View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by DSpace@MIT

Relaxing Routing Table to Alleviate Dynamism in P2P Systems

Hui FANG[1], **Wen Jing HSU**[2], and **Larry RUDOLPH**[3]
[1] Singapore-MIT Alliance, National University of Singapore
[2]Nanyang Technological University, Singapore
[3]CSAIL, MIT, USA

***Abstract*— In dynamic P2P networks, nodes join and depart from the system frequently, which partially damages the predefined P2P structure, and impairs the system performance such as basic lookup functionality. Therefore stabilization process has to be done to restore the logical topology. This paper presents an approach to relax the requirement on routing tables to provide provably better stability than fixed structured P2P systems. We propose a relaxed Chord that keeps the $O(\log N)$ number of hops for greedy lookup, but it requires less stabilization overhead. It allows a tradeoff between lookup efficiency and structure flexibility without adding any overhead to the system. In the relaxed routing structure, each routing entry("finger") of the node is allowed to vary within a set of values. Each node only needs to keep a certain number of fingers that point to nodes in its anchor set. This relaxation reduces the burden of state management of the node. The relaxed routing scheme provides an alternative structure other than randomized P2P and deterministic P2P, by relaxing on finger selection. It provides good flexibility and therefore extends the system functioning time.**

***Index Terms*— P2P, Chord, greedy routing, relaxed DHT, anchor set**

Manuscript received November 7, 2005. This work was supported by the program of Singapore-MIT Alliance.

H. Fang is with the Singapore-MIT Alliance, National University of Singapore (e-mail: fanghui@nus.edu.sg).

W.J. Hsu is with the Singapore-MIT Alliance, Nanyang Technological University, Singapore (e-mail: hsu@theory.csail.mit.edu).

L. Rudolph is with the Computer Science and Artificial Intelligence Laboratory of Massachusetts Institute of Technology (e-mail: rudolph@csail.mit.edu).

I. Introduction

A P2P network is a distributed system in which peers employ distributed resources to perform a function in a decentralized fashion. In the past few years, many structured peer-to-peer (P2P) systems have been proposed, e.g. Chord [1], Koorde [4], Pastry [5] and Symphony [6]. Most of them utilize the Distributed Hashing Table(DHT) technique to realize better scalability and routing efficiency. In terms of topology formation, they can be classified into two categories: deterministic and randomized networks. Examples of the deterministic networks include Chord, CAN [3] and Pastry. Examples of the randomized networks include Symphony and randomized Chord [13]. The P2P network infrastructure is potentially symmetric for each node and there is no center server. Greedy routing can achieve efficient searching in many structured P2P systems [2], [12].

The peers may join or leave the P2P system frequently; hence, P2P networks are dynamic in nature. The dynamics of P2P system could impair the network structure and reduce system performance in terms of, e.g. routing efficiency and fault resilience [9], [10]. The metrics of routing efficiency include for instance, query path length, routing table size and message/time complexity [11], [12] etc. The resilience of a P2P system lies in its ability to maintain its performance in the presence of dynamics.

Many P2P routing protocols- like CAN, Chord and Pastry- rely on a rationalized structure that enables efficient searching. A typical approach to the design of such networks is roughly as follows [9]. First, an

"ideal" structure is specified, under which routing is efficient. Then, consider a protocol that allows a node to join or leave the network, then properly recover the network connectivity in view of the change. One then shows that the protocol for the ideal network can still keep routing efficiently even after the failure of some portion of the nodes. For example, Chord uses stabilization protocol [1] to regulate the dynamically formed topology. Considering the fact that a P2P network is a continuously evolving system, we approach the dynamism alternatively, that is, to design a relatively relaxed network structure which itself can tolerate network dynamism without having to design complicated routing protocols, while still keeping the routing sufficiently efficient.

Recently, some results have been done on the randomization of P2P network such as "small-world network" and randomized Chord [8]. The study of small-world network started with Milgram's study on social networks which shows that any two people in the world are connected via a chain of six acquaintances on average. Kleinberg modeled the social networks with a family of random graphs with long links. In particular, Kleinberg considered a 2D n x n grid with n^2 nodes. Each node is with a small set of local contacts and one long-range contact defined by a harmonic distribution. With greedy routing, the path-length between any pair of nodes is $O(\log^2 N)$ hops w.h.p. Symphony adapts Kleinberg's construction to arrive at a randomized P2P routing network. It uses a ring (one dimension) and each node with multiple long links (instead of one). The average length of greedy routes has been shown to be $O(\log^2 N/k)$, where k denotes the number of links per node. Other models of randomized P2P include: randomized-hypercube, randomized-Chord, and skip-graphs [13], [14]. Randomized-Chord is a variation on a deterministic Chord where each node x creates its i-th finger connecting to one node within $[x + 2^i, x + 2^{i+1})$. The routing performance for deterministic network is generally better than the randomized networks.

With regard to the challenges posed earlier, we suggest to relax on the structure of the deterministic P2P networks such that the fault resilience can be improved while not sacrificing much on the routing efficiency. We have applied this approach to enhance Chord and obtained encouraging results. We believe that this approach can be applied to other deterministic P2P networks. In Section 2, we will present our approach. Section 3 draws a conclusion and suggests future research.

II. Model and Approach

A. Uniform greedy routing on P2P

The DHT scheme uses hashing to assign each node an identifier in the name space, and to distribute the burden of maintaining the routing table. Each node stores objects that lie in a certain portion of the key space, and forwards the requests for the other keys via the other suitable nodes.

Assume that each node x keeps $k = k(x)$ pointers to other nodes. Denote them as $f = \{f_1, f_2, ..., f_k\}$, where $f_i \triangleq$distance between x and the i-th pointer. Without loss of generality, f is in strictly ascending order, i.e. $f_1 < f_2 ... < f_k$.

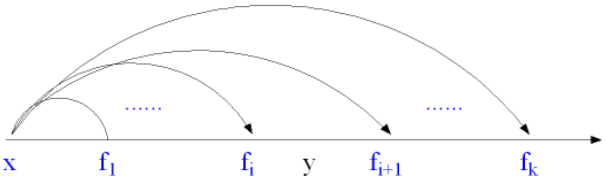

Fig. 1. Each node keeps k pointers, where x denotes the source node and y denotes the target

The *uniform greedy routing* scheme [11] in P2P systems is like this: When a request destined for key y reaches node x, x will forward it to the node $x + f_i$ where $f_i \leq y - x \leq f_{i+1}$. The algorithm is uniform in that each node takes action according to the same principal. Chord uses similar greedy routing protocol.

B. Relaxed routing table

Theorem 1: Let c and β denote two constants and $\beta > 1, c \geq 1$. In a ring-based DHT network, each node keeps $k > 1$ pointers $\{f_1, f_2, ..., f_k\}$. If for any node, it satisfies two conditions:
(1) Spacing: $f_i < f_{i+1} \leq \beta f_i$;
(2) Boundary: $f_1 = 1, f_k = \frac{N}{c}$.
Then uniform greedy lookup can be achieved in $c + \log_d N$ time, where $d = \frac{\beta}{\beta - 1}$.

Proof:

The distance between the target and the querying node is shortened by the ratio: $\frac{y-x-f_i}{y-x} = 1 - \frac{f_i}{y-x} < 1 - \frac{f_i}{f_{i+1}}$.

By the spacing condition, the decreasing ratio is always lower than $\frac{1}{d}, d > 1$. Therefore the query to target key y can be achieved (within offset f_1) in at most $\log_d(y-x) \leq \log_d N$ steps.

Remarks: f_1 is the minimum jump from the querying node, which implies how near to the target the lookup can achieve by recursively shortening the offset. Setting $f_1 = 1$ makes it necessary to execute exact lookup. f_k is the maximum jump from the querying node. By the boundary condition, a constant times of f_k can cover the whole ID space ($N = c * f_k$). Otherwise the number of forwarding hops may be $\frac{N}{f_k} + \log_d f_k >> \log_d N$ when $N >> f_k$.

Notice that $f_k \leq \beta f_{k-1} \leq ... \leq \beta^{k-1} f_1$ and if $f_1 = 1, f_k = N/c$, then $k \geq \log_\beta \frac{N}{c} + 1$. That implies each node must keep at least $\log_\beta \frac{N}{c} + 1$ pointers to satisfy the above two conditions.

As we can see, when the parameter β becomes larger, d will become smaller and uniform greedy routing needs more hops. When β becomes smaller, d will become greater and hence less hops. This means the dense spacing between pointers helps to improve lookup efficiency.

In implementation, β can be a predetermined parameter to control the tradeoff between pointer spacing and lookup efficiency. The routing table is thus set up with parameter β, without requirement that each pointer must be the successor of certain exact distance. That is, Condition (1) and Condition (2) ensure the correctness and $O(\log N)$ efficiency of lookup. Because each pointer f_{i+1} can be any random number between $[f_i, \beta f_i)$, we call the routing table satisfying (1) and (2) a $\beta - relaxed$ routing table, to distinguish with the other routing schemes such as Chord and De Brujin network.

```
procedure x.updateRoutingTable
  input: global parameter β
  for i=1 to log_β N + 1 do
    if f_i failed or f_i > β f_{i-1} do
        f_i = lookup(f_{i-1} + 1);
    endif
  endfor
end
```

TABLE I
THE SETUP/UPDATE OF FLOATING ROUTING TABLE

Corollary 1: Denote $x.f_i$ to be the distance between node x and its i-th pointer, $1 \leq i \leq k$. Let $\beta = \max_{(x,i)}\{\frac{x.f_{i+1}}{x.f_i}\}$, $f_{min} = \max_x\{x.f_1\}$, and $f_{max} = \max_x\{x.f_k\}$. If $f_{max} = \Theta(N)$, then the uniform greedy lookup can be achieved within distance f_{min} in $c + \log_{\frac{\beta}{\beta-1}} N$ steps, where c is a constant.

Proof: (Directly from Theorem 1.)

In terms of Theorem 1, we write the algorithm to set up the floating routing table for uniform greedy searching in P2P systems (Seen in Table 1).

C. Anchor set

One disadvantage of the above method is that one failed $x.f_i$ may trigger the updates of the following pointers if the invariance $f_{i+1} \leq \beta f_i$ does not hold. To avoid this, the algorithm is designed to anchor each f_i at first and then randomly choose a node around its anchor.

Theorem 2: Given a partition $\{\hat{f}_i\}$ on ID space that satisfies condition (1) and (2) in Theorem 1,

where $\beta = \max_i\{\frac{\hat{f}_{i+1}}{\hat{f}_i}\}$, and constant $0 < \alpha < \min_i\{\frac{\hat{f}_{i+1}}{\hat{f}_i}\} - 1$.

(a) If $\forall x, i : x.f_i \in [\hat{f}_i, (1+\alpha)\hat{f}_i)$, then the lookup can be achieved in $\log_{\frac{\beta'}{\beta'-1}} N$ steps, where $\beta' = (1+\alpha)\beta$.

(b) If $\forall x, i : x.f_i \in [\hat{f}_i, \hat{f}_{i+1})$, then the lookup can be achieved in $\log_{\frac{\beta^2}{\beta^2-1}} N$ steps, which is greater than the number of steps in (a).

Proof:

For (a), if $f_i \in [\hat{f}_i, (1+\alpha)\hat{f}_i)$, then
$f_{i+1} \geq \hat{f}_{i+1} \geq (1+\alpha)\hat{f}_i > f_i$
and
$f_{i+1} < (1+\alpha)\hat{f}_{i+1} \leq (1+\alpha)\beta\hat{f}_i \leq (1+\alpha)\beta f_i$.

It shows that the set $\{f_i\}$ also satisfies condition (1) and (2), with parameter β'. By applying Theorem 1, we got result (a).

For (b), if $f_i \in [\hat{f}_i, \hat{f}_{i+1})$, assume that node x is the source and y the target lying in $[\hat{f}_i, \hat{f}_{i+1})$. If $f_i \leq y$, then x will forward the query to f_i. Otherwise x will forward query to f_{i-1}. In any case the distance between the target and querying node is shortened by ratio at least: $\frac{y-x-f_{i-1}}{y-x} = 1 - \frac{f_{i-1}}{y-x} < 1 - \frac{\hat{f}_{i-1}}{\hat{f}_{i+1}} = 1 - \frac{\hat{f}_{i-1}}{\hat{f}_i}\frac{\hat{f}_i}{\hat{f}_{i+1}} < 1 - \frac{1}{\beta^2}$. Therefore the total number of steps is no more than $\log_{\frac{\beta^2}{\beta^2-1}} N$.

Since that $\beta' = (1+\alpha)\beta < \beta^2$, the number of steps for (b) is greater than (a). This completes the proof.

$\{\hat{f}_i\}$ is called the anchor set in the paper. The anchor set provides a good way to position each routing pointer around the specified place(without floating too far away), while still keeping the $O(\log N)$ lookup. The parameter α in algorithm can control the floating distance from the anchor point. As seen from Theorem 2, the relatively small β is related to efficient lookup. However, it could constrain the value of the parameter for floating area, α. So there's a trade-off between flexibility and routing efficiency. When the anchor set and global parameter β, α have been decided, the pointers then are confined in an area around the anchors.

```
procedure x.updateRoutingTable2
  input: Global anchor set {f̂_i} with parameter β,
  for i=1 to log_{β_max} N + 1 do
    if f_i failed do
        f_i = lookup(f̂_i);
    endif
  endfor
end
```

TABLE II
THE SETUP/UPDATE OF FLOATING ROUTING TABLE WITH ANCHOR SET

The result of relaxed routing table can be applied to the Chord protocols. For example, $\{1, 2, 2^2, ..., 2^{[\log N]}\}$ can be chosen as an anchor set with $\beta = 2$. If the routing fingers are relaxed to only satisfy: $f_i \in [2^{i-1}, 2^{i-1}(1+\alpha))$, where $\alpha = 1/4$, then the lookup can be achieved in about $1.36\log_2 N$ hops. The relaxed routing table with anchor set provides flexibility on node connection for dynamic networks.

With the anchor set, the relaxed routing table can be set up as in Table 2.

Definition 1: A relaxed DHT is a DHT with anchor set $\{\hat{f}_i\}, 1 \leq i \leq k$, where each node keeps k pointers $\{f_i\}$ and satisfies: $f_i \in [\hat{f}_i, \hat{f}_{i+1})$.

Remarks: The parameters α, β in Theorem 2 can be introduced in the relaxed routing P2P to further define the floating area of the nodes in space.

Definition 2: Given a relaxed DHT with anchor set $\{\hat{f}_i\}$. For node x, its $i-th$ pointer is f_i. The x's $relax\ factor, r$ is defined as:

$$r = \frac{1}{k}\sum_{i=1}^{k}(f_i/\hat{f}_i - 1)$$

Theorem 3: Given a relaxed routing P2P with anchor set $\{\hat{f}_i\}$. For node x: (1) Any number of

concurrent node join will not increase x's relax factor; (2) Node x will tolerate at least $k*r$ number of node failure (other than itself) to keep its relax factor non-increased.

Proof: When new node joins, node x will have more choices to select f_i to be the closer to $\hat{f}_i$. So it will not increase the value of relax factor. Instead, the value could possibly be decreased.

In terms of algorithm in Table 2, if there is any live node between $\hat{f}_i$ and f_i, the relax factor can not be increased. Therefore the i-th finger of node x can tolerate $f_i - \hat{f}_i$ node failures. Summing up all values with regard to i, node x can tolerate at least S number of node failures, where

$$S = \sum_{i=1}^{k} (\hat{f}_i - f_i) > k * r, \text{ since } \hat{f}_i \geq 1.$$

Remarks: Normally f_i is chosen from x's routing table as the closest candidate whose pointer is greater than $\hat{f}_i$. In general, the maximum relax factor reflects the ability of the system to tolerate structural dynamism due to node joining or leaving. In this sense, no stabilization work is needed ever when large number of nodes join as long as the relax factor does not increase. However, the bigger the relax factor is, the lower the performance of lookup will be.

The relaxed routing structure can be applied to revise most structured P2P systems. The relaxed DHT bridges, in a sense, between unstructured P2P and structured P2P networks.

III. Conclusions

For dynamic P2P networks, we propose a relaxed Chord that keeps the $O(\log N)$ number of hops for greedy lookup, allowing a tradeoff between lookup efficiency and structural flexibility.

Our work is inspired by the study on structured and unstructured P2P network. Structured P2P network such as Chord owns better routing efficiency than the unstructured ones. However, in practice the routing function is often impaired by continuously node joins and departures. The relaxed routing scheme introduces the randomization and flexibility on pointer selection, therefore extending the system operation.

Further study will be done on the choice of anchor set, and the effect of sampling the neighbor pointers around the specific anchor set.

References

[1] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, and H. Balarikishnan, *Chord: A scalable Peer-to-peer Lookup Service for Internet Applications*, ACM SIGCOMM, San Deigo, USA, pp. 149-160, August 2001

[2] M. Naor and U. Wieder, *Novel Architectures for P2P Applications: the Continuous-Discrete Approach*, in Proc. of the 15th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA 2003), San Diego, USA, pp. 50-59, June 2003.

[3] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, *A Scalable Content-Addressable Network*, ACM SIGCOMM,2001.

[4] F. Kaashoek and D Karger, *Koorde: A Simple Degree-optimal Distributed Hash Table*, In 2nd International Workshop on Peer to Peer Systems, LNCS Hot Topics, Berkeley, CA, January 2003. Springer.

[5] A. Rowstron and P. Druschel, *Pastry: Scalable, Decentralized Object Location and Routing for Large-Scale Peer-to-peer Systems*, IFIP/ACM International Conference on Distributed Systems Platforms, Heidelberg, Germany, pages 329-350, 2001.

[6] G.S. Manku, M. Bawa, and P. Raghavan, *Symphony: Distributed hashing in a small world*, In Proc. 4th USENIX Symposium on Internet Technologies and Systems (USITS 2003), pages 127-140, 2003.

[7] G. Manku, *Routing Networks for Distributed Hash Tables*, PODC'03, Boston, pp. 133-142, July 2003.

[8] J. Kleinberg, *The small-world phenomeon: An algorithmic perspective*, in Proc. 32nd ACM Symposium on Theory of Computing (STOC 2000), pp. 163-170, 2000.

[9] D. Liben-Nowell, H. Balakrishnan and D. Karger, *Analysis of the Evolution of Peer-to-peer Systems*, PODC'02, pp. 233-242, July 2002.

[10] D. Xuan, S. Chellappan and X. Wang, *Resilience of Structured P2P Systems: Analysis and Enhancement*, In Handbook on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless and Peer-to-Peer Networks, CRC press LLC, 2004.

[11] J. Xu, A. Kumar, and X. Yu, *On the Fundamental Tradeoffs between Routing Table Size and Network Diameter in Peer-to-peer Networks*, IEEE Journal on Selected Areas in Communications, Vol. 22, NO.1, pp. 151-163, January 2004.

[12] D. Loguinov, A. Kumar, V. Rai, and S. Ganesh, *Graph-Theoretic Analysis of Structured P2P Systems: Routing Distances and Fault Resilience*, In Proc. of 2003 conference on Applications, technologies, architectures, and protocols for computer communications, 1-58113-735-4, 395–406, ACM Press, Karlsruhe, Germany, 2003.

[13] H. Zhang, A. Goel, and R. Govindan, *Incrementally improving lookup latency in Distributed Hash Table Systems*, In ACM SIGMETRICS 2003, pages 114-125, June 2003.

[14] N. Harvey, M. Jones, S. Saroiu, M. Theimer, A. Wolman, *SkipNet: A Scalable Overlay Network with Practical Locality Properties*, In Fourth USENIX Symposium on Internet Technologies and Systems (USITS), 2003.