http://www.zju.edu.cn/jzus; E-mail: jzus@zju.edu.cn
ISSN 1009-3095 *Journal of Zhejiang University SCIENCE* V.5, No.1, P.16-21, Jan., 2004

Super-proximity routing in structured peer-to-peer overlay networks*

WU Zeng-de(吴增德)†, RAO Wei-xiong(饶卫雄), MA Fan-yuan(马范援)

(*Department of Computer Science & Engineering, Shanghai Jiaotong University, Shanghai 200030, China*)

†Email: wu-zd@cs.sjtu.edu.cn

Received Dec.3, 2002; revision accepted Mar.10, 2003

Abstract: Peer-to-Peer systems are emerging as one of the most popular Internet applications. Structured Peer-to-Peer overlay networks use identifier based routing algorithms to allow robustness, load balancing, and distributed lookup needed in this environment. However, identifier based routing that is independent of Internet topology tends to be of low efficiency. Aimed at improving the routing efficiency, the super-proximity routing algorithms presented in this paper combine Internet topology and overlay routing table in choosing the next hop. Experimental results showed that the algorithms greatly improve the efficiency of Peer-to-Peer routing.

Key words: Routing, Peer-to-Peer network, Distributed systems, Internet
Document code: A **CLC number:** TP393

INTRODUCTION

Peer-to-Peer (P2P) techniques have attracted much attention, and have been used for all kinds of applications (Cheng, 2002; Rao *et al.*, 2002; Wu *et al.*, 2003). The core of these applications is P2P routing algorithm. The efficiency of the algorithm is of great importance to P2P systems' performance.

In existing P2P routing algorithms, each node maintains a routing table. Node of Chord (Stoica *et al.*, 2001), for example, maintains a routing table with at most 160 entries. The ith entry in the routing table at node n contains the identifier of the first node, s, that succeeds n by at least 2^{i-1} on the identifier circle, i.e., $s = successor(n + 2^{i-1})$, where $1 \leqslant i \leqslant 160$ (and all arithmetic is modulo 2^{160}). Since the routing table is setup based on identifiers, the routing path is optimal in identifier space, but it may not be optimal in geographic space. However, the true efficiency is measured by the end-to-end latency of the path. Routing algorithms that ignore the latencies of individual hops are likely to result in high latency paths. Thus, Topology-aware routing is more efficient for P2P overlay network than identifier based routing.

There are four kinds of techniques for coping with topology: proximity neighbor selection (Castro *et al.*, 2002), geographic layout (Ratnasamy *et al.*, 2002), super-network routing (Yang and Hector, 2003; Zhao *et al.*, 2002), and proximity routing (Ratnasamy *et al.*, 2002). These techniques improve routing performance in some degree, but have limitations. Proximity neighbor selection and geographic layout only apply to certain kinds of routing algorithms; the load and the space overhead of super-network routing are concentrated on super-nodes; proximity routing is of low efficiency. There are no achievements that can effectively improve the performance of the Chord algorithm.

For addressing the limitations of existing works, Wu *et al.* (2002) presented a topology-aware routing algorithm called TRA algorithm, which greatly improves P2P routing performance and may be applied to all kinds of P2P networks. The load and the space overhead of the TRA algorithm are much lower than that of the super-network routing algorithm. However, sending route query to super-node may lead to an increase in the total number of hops taken.

This paper proposes super-proximity routing

* Project (No. 025115032) supported by the Science & Technology Committee of Shanghai Municipality Grid Research Grant, China

algorithms which combine physical network and overlay network in choosing the next hop. In super-proximity routing algorithms, the P2P overlay network and the clustered physical network (Krishnamurthy *et al.*, 2001; Zhao *et al.*, 2002) are presented as directed graphs. Let $\boldsymbol{R}$ be the connection matrix of the P2P overlay network, and let $\boldsymbol{T}$ be the connection matrix of the clustered physical network. Connection matrixes of the super-proximity routing algorithms are the combinations of $\boldsymbol{R}$ and $\boldsymbol{T}$. The efficiency of routing based on the connection matrixes will be greatly improved.

SUPER-PROXIMITY ROUTING

This section presents the super-proximity routing algorithms to address the limitations of the identifier based routing algorithms. Before presenting the algorithms, we will first introduce the overlay connection matrix and the physical network connection matrix.

1. Overlay connection matrix

For an overlay network such as Chord (Stoica *et al.*, 2001), the connection matrix of the overlay graph (V, E) is given by:

$$\boldsymbol{R} = \{\boldsymbol{r}_1, \boldsymbol{r}_2, \cdots, \boldsymbol{r}_i, \cdots, \boldsymbol{r}_n\}^{\mathrm{T}} \quad i = 1, \cdots, n \tag{1}$$

Vector $\boldsymbol{r}_i$, which is maintained by v_i, $v_i \in V$, contains n entries.

$$\boldsymbol{r}_i = \{r_{i1}, r_{i2}, \cdots, r_{ij}, \cdots, r_{in}\} \quad i = 1, \cdots, n \tag{2}$$

Where

$$r_{ij} = \begin{cases} 1, & \text{if } (v_i, v_j) \in E \\ 0, & \text{otherwise} \end{cases}$$

2. Physical network connection matrix

In super-network routing (Yang and Hector, 2003; Zhao *et al.*, 2002), nodes that are topologically close and under common administrative control are clustered as a group. Super-node of the group refers to client nodes and the client nodes refer to the super-node. The clustered physical network (Krishnamurthy *et al.*, 2001; Zhao *et al.*, 2002) may be given by:

$$\boldsymbol{T} = \{\boldsymbol{t}_1, \boldsymbol{t}_2, \cdots, \boldsymbol{t}_i, \cdots, \boldsymbol{t}_n\}^{\mathrm{T}} \quad i = 1, \cdots, n \tag{3}$$

$\boldsymbol{t}_i$ in Eq.(3) is further given by:

$$\boldsymbol{t}_i = \{t_{i1}, t_{i2}, \cdots, t_{ij}, \cdots, t_{in}\} \quad i = 1, \cdots, n \tag{4}$$

Where

$$t_{ij} = \begin{cases} 1, & v_i \text{ refers to } v_j \\ 0, & \text{otherwise} \end{cases}$$

Matrix $\boldsymbol{T}^2$ represents full connection between the nodes of the same group, where each node refers to all other nodes in the same group. t^2_{ij} is the element in the ith row, jth column of matrix $\boldsymbol{T}^2$, the value of which is given by:

$$t^2_{ij} = \sum_{k=1}^{n} t_{ik} \cdot t_{kj} \tag{5}$$

The operation in Eq.(5) is Boolean operation.

In the rest of the paper, $id(v)$ denotes the identifier of node v. Let $\boldsymbol{P}_C = \boldsymbol{R}$, $\boldsymbol{P}_L = \boldsymbol{R} + \boldsymbol{T}^2$ and $\boldsymbol{P}_B = \boldsymbol{R} \times \boldsymbol{T}^2$. $\forall\, i, j,\ 1 \leqslant i, j \leqslant n$, $P_{C,ij}$, $P_{L,ij}$, and $P_{B,ij}$, denote the element in the ith row, jth column of $\boldsymbol{P}_C$, $\boldsymbol{P}_L$ and $\boldsymbol{P}_B$, respectively.

3. Routing algorithm

Super-proximity routing combines super-network routing and proximity routing. The combinations are achieved by the operations on $\boldsymbol{R}$ and $\boldsymbol{T}$, which play a large part on how efficient the resulting algorithms are. This section presents two combinations:

Load The connection matrix of the local algorithm is $\boldsymbol{P}_L$. Compared with the TRA algorithm (Wu *et al.*, 2002), the load is well balanced and there are no extra hops taken, but the space overhead of this approach is higher than that of the TRA algorithm.

Fig. 1 shows the pseudo-code that implements search process of the local algorithm. *find* (v_i, key) returns the node with the identifier closest to *key*. v_i is the starting routing node. v_j is the successor of v_i in Chord routing table. Line 1 decides if the destination is reached. If the destination is reached, line 2 returns the destination node. Otherwise, line 4 – 5 choose the next routing hop v_k which satisfies that $key \leqslant id(vk)$ and the identifier of v_k is the closest to *key*.

```
find(v_i, key){
1  if(id(v_i) < key and key ≤ id(v_j))
2    return v_j;
3  else
4    next hop is v_k, which satisfies:
5      P_{L,ik} ≡ 1, key ≤ id(v_k) and
       ∀ P_{L,il}, P_{L,il} ≡ 1, v_l ∉ [key, id(v_k)];
6    find(v_k, key);
}
```

Fig.1 The pseudo-code to find the node responeing for identifier *key*

Broaden The connection matrix of the broaden algorithm is $\boldsymbol{P}_B$. This algorithm is more efficient, since the number of neighboring nodes of each node is greater than that of the local algorithm. Each node in this algorithm refers to other nodes of the group, i.e. v, and the nodes in the successor list of node v. The search process of the broaden algorithm is similar to that of the local algorithm.

ANALYSIS

In this section, we present the theoretical analysis on the super-proximity routing algorithms. The experimental results presented in the next section will verify the analysis.

Theorem 1 $\forall v_i, v_j \in V$, let $Hop_C(i,j)$ and $Hop_L(i,j)$ be the numbers of hops traversed from v_i to v_j by Chord and the local algorithm, respectively. Then, $\forall v_i, v_j \in V$, $\exists k \in N$, if $Hop_C(i,j) < k$, then $Hop_L(i,j) < k$; $\exists v_i, v_j \in V$, if $Hop_L(i,j) = k+1$, then $Hop_C(i,j) > k+1$.

Proof $\because \boldsymbol{P}_L = \boldsymbol{R} + \boldsymbol{T}^2$, and $\boldsymbol{P}_C = \boldsymbol{R}$.

$\therefore \boldsymbol{P}_L - \boldsymbol{P}_C = \boldsymbol{T}^2$.

$\therefore \forall i,j,\ 1 \leqslant i,\ j \leqslant n,\ P_{C,ij} \equiv 1 \Rightarrow P_{L,ij} \equiv 1$,

$\exists i,\ j,\ 1 \leqslant i,\ j \leqslant n,\ P_{L,ij} \equiv 1 \Rightarrow P_{C,ij} \equiv 0$.

$\therefore \forall v_i, v_j \in V$, if $Hop_C(i,j) < k \Rightarrow Hop_L(i,j) < k$,

$\exists v_i, v_j \in V$, if $Hop_L(i,j) = k+1 \Rightarrow Hop_C(i,j) > k+1$.

Theorem 1 means that the performance of the local algorithm is higher than that of Chord. We can also prove that the performance of the broaden algorithm is higher than that of the local algorithm.

Definition 1 $\forall v_i, v_j \in V$, the identifier distance from v_i to v_j is given by:

$$d(i,j) = \begin{cases} id(v_j) - id(v_i), & id(v_j) \geqslant id(v_i) \\ id(v_j) - id(v_i) + 2^{160}, & id(v_j) \leqslant id(v_i) \end{cases}$$

Lemma 1 There are k node $\{nd_1, nd_2, \ldots, nd_k\}$ distributed in $[id_k, id_k + l)$, where $l \geqslant k \geqslant 0$. Each identifier in $[id_k, id_k + l)$ corresponds to one node. $L(nd_i, k, l)$ is the identifier corresponding to nd_i. $L_+(k,l) = \max(L(nd_1, k, l), L(nd_2, k, l), \ldots, L(nd_k, k, l))$. $EL_+(k,l)$ is the expected value of $L_+(k, l)$. Then:

$$(l + k - 2)/2 \leqslant EL = (k, l) < l \quad (6)$$

Proof Let $p_i = P(L_+(k,l) = id_k + i)$, then $p_{i+1} \geqslant p_i$, and $\sum_{i=k-1}^{l-1} p_i = 1$.

$\therefore (l+k)/2(p_{l-1} - p_{k-1}) + (l+k-2)/2 \times (p_{l-2} - p_k) + \ldots + (p_{(l-k-1)/2} - p_{(l-k-3)/2}) \geqslant 0$.

$\therefore EL_+(k,l) = \sum_{i=k-1}^{l-1} i \times p_i \geqslant (l+k-2)/2 \times (p_{k-1} + p_k + \ldots + p_{l-1}) = (l+k-2)/2$.

Also $\because EL_+(k,l) = \sum_{i=k-1}^{l-1} i \times p_i < \sum_{i=k-1}^{l-1} l \times p_i = l$,

then $(l+k-2)/2 \leqslant EL_+(k,l) < l$.

Lemma 2 v_i caches m nodes, which are evenly distributed in $[0, 2^{160})$. v_i lookups v_j which satisfies $id(v_i) + 2^k \leqslant id(v_j) < id(v_i) + 2^{k+1}$. The expected identifier distance traversed by Chord in one hop is $D_C = 2^k$, and the expected distance traversed by the super-proximity routing algorithms is given by:

$$D_P(k) = 2^k + \sum_{l=0}^{2^k-1} L_+(m,l) \times (1 - e^{-lm/2^{160}}).$$

The identifier distance improvement is given by:

$$D_{P\_}(k) = \sum_{l=0}^{2^k-1} L_+(m,l) \times (1 - e^{-lm/2^{160}}).$$

Proof Fig.2 shows the identifier distances traversed by Chord and the super-proximity routing algorithms in one hop. The message will route to $v_{i,k}$ and v_j in Chord and the super-proximity routing algorithms, respectively, then $D_C = 2^k$. Let X be the number of nodes distributed in $[id_k, id_{k+l})$, then $P(X \geqslant 1) = 1 - e^{-lm/2^{160}}$. Let $l = id(v_j) - id(v_{i,k})$, then $D_P(k) =$

$$\sum_{l=0}^{2^k-1}(2^k + EL_+(m,l)) \times P(X \geqslant 1) = 2^k + \sum_{l=0}^{2^k-1} EL_+(m,l) \times (1 - e^{l-m/2^{160}}).$$

Thus $D_{P_-}(k) = \sum_{l=0}^{2^k-1} L_+(m,l) \times (1 - e^{-lm/2^{160}})$.

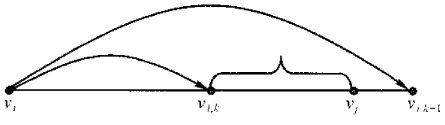

Fig. 2 The identifier distance traversed by Chord and the super-proximity routing algorithms in one hop. $id(v_{i,k}) \leqslant id(v_j) < id(v_{i,k+1})$, $id(v_{i,k}) = id(v_i + 2^k)$, **and** $id(v_{i,k+1}) = id(v_i + 2^{k+1})$

Theorem 2 $\forall v_i, v_j \in V$, $D_{P_-}(k)$ increases with $d(i, j)$.

Proof Since an increase in $d(i,j)$ will lead to an increase in the expected value of k. $\because$

$$D_{P_-}(k) = \sum_{l=0}^{2^k-1} EL_+(m,l) \times (1 - e^{-lm/2^{160}}).$$

$\therefore k_2 > k_1 \Rightarrow D_{P_-}(k_2) > D_{P_-}(k_1)$.

$\therefore D_{P_-}(k)$ increases with $d(i,j)$.

Theorem 2 shows that the larger $d(i,j)$ is, the greater the routing performance improves.

EXPERIMENTAL RESULTS

In this section, we evaluate the algorithms by experiments. The results were obtained using GT-ITM Models (Zegura *et al.*, 1996). We compare the logical hops, the distance traversed, and the load of the algorithms.

We use "transit-stub" model to obtain topologies that is more closely resemble the Internet hierarchy than pure random graph. Unless otherwise specified, a topology of 28800 nodes with a cluster size of 400 is used for the experiments. The figures are obtained by 288000 lookups with random selected keys from random nodes under GT-ITM Internetwork.

1. Logical hops

Fig.3 shows the probability density function (PDF) of the number of logical hops per message. The numbers of logical hops with the highest PDF for Chord, the TRA algorithm, the local algorithm, and the broaden algorithm are approximately 8, 7, 4, and 2, respectively. The number of the local algorithm is much smaller than that of Chord, while the number of the TRA algorithm is close to that of Chord. The reason for this variation is that, in the TRA algorithm, every lookup message routes to the super-node first, which leads to an increase in the total number of logical hops taken. Moreover, the figure shows that the PDF of odd hops is larger than that of even hops. This is because, when the number of hops is even, the lookup message will very probably be sent to the super-node, which will redirect the lookup message to the next node. But in the local algorithm and the broaden algorithm, the client nodes need not route messages to the super-node first. The experimental results agreed with Theorem 1.

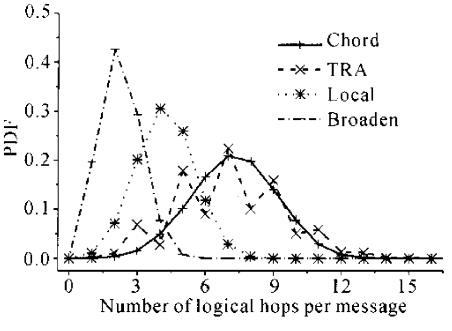

Fig.3 The PDF of number of logical hops per message

Fig.4 shows the identifier distance versus the number of routing hops. Each dot corresponds to one message lookup. The real lines are fitted

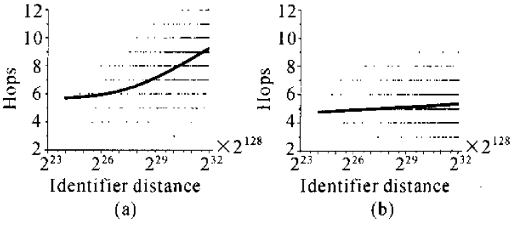

Fig.4 The identifier distance versus the number of routing hops(a) Chord; (b) Local

with an expression in the form of $y = a - b \times$

$\ln(x+c)$, where x is $d(i, j)$ given in definition 1, and y is the number of logical hops. The figure shows that the larger the identifier distance is, the greater the number of logical hops improves. The experimental results in Fig.4 agreed well with Theorem 2.

2. Distance traversed

Fig.5 plots the PDF of distribution of distance traversed per message. The distances traversed with the highest PDF by Chord, the TRA algorithm, the local algorithm, and the broaden algorithm are approximately 200, 100, 95, and 75, respectively. Even though the number of logical hops of the TRA algorithm is close to Chord's (refer to Fig.3), the distance traversed by the TRA algorithm is much smaller than that of Chord. Moreover, the distance traversed by the local algorithm is slightly smaller than that of the TRA algorithm, though the number of logical hops of the local algorithm is much smaller than that of the TRA algorithm (refer to Fig.3). The reason for these variations is that, in the TRA algorithm, there are extra hops from the client nodes to the super-nodes, the distance of which is small, but there are no such extra hops in the local algorithm and the broaden algorithm.

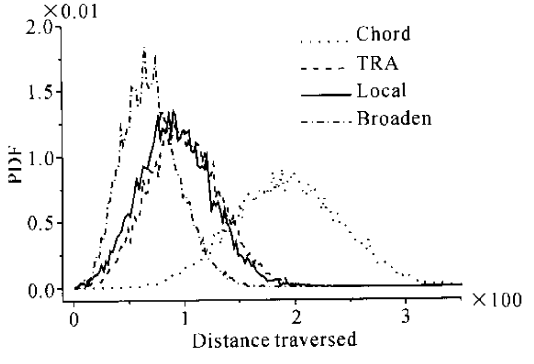

Fig.5 The PDF of distance traversed per message

3. Load evaluation

The load of nodes and the load balancing are important parameters for evaluating P2P systems. This paper uses the number of messages of each node routes as parameters for evaluation of the load. Fig.6 shows that the load of the super-proximity routing algorithms is lighter than that of Chord. To get a good view of some heavily loaded nodes in the TRA algorithm, we break the horizontal coordination and the vertical coordination. The figure shows that some nodes of the TRA algorithm are heavily loaded, while there are no such nodes in the super-proximity routing algorithms, which means that there are no load concentrations in the super-proximity routing algorithms.

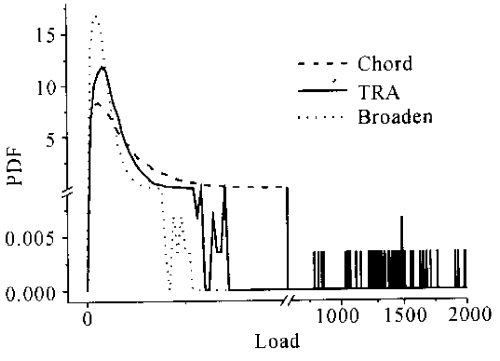

Fig.6 The PDF of load

Fig.7 shows the average load and the load balance value of the algorithms. The smaller the load balance value, the better the load balances. The loads of broaden algorithm and the local algorithms are much lower than that of Chord. We conclude that the super-proximity routing algorithms are effective in reducing the load of the node of Chord.

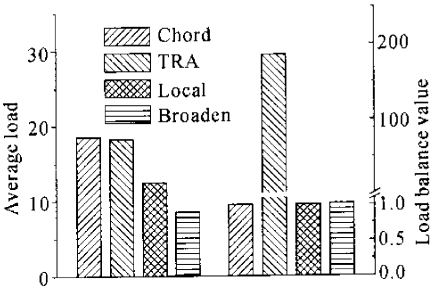

Fig.7 The average load and the load balance value

CONCLUSIONS

P2P applications have become a popular medium for sharing huge amounts of data. One critical part of the P2P applications is P2P routing algorithms. Structured routing algorithms guarantee that any key may be found in $O(\log N)$

steps. However, the algorithms based on identifier have low efficiency. This paper presents super-proximity routing algorithms called local algorithm and broaden algorithm, which combine overlay network and physical network to choose the next hop.

Experimental results showed that, for a 28800-node GT-ITM Internet topology with a cluster size of 400, the super-proximity routing algorithms are able to improve the routing performance by more than twice, while the load of the algorithms is much lighter than that of Chord. Even though the study is based on Chord overlay network, the results of this paper may be applied to other structured overlay networks. In summary, the algorithms presented in this paper proved to be more efficient than others.

References

Castro, M., Druschel, P., Charlie, Y. and Rowstron, A., 2002. Exploiting Network Proximity in Distributed Hash Tables. Proc. of the International Workshop on Future Directions in Distributed Computing. Bertinoro, Italy, p. 60 – 63.

Cheng, S., 2002. Optional and responsive fine-grain locking in Internet-based collaborative systems. *IEEE Transactions on Parallel and Distributed Systems*, **13**(9): 994 – 1008.

Krishnamurthy, B., Wang, J. and Xie, Y.L., 2001. Early Measurements of A Cluster-based Architecture for P2P Systems. Proc. of the 1st ACM SIGCOMM Internet Measurement Workshop. San Francisco, USA, p. 105 – 109.

Rao, W.X., Wu, Z.D. and Ma, F.Y., 2002. Mtree-Cast: A New Application Level Mutlicast. Proc. of International Workshop on Grid and Cooperative Computing. Sanya, China, p. 45 – 56.

Ratnasamy, S., Shenker, S. and Stoica, I., 2002. Routing Algorithms for DHTs: Some Open Questions. Proc. of International Workshop on Peer-to-Peer Systems. Cambridge, MA, USA, p. 45 – 52.

Stoica, I., Morris, R., Karger, D. and Kaashoek, M., 2001. Chord: a scalable peer-to-peer lookup service for Internet applications. *Computer Communication Review*, **31**(4): 149 – 160.

Wu, Z.D., Rao, W.X. and Ma, F.Y., 2002. Efficient Topology-Aware Routing in Peer-to-Peer Network. Proc. of International Workshop on Grid and Cooperative computing. Sanya, China, p. 172 – 185.

Wu, Z.D., Rao, W.X. and Ma, F.Y, 2003. DEBIZ: A Decentralized Lookup Service for E-commerce, Accepted by Proc. of APWeb'2003, Lecture Notes on Computer, Springer-Verlag, Xi'an, China.

Yang, B. and Hector G.M., 2003. Designing A Super-Peer Network. Proc. of 19th International Conference on Data Engineering. IEEE, Bangalore, India.

Zhao, B., Duan, Y., Huang, L., Joseph, A. and Kubiatowicz, J., 2002. Brocade: Landmark Routing on Overlay Networks. Electronic Proc. of the International Workshop on Peer-to-Peer Systems. Cambridge, MA, USA, p. 34 – 44.

Zegura, E., Calvert, K. and Bhattacharjee, S, 1996. How to Model An Internetwork. Proc. of IEEE Infocom '96. San Francisco, CA, p. 594 – 602.