TB_Chord takes both identifier space and physical network into account.

The rest of this paper is structured as follows. The related work is presented in Section Ⅱ. Section Ⅲ describes the TB_Chord routing design. We evaluate routing performance of TB_Chord in SectionⅣ. Finally, the conclusion and future works are presented in Section Ⅴ.

II. RELATED WORK

To resolve mismatching between overlay topology and physical network topology in P2P systems, there are many types of approaches proposed.

Proximity neighbor selection is to choose routing table entries to refer to the topologically closest Node among all nodes with nodeid in the desired portion of the nodeid value space. The idea is suitable for Pastry, which has several choice entries in each routing table, and leads to low delay stretch. But this idea does not work for overlay protocols like Chord, whose next hop desired node has a fixed value in the ID value space.

PChord[3] achieves better network proximity than the original Chord by using proximity lists, i.e. the list of the close nodes that a node discovers in its lifetime. The next hop is decided by the entries in both the proximity list and finger table. Although this approach achieves better routing efficiency than Chord while keeping lightweight maintenance costs, it has problems of slow convergence and inefficiency in the case of churn, where the lifetime of a node in the overlay is relatively short. And also, when number of nodes is large enough, the performence of PChord get worse dramatically.

Topology-based clustering approaches include Brocade[4] and central controlling topology-based clustering[5]. The Brocade effort adds a new overlay layer to the original structured P2P networks. Supernodes are configured to manage nodes in AS (Autonomy System) and route messages among intra-AS to improve the performance of P2P routing.

978-1-4244-3693-4/09/$25.00 ©2009 IEEE

However, supernodes make self-organization and load balancing more difficult. A supernode is supposed to have strong computing capability. Additional protocols should be integrated into routing algorithms to process messages among supernodes and normal nodes.

The topology-based clustering in centralized mode is easy to practice. With the rapid development of the scale, the more critical computing ability of central clustering server is required, which unavoidably causes the clustering server as a bottleneck in the system.

As discussed above, the methods mentioned neither need super node to supply the physical information nor destroy original DHT systems' advantages.

III. CORE DESIGN

We assume there are twelve P2P clients in the Chord ring; their real physical location appears like Fig 1.

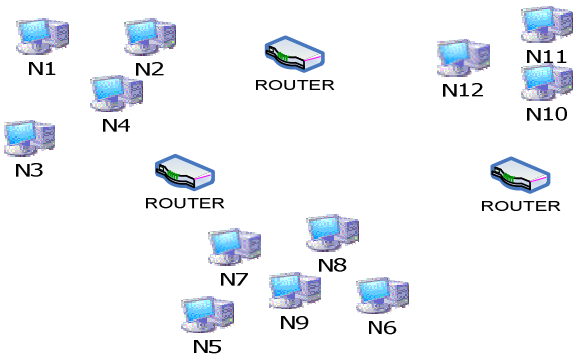

Figure 1. Physical location

If using traditional chord mechanism, the overlay chord ring maybe like Fig 2. We can see even though two clients' locations are very close, they may contact each other through many overlay hops and even if number of hops between two nodes in the overlay is one, their physical location may be very far.

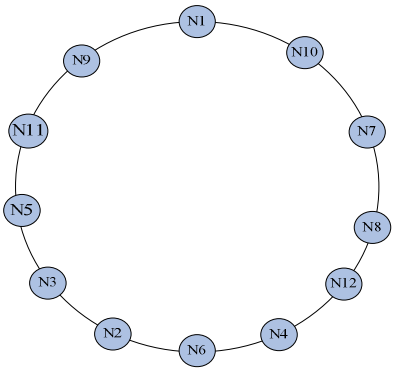

Figure 2. Traditional overlay

However, using TB_Chord, the overlay ring will look like Fig 3, which takes clients physical location into account.

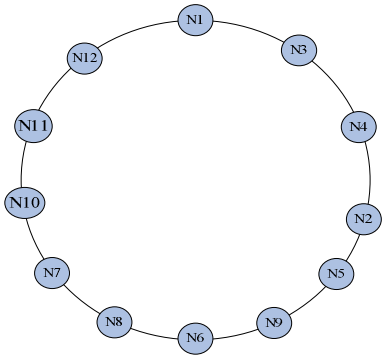

Figure 3. TB_Chord overlay

A. Core Concept

Landmark: For the sake of making use of topology-aware information in TB_Chord routing desigh, landmark+RTT approach [10] is used to help building TB_Chord's topology. Prior to joining the overlay network, a joining node has to measure its RTT to all landmarks. Then it gets the nodeID's prefix, which means and dertermines the joining node's domain in TB_Chord ring. Here, landmarks just supply topology information and have nothing to do with DHT or P2P function.

Leveled domain: Based on geographical location, TB_Chord divides the entire network into k-levelled domains. Nodes in the same level domain, means these nodes are in the same physical location at some level. Smaller the level is, bigger the domain's size is. For example, as Fig 4 shows we divide the entire internet into 3 levelled domain: level 1 is WAN, level 2 is MAN and level 3 is LAN. When a lookup flow begins, it will search the wanted resourceID in Level 3. If Level 3's domain doesn't have the resource, the request message will spread in Level 2 which contains many Level 3 domains. The same way as above until the entire network is searched.

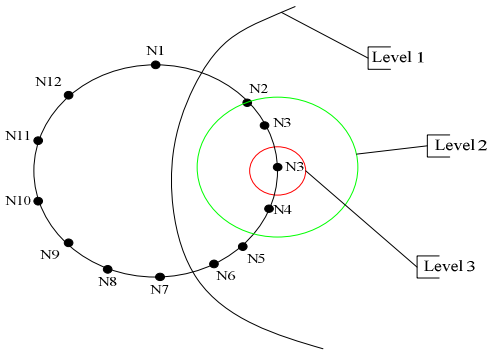

Figure 4. Levelled domain

NodeID: As shown in Table 1, each node in TB_Chord ring has a unique identifier named nodeID. It consists of domainID and a hID. DomainID represents node's physical infomation which determines its location in TB_Chord ring, while hID is generated by mapping node's IP address into an identifier of m bits using a hash function such as SHA-1. We can assign proper number of bits to every level according to how many

domains and how many levels do you have. Level 1 domain's scope is larger than Level 2 domain and Level 1 domain may contain many Level 2 domains since Level 1 bits are in the highest position.

TABLE I NODE ID AND RESOURCE KEY

DomainID			Random m bits
Level 1	Level 2	Level 3	hID

ResourceID: The same as NodeID composed of domainID and a hID. But the most important thing is: one resource may have two or even more resoureID when it's searched among different domains. Take the resource MP3 "Country Road" as an example; their m random bits generated by SHA-1 are same while their domainIDs are different if they are in different domains.

Bi-Dimensional Lookup: proposed by [2], to resolve the single directional lookup mechanism's drawback.

Finger talbe: Each node in TB_Chord has two finger tables. One is traditional finger table, the other is R_finger table. Difference between them is, when TB_Chord's node receives a routing message it compares the resourceID and its own nodeID to determine whether finger table or R_finger table to use. If the resourceID is greater than nodeID, finger table is used, while nodeID is bigger than resourceID, R_finger table is looked up.

B. Resource Lookup Flow And New Joining Flow

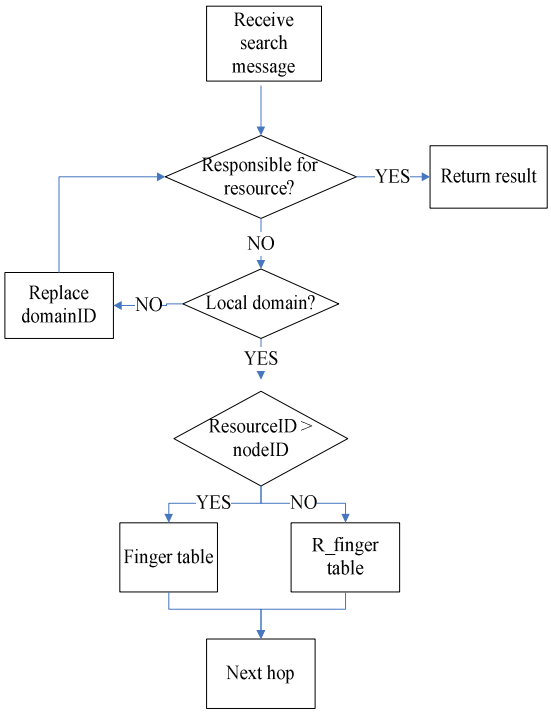

Figure 5. Resource lookup flow

As shown in Fig 5, when a node receives a resource search message, the processing flow is shown in Figure 6:

1) The node checks resourceID table scope to see if it is responsible for this resource, if yes, return the result to searcher;

2) if not, node gets resourceID's domainID to confirm if it belongs to the node's domain;

3) if yes, node compares the resourceID and its own nodeID to determine which finger table to use; if not, replace the resourceID's domainID with the node's domainID and return to step2.

4) lookup the chosen finger table (R_finger table or finger talbe) to get the next hop's IP:port.

C. *New Node Joining Flow*

When a new node wants to join the TB_Chord ring, the processing as Fig 6 follows:

1) according to its own configure, use command PING to get RTT value from all the landmarks;

2) obtain the domainID from the smallest RTT landmark to determine which domain it will get in

3) get joining necessary information from the bootstrap node which belongs to the same domain as new arrival node

4) the same as traditional joining procedures

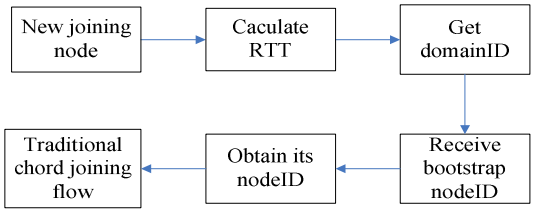

Figure 6. Joining flow

D. Leaving flow , failing flow , maintenance and other respects are the same as traditional Chord.

IV. PERFORMANCE EVALUATION

In P2P overlay network, average routing hop and average routing delay are indicator of routing performance. The average routing hop is the average forward times invoked by a routing process and the average routing delay is the average time delay over the path from source node to target node for routing process. In order to evaluate the performance of the TB_Chord, we perform large-scale experiments by simulation to the TB_Chord and traditiional Chord, respectively. We constructed our simulator on Omnet++ which is a public-source, component-based, modular and open-architecture simulation environment with strong GUI support and an embeddable simulation kernel. There are six different network topologies in our experiment and all of them include 4 levelled domains. During this simulation, 60 query requests are invoked per node and destination node are generated at random.

A. Routing hop in TB_Chord

Fig 7 shows routing hops comparison between TB_Chord and traditional Chord. We set the number of nodes 1000, 4000, 8000, 12000, 16000, and 20000. From the fig, we can conclude that,

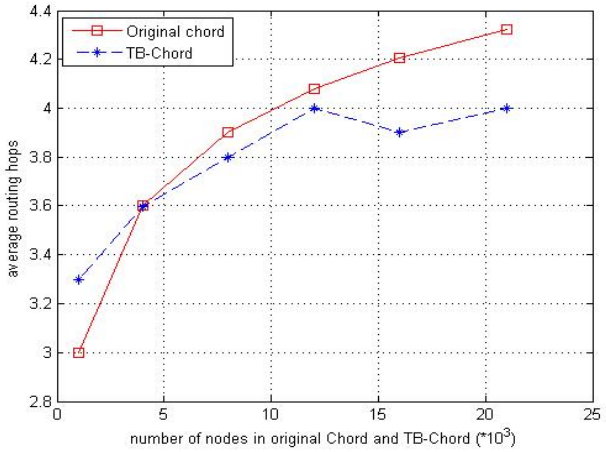

Figure 7. Routing hops in TB_Chord and Chord

1) when the number of nodes is smaller than 4000, traditional Chord routing hops are smaller than TB_Chord. This is because traditional Chord routing hops follow O(logN). When N is not big enough, the O(logN) is small; while nodes in TB_Chord need transfer domain message.

2) when the number of nodes is bigger than 4000, TB_Chord's routing performence is better than traditional Chord. Since when number of total nodes grows large, number of every domain grows larger and number of resource becomes bigger too.

3) number of nodes between 12000 and 20000, the fig looks like a parabola with a lowest point at 16000. The resean is TB_Chord does'nt change the traditional lookup method within every domain.

B. Routing delay in TB_Chord

Fig 8 shows routing delay comparison between TB_Chord and traditional Chord. The set is same as hop comparison. TB_Chord routing delay is shorter than traditional Chord which means our design's routing performence is better than the original. This is because TB_Chord take topology information into account when building chord ring, so when node search resource it looks up locally first.

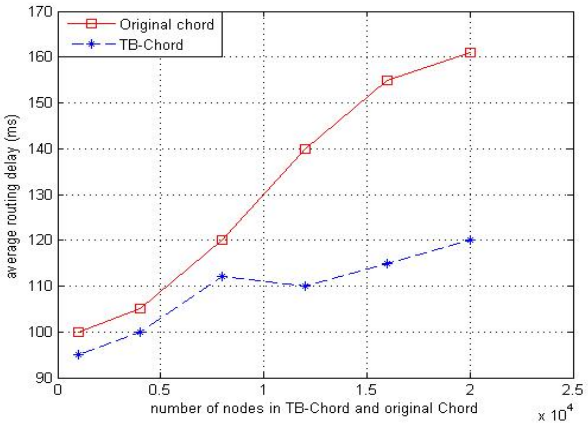

Figure 8. Routing delay in TB_Chord and Chord

At last, TB_Chord ring does not have any supernode and does not change the original chord's maintenance mechanism. So the keepalive overhead and maintenance overhead remains the same as original chord.

V. CONCLUSION AND FUTURE WORK

To deal with the routing delay and ignorance of physical topology infomation in P2P, this paper presents a topology-aware structured P2P system, named TB_Chord, which applies a suit of mechanisms to extend Chord to optimize the utilization of both information- physical network topology and overlay network. In the TB_Chord, nodes are configured with prefix according to their levelled domain; the related finger tables are set bidirectional and the entry is pointed to the nearest nodes in the special domain. The system facilitates the landmark+RTT method to generate topology information. The TB_Chord makes effectively use of network topology structure during network routing. The result of experiments show that the TB_Chord, compared with traditionalChord, has obvious improvements in the routing delay and the hops of overlay networks. At the same time, TB_Chord does not need super node or plus layer which will pay more maintenance costs.

Our work provides an initial study of the TB_Chord system. Although simulation results are encouraging, there are many open problems to be addressed. We do not consider the mobility when node get its domainID. Whereas it is very important in DHT and we will address this problem in the future. We will also present the hotpiont problem in TB_Chord.

REFERENCES

[1] I. Stoica, R. Morris, D. Karger, M. Kaashoek, and H.Balakrishnan. "Chord: A scalable peer-to-peer lookup service for internet " IEEE/ACM Transactions on Networking, Vol.11, No.1, February 2003

[2] Hongwei Chen, Zhiwei Ye. "BChord: Bi-directional Routing DHT based on Chord"School of Computer Science and Technology, Hubei University of Technology, Wuhan, P.R. China..

[3] L. H. Dao, J. Kim. AChord: Topology-aware Chord in anycast-enabled networks. In Proceedings of the 2006 International Conference on Hybrid Information Technology (ICHIT 06), 2006..

[4] B.Y. Zhao, Y. Duan, L. Huang, A.D. Joseph, and J.D.Kubiatowicz. Brocade: Landmark routing on overlay networks. In IPTPS'02. Protocol .

[5] B.Krishnamurthy,J.Wang,Y.Xie, Early Measurements of a Cluster-based Architecture for P2P Systems, Proceedings of the ACM SIGCOMM InternetMeasurement Workshop, 2001..

[6] F. Hong, M. L. Li, M.Y. Wu, J.D, Yu. PChord: Improvement on Chord to Achieve Better Routing Efficiency by Exploiting Proximity. In IEICE Transactions on Information and Systems, v E89-D, n 2, Feb. 2006, pp. 546-554..

[7] F. Dabek, M. F. Kasshoek, D. Karger, R. Morris, I.Stoica. Wide-area cooperative storage with CFS. In Proceedings of ACM SOSP, 2001..

[8] Jung-Heum Park and Kyung-Yong Chwa. Recursive circulant: a new topology for multicomputer networks. In Proceedings of the International Symposium on Parallel Architectures, Algorithms, and Networks (ISPAN '94), pages 73–80, 1994..

[9] K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim. A survey and comparison of peer-to-peer overlay network schemes. IEEE Communications Survey and Tutorial, 2004

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