BLC：A Scalable peer-to-peer Lookup Service and Data Sharing Architecture

BLC：可擴展式點對點查詢與資料分享架構

Chia-Ming Sung, Chi-Hung Chen, Wen-Nung Tsai
Department of Computer Science and Information Engineering,
National Chiao-Tung University
{*chiaming, chihung, tsaiwn*}*@csie.nctu.edu.tw*

Abstract

The peer-to-peer applications become more and more popular. There are many different kinds of peer-to-peer architectures. However, most of them only focus on how to search data and/or how to maintain neighbor relations. Unfortunately, some hosts selected to be hot points (via points) may result in low performance due to heavy overhead.

In order to solve these problems, we propose a new P2P architecture, BLC, which is based on Chord[3]. In BLC, we also pay more attention to the transmission and search over this logical topology. Experimental results show that our approach can improve the efficiency of query and data transmission and thus reduce the network resources consumption.

Keyword: P2P, data sharing

中文摘要

點對點的應用軟體日漸普及，目前有許多不同種類的點對點架構，然而大部分的架構都著重於資料查詢或者如何維護與其他用戶之間的溝通，往往沒有考慮到部分用戶端因被選為熱點(中繼點)而造成該用戶系統網路壅塞及效能低落。

本論文提出的BLC是修改自既有點對點架構Chord[3]，試圖解決上述問題。另外，我們額外在查詢與傳輸著墨，以提升有效查詢、有效傳輸、減少網路資源浪費。

關鍵詞: 點對點傳輸, 資料分享

1. Introduction

As the users' equipments and the network bandwidth get much progress, P2P transmission applications become more and more popular. Among P2P applications, there are three main architectures, index server, distributed service, and distributed hash table (DHT). However, all these pay attention to data searching and don't consider the "via point" problem. When a node is selected to be a via point, the busy communication and transmission works decreases the performance on this node. If this node departs, it costs much to recover. In addition, since the users' equipments are getting more powerful than before, it is beneficial to use their resource to help packet routing and decrease network traffic.

Hashing is a good search method for large data. In this paper, we apply hashing not only to the data searching but also to the data transmission in P2P network. Besides, we add proxy concept, Bittorrent sharing concept and application layer routing in our approach to increase data delivery efficiency.

2. Background

In the following, we will briefly introduce the classification and some popular instances of P2P applications. We also describe the concept of locality and how it makes the logical network close to the physical network in order to help data delivery.

2.1 Peer to Peer(P2P) Applications

In P2P applications, they construct a logical network over the physical network. Emule[11], Edonkey[12] and Bittorrent[9] are popular P2P applications for file downloading. The main idea is to share peers' data files.

Searching is the most important functionality in P2P applications. Most of them except Bittorrent provide search in their friendly user interface, and users can find the desiring files easily. In Bittorrent, users have to find the seed of desiring files by

themselves. Searching depth, searching time, data availability, and searching bandwidth are the criteria for their performance.

2.2 P2P Architecture Generations

In this section, we classify the P2P architecture by time into Client and Server, first generation, second generation, third generation and beyond 3rd generation.

Client and Server

Client and server architecture is the first file transmission architecture. The client and server would communicate with each via the protocol and port on which both of them agree. FTP is one of the common applications of this type.

First Generation

There are index servers to assist data sharing in first generation P2P architecture. This kind of P2P system is composed of index servers and peers. Index servers provide central indexing and then peers can know all data shared by peers which connect to this system. Data is shared by peers, and index servers don't keep those data files. This is the main difference to client and server architecture. Napster[8] is one of the first generation P2P systems.

Second Generation

In second generation, there are no index servers. Each peer maintains a local index table of its own sharing data information. The concept of neighbor helps exchange the sharing data information. This generation is also called the distributed service.

Gnutella[15] the first system of this distributed service. The query message is sent to all neighbors and each neighbor forwards this query message to all its neighbors. The forwarding keeps on and results in Query Flooding. FastTrack[14], which is the fundamental technique of KaZaA[18], let the query messages be sent among peers. Responses are sent back only when the file is found. It is obvious that search loop may occur and that a large number of queries may downgrade the network performance.

Third Generation

In the third generation P2P systems, they add the Distributed Hash Table(DHT) to send query messages in systematic approach. Each peer has a unique node number. Every shared file has a unique data number which can be map to the node number which represents the node that is responsible to this data.

In a query process, the peer calculates the data number first. Then it gets the node number by mapping. The query message is sent to the neighbor whose node number is the most closest to that owner peer. It avoids the query flooding. There is no need to limit the search depth and then the Global Search can be dealt with. Furthermore, choosing a suitable neighbor can decrease the hops counts and increases the performance.

EDonkey[12], Kademlia[5] and Morpheus[20] are the third generation P2P systems. According to the report of BayTSP[10], EDonkey has more users than KaZaA does, and it becomes the most popular P2P transmission software in the world.

Besides those we have described, there are other architectures, such as KaZaA and Bittorrent [9].

KaZaA

KaZaA is a closed architecture application. But J. Liang, R. Kumar and K.W. Ross presumed its architecture by analyzing its behavior in 2002[4]. In this architecture, peers(nodes) are classified according to their transmission ability. There are Ordinary Node (ON) and Super Node (SN).ON's are general peers and SN's are peers with high network bandwidth.

Bittorent

Bittorrent does not provide searching functionality. Users should find the SEED (*.torrent), which includes the Trackers' position, original source position and data hash value, and then they can download the files. Trackers assist in searching the files by recording that which peer has the segments of this file.

2.3 Locality

Locality is the concept which is used to make the logical network close to physical network. Because the data search time includes the transmission delay of forwarding the request, to fast the search process we can attempt decreasing the transmission delay affected by physical network environment.

When constructing 3rd generation P2P network, if the node number is given randomly, two nodes that have near node numbers may have their position far away in real network. If applying the locality to P2P network, it makes the peers with adjacent node numbers indeed closed to each other in reality. So that decreases the transmission path and time.

3. Related Works

Now, we're going to introduce some relative researches, RI[1], CAN[6], Chord[3] and Pastry[2].

Since the second generation, peers who know more neighbors get more searching result. Routing Index (RI)[1] introduces how to choose the most suitable neighbor to forward query message. In RI, it divides data into groups and calculates a value to be the choosing basis. RI topology is in tree architecture.

Each peer is a node in this tree. By features of the tree, query loop problem can be solved.

When to stop searching for a file is the key to decrease the searching traffic. CAN[6] is designed to solve this problem. It turns a searching problem to a positioning problem by hashing. Data file is stored in a specific place that is compute from data key by hashing. Therefore, peers can get the data logic position fast.

Chord[3] topology is a one dimension ring. It decreases the searching length by binary search. Therefore the hop counts for searching is limited to $\log_2 N$ (N is number of nodes).

Pastry[2] is a variety of CAN. It is a one dimension ring in topology. Pastry adopts the idea of common prefix to decrease the searching length.

4. BLC System Architecture

Based on Chord, we propose a Bi-direction Locality Chord (BLC). We add bi-direction search and locality concept into Chord. Besides that, BLC has trackers like BT.

4.1 System Overview

In BLC system there are RTT(Round Trip Time) servers that assist to divide peers into groups(regions) within which the delivery of messages and data files is fast. The logical network topology is ring and peers are the nodes on it. This logical ring network is divided into several logical subnetworks, which represent physical regions.

Nodes in the same logical subnetwork are closed to others not only logically but also physically. In Figure 1 and Figure 2, the logical space size is 2^n (n > 3). Every region has at most eight peers. Each peer must maintain a finger table, which in Chord means routing table, to make the P2P network function well.

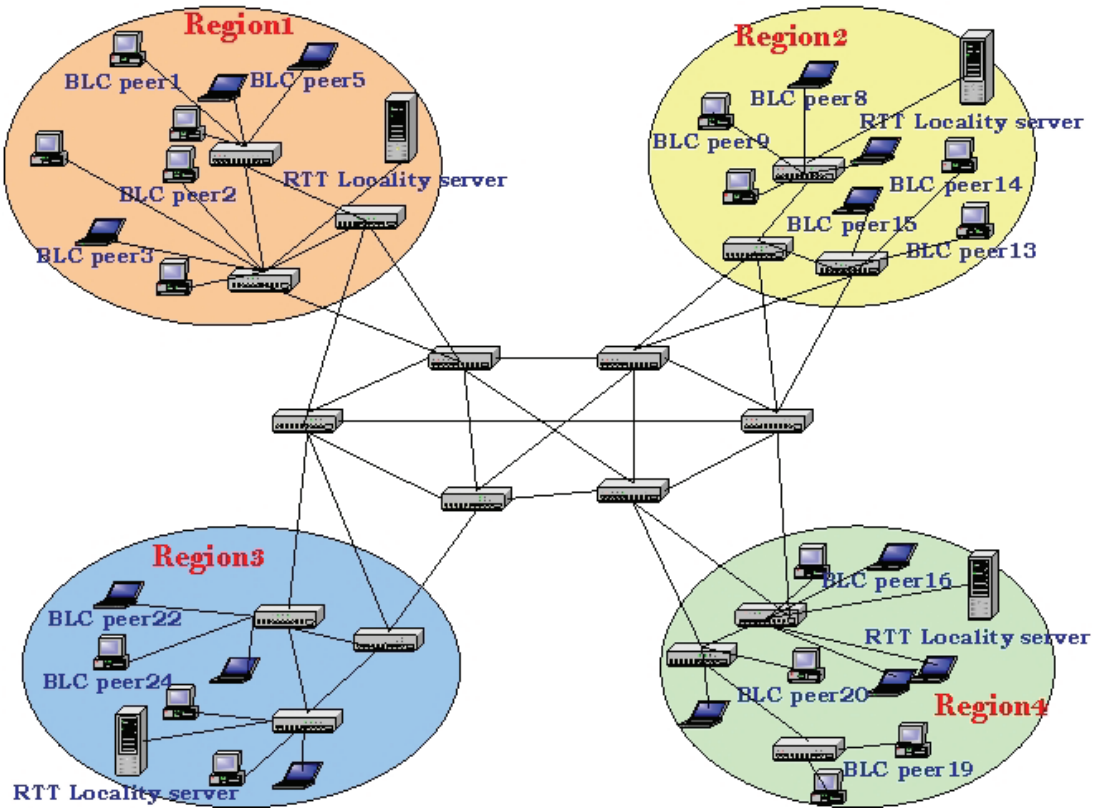

Figure 1 Physical network of BLC

4.2 Assumptions and Detail Design

BLC architecture is based on Chord and is added in Bi-direction search, Active Locality Proxy, and Tracker transmission. We assume that there are 8 logic subnetworks in our BLC logic network. Clients' node numbers are arranged between 0 and 2^n–1 (n>3), that can be represented by n bits.

4.2.1 Bi-Direction Search

Bi-Direction Search (BDS) is used to make same interval of node numbers have same hop counts. In this paper, we change the content in finger table to help bi-direction search.

The finger table in BLC has three fields, interval, start, and successor, just as Chord. The interval is a range of nodes within one logical subnetwork. The numbers representing in interval are node numbers. Start is the start node number in this interval. Successor is the node which is responsible for this interval. Bi-direction search can be achieved by forward and backward intervals. In Chord, the interval is $[(r+2^k) \bmod N, (r+2^{(k+1)}) \bmod N)$, where r is its node number and $0 \le k < \log_2 N$. In BLC, we have the forward interval concept like Chord does, and we also have backward interval$[(r-2^k) \bmod N, (r-2^{(k+1)}) \bmod N)$, where $0 \le k < \log_2 N-1$. We have total $2 * (\log_2 N - 1)$ intervals. ◦

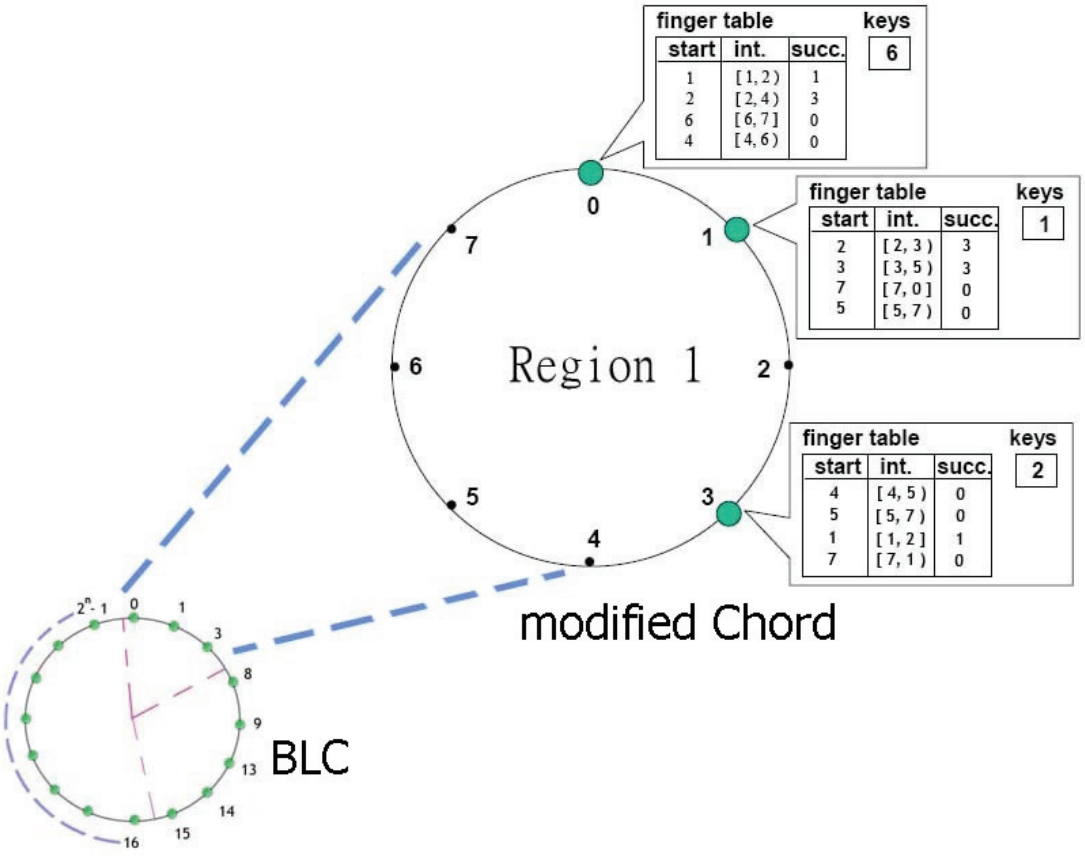

Figure 2 Logical network of BLC

Table 1 Finger table intervals in Chord and BLC

rows	Chord interval	BLC interval	Modified BLC interval
1	[1, 2)	[1, 2)	[1, 2)
2	[2, 4)	[2, 4)	[2, 4)
3	[4, 8)	[4, 8)	[4, 8)
4	[8, 16)	**[8, 16)**	**[8, 16]**
5	[16, 0)	[31, 30)	[31, 30)
6	X	[30, 28)	[30, 28)
7	X	[28, 24)	[28, 24)
8	X	**[24, 16)**	[24, 16)

In the case of n=32 and r = 0, as described in Table 1, BLC changes the interval of **[8,16)** into **[8,16]**, otherwise the node r=16 is lost.

When searching for a data, peers can easily decide to hop by forward interval or reverse interval. In the general case, hops are not more than Chord as shown in Figure 5.In some cases, the number of hops is decreased (Figure 3 and Figure 4).

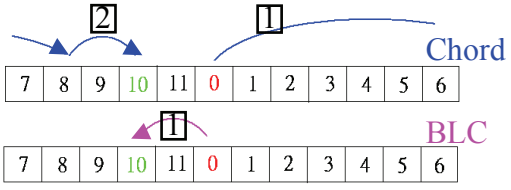

Figure 3 Comparison of BLC and Chord(1)

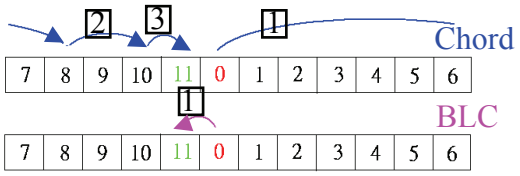

Figure 4 Comparison of BLC and Chord (2)

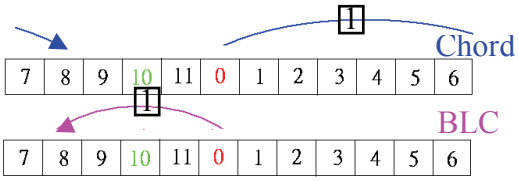

Figure 5 Comparison of BLC and Chord (3)

4.2.2 Active Locality Proxy

Proxy is generally used to increase transmission efficiency and decrease the loading of via points. How to distribute the proxy servers is an important issue. In our approach, we adopt the concept of locality. According to CAN that suggests two dimensions with four regions. In fact, the two dimensions of CAN are two layered one dimension rings. Therefore, we design BLC as one dimension ring with eight regions. Each region has an active locality proxy server.

The nodes are numbered as follows. We divide the node number into two parts: 3 most significant bits for region number and the other n-3 bits for peer number. Peer number is got from hashing.

When data is shared, the node responsible for this data in every region will receive the shared data information. These responsible nodes for this shared data have the same peer number. If the responsible node does not exist, the successor turns to be the responsible node. When searching for this data, peers just search in their region.

The locality concept can decrease the search loading on via points and decrease the probability of loosing all information owned by one peer departed.

4.2.3 Tracker Transmission

In BLC, a peer is also the tracker of the shared data relative to its peer number. Trackers are responsible for helping the shared data transmission. Since trackers are distributed in every region, the loading of trackers is decreased.

4.2.4 Client Software Architecture

BLC is a 3rd generation P2P architecture. There are nodes only and each node works the same. In the design of software architecture, BLC is between application layer and TCP/IP layer. There are four main components in BLC, Transmission Manager, Routing Manager, Tracker Manager and Share Data Manager.

Communication messages are received and sent by **Message Controller**. If it is a network topology update message, Message Controller will send it to the Routing Manager after analyzing. If it is a metadata update message, it is passed to Tracker Manager.

Data delivery is taken care of by **Data Controller**. In the case of uploading, all data requests and the communication to Share Data Manager are handled by Data Transmission Controller. It should manage all data segments downloaded, check for segment integrity, and decide if retransmission or recovery is needed.

Routing Manager

Searching data is the most important functionality in P2P environment. Routing tables are used to make the query messages forwarding effective in searching procedure. In this paper, we construct a routing manager to maintain the routing table. The routing table is updated depending on the result of heartbeat monitor and on the update messages received. In order to make sure that the whole logic network functions well, heartbeat monitor is periodically enforced by checking if the successor still exists. If the successor responses with informing a better successor, it triggers a series of updates. When the successor leaves, this peer tries other successors and also triggers a series of updates.

In Chord, there are $(\log_2 N)^2$ messages in the update procedure. In our approach, it needs $(2*(\log_2(N/8)-1)^2+7)$ because of the additional communication with other localities to achieve double direct search.

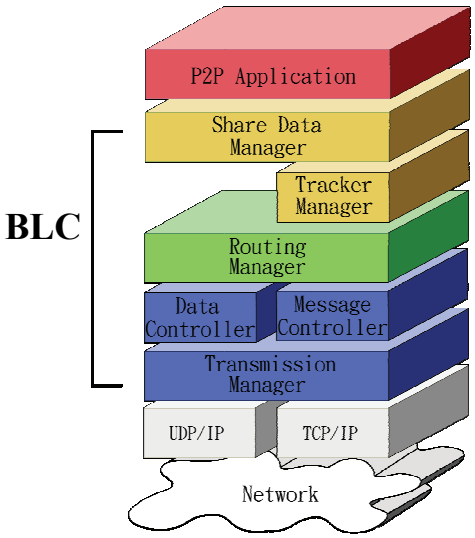

Figure 6 BLC client software architecture

Tracker Manager

Because a peer is also a tracker, Tracker

Manager should handle the tracking works. When a peer joins BLC, there are metadata and query packets sent to it. If the peer has the metadata for the query data, it replies with the node who owns this data. Tracker Manager also keeps communicating with the querying peer to know if the querying peer can be a new data source and update its metadata. In addition, Tracker Manager sends metadata update messages to corresponding peers in each region periodically.

Share Data Manager

Share Data Manager manages the data this peer shares. The main job is to generate metadata and to spread to every region responsible peer. This is why BLC can do global search.

Share Data Manager divides one single data file into several data segments. A data file is received by parts from peers who have any complete data segments.

Main Operation Messages

The main operations in P2P transmission architecture are update, join, depart, lookup share. In BLC, there are five operation messages: Look Up, Update, Share, Join and Depart. The following is the description for them.

- **Look Up**

Lookup is the way to find out shared data in P2P network. In BLC, peers first lookup for successor in its region, and then query to the successor.

Before lookup, peers should use the key feature (ex, file name) of data to compute the peer number where metadata would be. By the content in finger table, peers know the successor of that region. The lookup message is sent to the responsible peer eventually.

- **Update**

Update messages is used to maintain the peer sequence and the relation between peer and successor on logical network. Otherwise the routing path of searching may be too long. Through the periodic update messages, the content of finger table keeps updated.

- **Share**

Share means the new peer shares the data to the network. The share messages are of two types, shareupload_inner and shareupload_outer. shareupload_inner means the owner of data uploads the metadata to the responsible peer in the same region. shareupload_outer means the responsible peer upload the metadata to other region responsible peer.

- **Join**

Join means a new peer is coming in BLC. This new peer must know the information about its successor provided by a peer that is already in BLC network. Then the new peer sends update messages to all successors and these messages will trigger a series of update operation. After all, the new peer spreads out the data information it shared

- **Depart**

Peers may depart from the P2P network normally or abnormally. Peers that depart normally cause a series of update messages for other peers to find successor. Peers may depart unexpectedly. Through heartbeat monitor, peers can sense that other peers departed and trigger updates. In this paper we assume that peers depart in normal way.

5. Experimental Results

This chapter will show how BLC enhances the performance of Chord by simulations. We design a simulator, P2PNS, for P2P behavior and collect the statistic data about search, transmission and loading.

5.1 P2PNS introduction

P2PNS(Peer-to-Peer Network Simulation) is composed of P2P Network Framework, Action Generator and Clock.

P2P Network Framework is the basis of our simulation. Action Generator generates reasonable events, including join, depart, share and lookup. In every time unit, it generates several events respectively. Clock is responsible for time synchronization in the virtual P2P environment.

P2PNS is developed by JAVA version j2sdk1.4.2_07 and IDE tool is Eclipse Platform Version 3.1.0.

5.2 Criteria

The performance is evaluated in three aspects, search, transmission and load.

Search success rate =

$$\frac{search\_success\_times}{total\_search\_times}(\%)$$

Search path length =

$$\frac{total\_search\_path\_length}{total\_search\_times}$$

Transmission success rate =

$$\frac{transmission\_success\_times}{total\_transmission\_times}(\%)$$

Transmission progress =

$$\frac{transferred\_size}{total\_size}(\%)$$

Router transmission load =

$$\frac{router\_transferred\_packets\_count}{routers\_count}$$

Host search load =

$$\frac{total\_searchs\_count}{hosts\_count}$$

5.3 Performance Evaluation

We set the numbering space 2^{14}. Data number and node number are 14 bits in length. We experimented with BLC and Chord of 128, 256…, to 1024 peers, 20 times for each. Join, depart, share and lookup events are generated dynamically by Action Generator. Then, collect the results. The hardware specification is showed in Table 2.

Table 2 Hardware platform

Number	CPU	Ram	OS
1	Intel Pentium4 2.8 (HT)	1GB	Windows XP SP2
9	AMD Athlon(TM) XP 2600+	2GB	RedHat Linux 9.0 (Kernel 2.4.27)
2	AMD Athlon(TM) XP 2600+	1GB	RedHat Linux 9.0 (Kernel 2.4.27)
1	AMD Athlon(TM) XP 2600+	1.5GB	RedHat Linux 9.0 (Kernel 2.4.27)
3	AMD Athlon(TM) XP 2600+	512MB	RedHat Linux 9.0 (Kernel 2.4.27)

Assume search path length, router transmission load and host search load of 128 peers BLC be 1 and transform other corresponding data. The result is show in Table 3.

BLC has higher search success rate than Chord does. Active locality proxy makes BLC perform better than Chord does when peers depart. We also see that BLC transmission progress better because of the active locality proxy and tracker.

6. Conclusion

We conclude with the following comparison table of Chord, CAN, Pastry and our approach.

In BLC, it costs higher on updates and finger table size than CAN. But in the aspect of search path length, BLC is much better than CAN. And BLC is an enhancement of Chord except for the finger table. It is trade-off between search path length and finger table size.

BLC is a feasible scheme. It uses simple protocol to maintain the P2P network. For search, BLC is at least as good as other architectures. For transmission, it decreases the load of peers and network.

In this paper, we propose the BLC architecture to enhance Chord with bi-direction search, active locality proxy and tracker. And by using some peer resource, it increases the performance of BLC P2P network.

Table 3 Experimental results

	Chord					
	search		transmission		load	
nodes	success(%)	length	success(%)	progress	router	host
128	0.90	1.07	0.84	0.86	1.01	1.04
256	0.90	1.10	0.64	0.66	1.04	2.51
384	0.88	1.13	0.49	0.52	1.06	4.36
512	0.84	1.17	0.48	0.50	1.10	3.79
640	0.85	1.21	0.41	0.44	1.12	4.87
768	0.85	1.24	0.36	0.39	1.14	6.36
896	0.82	1.26	0.33	0.36	1.16	7.09
1024	0.81	1.28	0.32	0.34	1.19	7.46
	BLC					
	search		transmission		load	
nodes	success(%)	length	success(%)	progress	router	host
128	0.93	1.00	0.94	0.95	1.00	1.00
256	0.91	1.03	0.68	0.71	1.00	2.40
384	0.89	1.05	0.53	0.56	1.01	4.05
512	0.86	1.06	0.54	0.56	1.01	3.43
640	0.87	1.06	0.46	0.48	1.01	4.51
768	0.89	1.06	0.39	0.42	1.01	5.78
896	0.87	1.07	0.36	0.38	1.02	6.56
1024	0.88	1.07	0.34	0.36	1.01	6.44

Table 4 Comparison of Chord, CAN, Pastry and our approach

	Our approach	Chord	CAN	Pastry
Parameter	Num of Area g	None	Dimension d	Base b
Search path length	O(log(N/g))	O(logN)	$O(dN^{1/d})$	$O(\log_2{}^b N)$
Finger table size	2* (log(N/g) - 1) + (g-1)	logN	2d	$(2^b\text{-}1)\log_2{}^b N$

Reference

[1] A. Crespo and H. Garcia-Molina. Routing Indices For Peer-to-Peer Systems. In Proceedings of the 22nd International Conference on Distributed Systems, pages 23--32, Vienna, Austria, 2002.

[2] A. Rowstron and P. Druschel, "Pastry: Scalable,

distributed object location and routing for large-scale peer-to-peer systems". IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), Heidelberg, Germany, pages 329-350, November, 2001

[3] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan, Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications, ACM SIGCOMM 2001, San Deigo, CA, August 2001, pp. 149-160

[4] J. Liang, R. Kumar and K.W. Ross, "Understanding KaZaA," submitted, 2004

[5] P. Druschel, F. Kaashoek, and A. Rowstron (Eds.): IPTPS 2002, LNCS 2429, pp. 53–65, 2002. Springer-Verlag Berlin Heidelberg 2002

[6] S. Ratnasamy, P. Francis, M. Handley, R. Karp, amd S. Schenker, "A Scalable Content-Addressable Network," In Proceedings of SIGCOMM 2001, ACM. pp. 168-175.

[7] Yang-Hua Chu, Sanjay G. Rao, and Hui Zhang, "A case for end system multicast," in ACM SIFMETRICS, 2000, pp. 1-12

Links

[8] Napster: http://www.napster.com/

[9] Bittorrent: http://www.bittorrent.com/

[10] Baytsp: http://www.baytsp.com/

[11] Emule: http://www.emule-project.net/

[12] Edonkey:http://www.edonkey2000.com/

[13] Ezpeer: http://www.ezpeer.com/index.html

[14] FastTrack: http://www.slyck.com/ft.php?page=1

[15] Gnutella: http://www.gnutella.com/

[16] Gnutella PURE PEER TO PEER PROTOCOL http://www.ece.rutgers.edu/~parashar/Classes/01-02/ece579/slides/gnutella.pdf

[17] ICQ: http://www.icq.com/

[18] KaZaA: http://www.kazaa.com/us/index.htm

[19] Kuro: www.kuro.com.tw/

[20] Morpheus: http://www.morpheussoftware.net/

[21] MSN messenger: http://www.msn.com/

[22] Pastry Routing Table http://www.scs.cs.nyu.edu/V22.0480-005/notes/l24.pdf

[23] Skype: http://www.skype.com/

[24] SkypeOut and SkypeIn Gateways http://463west.blogspot.com/2005/05/skypeout-and-skypein-gateways.html

[25] U.S. Court of Appeals for the Ninth Circuit http://www.ce9.uscourts.gov/web/newopinions.nsf/0/c4f204f69c2538f6882569f100616b06?OpenDocument

[26] WinMax: http://www.agry.purdue.edu/max/

[27] Yahoo Messenger: http://www.yahoo.com/