F-Chord: Improved Uniform Routing on Chord

G. Cordasco, L. Gargano, A. Negro, and V. Scarano
Dipartimento di Informatica ed Applicazioni, Università di Salerno, 84084, Fisciano (SA), Italy

M. Hammar
Research and Development, Apptus Technologies AB, IDEON, 22370 Lund, Sweden

We propose a family of novel Chord-based P2P schemes retaining all positive aspects that made Chord a popular topology for routing in P2P networks. The schemes, based on the Fibonacci number system, allow to simultaneously improve on the maximum/average number of hops for lookups and the routing table size per node.
 NETWORKS, Vol. 52(4), 325–332 2008

Keywords: peer-to-peer; overlay network; uniform routing; greedy routing; fibonacci numbers

1. INTRODUCTION

Peer-to-peer (P2P) is a class of network applications in which each workstation has equivalent capabilities and responsibilities and communications are potentially symmetric. P2P computing takes advantage of existing computing power, computer storage, and networking connectivity, available at the edges of the Internet, by allowing users to leverage their collective power to the benefit of all. Because accessing these decentralized resources means operating in an environment of unstable connectivity and unpredictable IP addresses, P2P nodes must operate independently from central servers.

Many are the recent P2P applications that are available. Among the most popular, without any doubt, are the file sharing systems. Anyway, sharing other resources is also common (like file storage systems and systems that use CPU time on each node).

Scalability has been recognized as the central challenge in designing such systems [14]. To obtain a scalable system, several P2P systems are based on distributed hash table (DHT) schemes [3, 15, 16, 18]. In DHT schemes, data as well as nodes are associated with a key and each node in the system is responsible for storing a certain range of keys. Each node stores data that correspond to a certain portion of the key space, and uses a routing scheme to forward the request for data whose key does not belong to its key space to the appropriate next-hop node.

In this article, we propose a family of new routing schemes that reduce the routing table size, and the maximum/average number of hops for lookup requests in Chord-like systems [16] without introducing any other protocol overhead. The improvement is obtained with no harm to the simplicity and ease of programming that are some of the many good characteristics that made Chord a popular choice.

The basis of Chord can be seen as a ring of N identifiers labelled from 0 to $N-1$. The edges, representing the overlay network, go from identifier x to identifier[1] $x+2^i$, for each $x \in \{0,\ldots,N-1\}$ and $i < \log N$. The degree and the diameter are $\log N$, the average path length is $(\log N)/2$. Routing is greedy, never overshooting the destination.

Because of low diameter and average path length, Chord offers fast lookup algorithms. Moreover, low average path length also offers a good tolerance to malfunctioning, since short paths (on average) do not often incur in faulty nodes. By having also low degree, it provides efficient join/leave of nodes since the cost depends on the diameter and the degree and is, in fact, upper bounded by their product. Chord is scalable: with $n \leq N$ nodes present in the network the same performance (in terms of n rather than N) can be obtained w.h.p. Efficient routing in Chord is easy (a greedy algorithm is optimal) due to the fact that Chord is *uniform*: x is connected to y iff $x+z$ is connected to $y+z$. Since we are restricting ourselves to uniform routing schemes, we can use the term *jump of size s* to indicate the existence in the overlay network of an edge from x to $x+s$ for any identifier $x = 0,\ldots,N-1$ (e.g. Chord has jumps of size 2^i, for $i = 0,\ldots,\log N-1$).

Uniformity is a crucial requirement, since it makes any system a good candidate for real implementations: besides simplicity in the implementation it also offers an optimal

Received November 2006; accepted February 2008
Correspondence to: G. Cordasco; e-mail: cordasco@dia.unisa.it
DOI 10.1002/net.20253
Published online 3 June 2008 in Wiley InterScience (www.interscience.wiley.com).

[1] Throughout the article, arithmetics on node identities is always mod N where N is the number of identifiers. Similarly, all the logarithms are base 2, unless differently specified.

greedy routing algorithm. Greedy routing is very simple to implement and has some "implicit" fault tolerance capability. Routing occurs between the portion of key space delimited by source and destination (locality in the key space) and, therefore, any possible semantics present in the keys is not lost (e.g. proximity between keys implies proximity between peers)[10]. Besides, in a greedy routing algorithm, at each step, the current node x selects, among all its neighbors, the node y that is the closest to the destination t and forwards to y; hence, if during the routing a faulty edge (peer) is found, all the work done is not lost (i.e. each executed step, independently, gets the message closer to the destination). Furthermore a greedy routing on a uniform network is peer-congestion-free [17] and ensures a quick bootstrap to any peer entering the network: In uniform networks, quick bootstrap of a peer x can be easily obtained by using the routing table of x's predecessor (that is the peer that precedes x on the ring) as a starting point to build an updated routing table.

On the other hand, it is known that, as long as uniformity is required, the $O(\log N)$ values for the number of jumps and diameter cannot be asymptotically improved [17]. Chord's values of the three main parameters (degree vs. diameter and average path length) can be improved only if one removes the uniformity request [6, 9–12].

Because of its practicality and its known bounds, it assumes a certain relevance to improve the performance of Chord while retaining its simplicity and scalability. Moreover, due to the above, it is interesting and practically useful to improve known results for uniform systems, even only by constant factors. Our objective is to show improved bounds on all the important parameters of a P2P system that affect lookup time and join/leave cost, i.e., degree, diameter, and average path length. To this aim we propose and analyze a novel family of uniform routing schemes.

The lookup process in Chord can be seen as a binary search on an interval of N elements. A natural question to ask is whether the lookup can be realized with a more efficient search technique that can be translated into a uniform overlay network. Efficiency is measured in terms of degree and maximum/average path length. We notice that the problem poses several restrictions on the search model since we are assuming that all nodes are alike and, therefore, only queries taken from a globally given set can be used at any step.

In this article, we consider search techniques in the above model imposed by uniform routing algorithms (that is, when a fixed set of jumps is available). To the best of our knowledge, while the problem is related to several search problems investigated in the literature, no useful results on problems totally fitting the above model and goals are known (see [5] and references therein quoted). A previous work in the P2P context in this direction is contained in [17]. Our starting point is Fibonacci search [4].

We present a family of routing schemes F-Chord(α) that improves (for a range of α) Chord's degree (routing table size), diameter and average path length.

Then we prove that F-Chord(1/2) is optimal with respect to degree and diameter.

1.1. Related work

The Chord system was introduced in [16] to allow efficient lookup in a distributed hash table (DHT). By using logarithmic size routing tables in each node, Chord allows to find in logarithmic N number of routing hops the node of a P2P system that is responsible for a given key. Adding or removing a node is accomplished at a cost of $O(\log^2 N)$ messages.

A thorough study of uniform systems is given in [17]. The authors present a scheme that obtains an improvement over the path length and degree at the expenses of the average path length. In fact, they propose a uniform routing scheme that is able to achieve a diameter of $0.768 \log_2 N$ when the routing table size is $0.768 \log_2 N$ (i.e. it maintains 21.4% fewer neighbors than Chord and achieves a diameter 21.4% smaller). Unfortunatly, their uniform routing scheme has average path length $0,6135 \log_2 N$, that is 22.7% greater than Chord's.

In [7] Ganesan et al. propose a scheme, which achieves both lower diameter and average path length than the standard Chord protocol, this is achieved by using each edge bidirectionally, that is, using a double degree and, accordingly, augmenting the cost of maintaining the nodes' routing table.

In general, Chord can be improved at the expenses of uniformity [9, 11, 12]. Recently, some nongreedy routing algorithms were proposed, that use De-Bruijn based DHT [6, 13], whose goal is to reach an optimal trade-off between degree and average path length and, in particular, allow routing in $\Omega(\log N / \log \log N)$ with logarithmic degree.

One can also improve Chord's results by eliminating the deterministic requirement. In fact, it is possible (see [1, 10]) to route greedily in $\Theta(\log N / \log \log N)$ with logarithmic degree by using randomization and the so called *neighbor-of-neighbor (NoN) approach*: a node uses, at each step, its neighbor's neighbors to make greedy decisions. These techniques can be adapted to our scheme thus obtaining similar results [2].

However, in this article we focus on *deterministic* and *uniform* routing schemes. Among other advantages, they offer an optimal greedy routing strategy that provides simplicity, fault tolerance (as long as some node has edges toward destination, the routing succeeds) and locality (messages flow only on the portion of ring between source and destination), as noted in [10].

2. A DETAILED ROADMAP OF THE PAPER

We provide here an introduction to the results described in the article with references to the sections of the article. We remind the reader that our goal is to improve on Chord's efficiency, namely diameter, degree, and average path length without giving up uniformity.

First, we, briefly, recall here some basic facts on Fibonacci numbers which will be used in the sequel (see [8]). Let $\mathrm{Fib}(i)$ denote the i-th Fibonacci number. They are defined as $\mathrm{Fib}(0) = 0, \mathrm{Fib}(1) = 1$ and, for each $i > 1$,

$$\mathrm{Fib}(i) = \mathrm{Fib}(i-1) + \mathrm{Fib}(i-2).$$

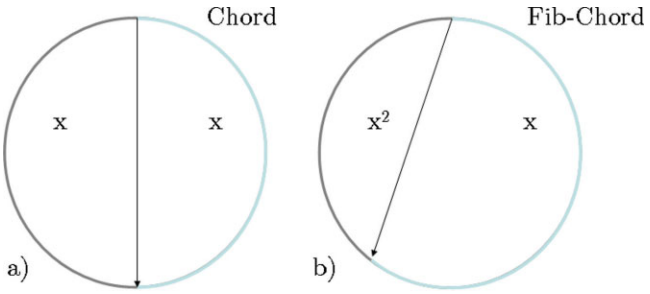

FIG. 1. (a) Chord's largest jump, (b) Fib-Chord's largest jump. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

For each index i, it holds

$$\mathrm{Fib}(i) = \left[\phi^i/\sqrt{5}\right],$$

where $\phi = \frac{1+\sqrt{5}}{2}$ is the golden ratio and [ ] represents the nearest integer function.

Now, we shall describe the process that has led to the results of this paper. The starting observation was to relate Chord strategy to binary search: given a segment of the ring where an ID i is known to be located[2], if i is in the second half of the segment, then a jump is taken to the middle of the segment; otherwise the search recurs on the first half of the segment. The consideration was that if i is in the first half then no cost is payed, i.e., no hop is taken. Therefore, we pondered if better performances could be achieved by increasing the size of the first portion of the segment. Of course, the resizing of the intervals had to be done carefully: it does not make sense to increase disproportionately the first interval of the segment. Furthermore, it must be possible to recur on the remaining segments, e.g., if the size of the interval is a power of two, then also the subintervals must have size that is a power of two.

To fully illustrate the motivating idea, let us follow the normalization process described in [17]. Consider the ring with unitary size and IDs being points placed on it i.e. with values within $[0, 1)$. In this case, Chord's strategy is to use, as largest jump, a value that halves the unitary circle, i.e. the solution of the equation $1 - 2x = 0$ (see Fig. 1a). In this case, by recurring on the subintervals, the obtained jumps are $1/2$, $1/4, \ldots, 1/N$.

Our first idea was to subdivide the ring in two segments, having different sizes, such that the first one has size x and the second one has size x^2, then for each segments of size x^i we subdivide it in two segments of size x^{i+1} and x^{i+2}, therefore the size of the largest jump is the solution to the equation $1 - x - x^2 = 0$ (see Fig. 1b). The only positive solution x turns out to be $x = 1/\phi$.

In this scheme (that we call Fib-Chord, described in Section 3) the jumps have sizes x, x^2, $x^3, \ldots, x^\delta$, where δ is the smallest integer such that $x^\delta \leq 1/N$ (i.e. $\delta \geq \log_\phi N$). The reader will notice that, apparently from nowhere, the Fibonacci numbers popped out in this scenario; in fact, in a nonnormalized N-node ring, the size of the ith jump is a corresponding Fibonacci number[3] since the jumps are Nx, Nx^2, $Nx^3, \ldots, Nx^\delta$.

Fib-Chord scheme does improve the diameter of Chord. In fact, we observed that if, at a given point of the search strategy, we are left with an interval of the ring with size $\mathrm{Fib}(i)$, then if we choose to take the jump of size $\mathrm{Fib}(i-1)$ then the next jump will have size at most $\mathrm{Fib}(i-3)$ since the remaining segment has overall size $\mathrm{Fib}(i-2)$ (see Fig. 2). Therefore, we can conclude that Fib-Chord improves the diameter of Chord $\left(\text{i.e. } \frac{\log_\phi N}{2} < \log_2 N\right)$ because no consecutive size jumps are ever taken in a path. Unfortunately, the scheme Fib-Chord with Fibonacci numbers, while aesthetically appealing, had worst degree than Chord, i.e., $\log_\phi N > \log_2 N$.

The idea of using a different base (other than 2) for representing IDs (and consequently to route among them) is not new. It was already pointed out in [16], in fact, that any base $b < 2$ can be used for a Chord-like routing that trades off routing table size (i.e. degree) with number of hops (i.e. diameter). As a result, the jumps would be b^i, for each $i < \log_b N$ and, consequently, Chord's results are slightly changed: the degree is raised to $\log_b N$ while the diameter is lowered to $\log_{b/(b-1)} N$. By using Fib-Chord, i.e., a Chord-based scheme with base ϕ, one could, therefore, improve on the diameter by paying off a corresponding increase in the degree.

To reduce the routing table size, we proved that one can prune the routing table by eliminating half of the jumps, carefully chosen, and still have a working search algorithm. By this result we could obtain a scheme, $\mathrm{Fib}\left(\frac{1}{2}\right)$-Chord, that improves both diameter and degree with respect to Chord.

Then, we considered the average path length of our new schemes and (unfortunately) proved (in Section 4) that by pruning half of the jumps the average path length of $\mathrm{Fib}\left(\frac{1}{2}\right)$-Chord was worst than Chord's. Therefore, our last step was to define a family of routing schemes, F-Chord(α), where α is a tuning parameter that drives the amount of pruning. For $\alpha = 1$, F-Chord(1) = Fib-Chord and when $\alpha = \frac{1}{2}$, F-Chord(1/2) = $\mathrm{Fib}\left(\frac{1}{2}\right)$-Chord.

We proved that, for a range of α, our scheme improves Chord on all the three parameters, allowing a trade-off between efficiency (i.e. average path length) and memory requirements[4] (i.e. degree).

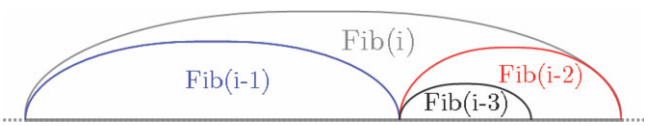

FIG. 2. If the jump of size $\mathrm{Fib}(i-1)$ is taken, then the next has at most $\mathrm{Fib}(i-3)$ size. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

[2]We assume that our search begins on the left-hand side of the segment.

[3]If N is a Fibonacci number, then $\frac{N}{\phi^i}$ is also a Fibonacci number.

[4]By reducing the routing table size, the cost of the operations on the table is also reduced.

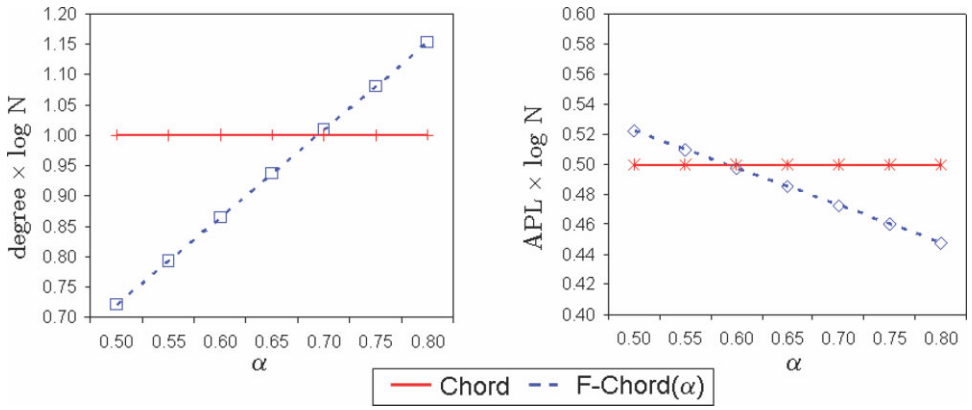

FIG. 3. Here we show the comparison between Chord and F-Chord(α) about *(left)* degree and *(right)* average path length (APL). [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

Finally, in Section 5 we proved that any uniform algorithm that uses up to δ jumps and has diameter d can reach at most $N(\delta, d)$ consecutive identifiers where $N(\delta, d) \leq \text{Fib}(\delta + d + 1)$. This gives us a tradeoff of $1.44042 \log N$ on the sum of the degree and the diameter in any P2P network using uniform routing on N identifiers. This lower bound implies that F-Chord(1/2) is optimal with respect to diameter and routing table size.

3. FIBONACCI-BASED CHORD-LIKE SYSTEMS

The purpose of this section is to introduce our schemes that are based on the Fibonacci number system applied to a Chord-like network, namely a ring of N identifiers labeled from 0 to $N-1$. We first define the generic Chord extensions to Fibonacci numbers Fib-Chord and Fib$\left(\frac{1}{2}\right)$-Chord and then present the family of routing schemes F-Chord.

The Fibonacci Routing Scheme Fib-Chord. For sake of simplicity, we first introduce the Fibonacci routing scheme Fib-Chord. Let $N \in (\text{Fib}(m-1), \text{Fib}(m)]$. The scheme uses $m-2$ jumps of size Fib(i), for $i = 2, \ldots, m-1$. Recall that $m \approx \log_\phi N = 1.44042 \log N$.

The results of next section will imply, in particular, that, by using Fib-Chord over a set of N identifiers, the degree (i.e., the number of jumps) is $1.44042 \log N - 2$, the diameter (i.e., the maximum number of hops) is $0.72021 \log N$, and the average path length is $0.39812 \log N$.

The Fibonacci Half-cut Routing Scheme Fib$\left(\frac{1}{2}\right)$-Chord. The Fib$\left(\frac{1}{2}\right)$-Chord routing scheme is obtained starting from Fib-Chord, by pruning all the jumps with odd indices. Formally, let $N \in (\text{Fib}(m-1), \text{Fib}(m)]$. The scheme uses $\left\lfloor \frac{m-1}{2} \right\rfloor$ jumps of size Fib($2i$), for $i = 1, 2, \ldots, \left\lfloor \frac{m-1}{2} \right\rfloor$.

The Family of Routing Schemes F-Chord. Intuitively, we obtain the family of routing schemes F-Chord by taking Fib-Chord and pruning them starting from smaller size jumps. The pruning is realized by eliminating a certain quantity (that is related to the parameter α) of the jumps with odd indices. More formally, our family is defined below.

Definition 1 (F-Chord). *Let* $N \in (\text{Fib}(m-1), \text{Fib}(m)]$ *and* $\alpha \in [1/2, 1]$. *The F-Chord*(α) *scheme uses the* $\lceil \alpha(m-2) \rceil$ *jumps*

$$\text{Fib}(2i), \text{ for } i = 1, \ldots, \lfloor (1-\alpha)(m-2) \rfloor$$

and

$$\text{Fib}(i), \text{ for } i = 2\lfloor (1-\alpha)(m-2) \rfloor + 2, \ldots, m-1.$$

Notice here that Fib-Chord = F-Chord(1); moreover, Fib$\left(\frac{1}{2}\right)$-Chord = F-Chord(1/2).

4. PROPERTIES OF F-CHORD

In this section, we investigate and bound the main parameters of the F-Chord systems (see Fig. 3).

For the rest of this section, we always assume that $N \in (\text{Fib}(m-1), \text{Fib}(m)]$ and $\alpha \in [1/2, 1]$.

Theorem 1. *For any value of* α, *the diameter of F-Chord*(α) *is* $\lfloor m/2 \rfloor \approx 0.72021 \log N$.

Proof. Since we use the greedy strategy it is enough to consider the case $\alpha = 1/2$, i.e., when there are only jumps of even Fibonacci index.

We can prove, by induction, that after performing i jumps, the search interval is reduced to Fib($m - 2i + 1$). In fact, if $i = 1$ then we are considering a jump of size greater or equal to Fib($m-2$), and since $N \leq \text{Fib}(m)$ the search interval is bounded by $\text{Fib}(m) - \text{Fib}(m-2) = \text{Fib}(m-1)$. Consider the situation after t jumps. If we took a jump in the previous step, the size of the search interval is bounded by Fib($m - 2t + 1$). Hence, $\text{Fib}(m-2t+1) - \text{Fib}(m-2t) = \text{Fib}(m-2t-1)$. If we did not take a jump in the previous step, the search interval is bounded by Fib($m - 2t$). Performing the jump restricts the search interval to $\text{Fib}(m-2t) - \text{Fib}(m-2t-2) = \text{Fib}(m-2t-1)$. For $i = \lfloor m/2 \rfloor$ the interval is thus reduced to Fib(2) = 1 and the search ends. ∎

Average Path Length. We can precisely evaluate the average path length of F-Chord(α). Let us, first, denote by $L_1(i, m)$ (resp. $L_{1/2}(i, m)$) the load for an edge of size Fib(i) on all

intervals of size $\mathrm{Fib}(m)$ when we use F-Chord(1) (resp. F-Chord(1/2)).

The following properties can be easily obtained:

1. $L_1(m,m) = 0$;
2. $L_1(m-1,m) = \mathrm{Fib}(m-2)$;
3. $L_1(i,m) = L_1(i,m-1) + L_1(i,m-2)$, for $i < m-1$.

Indeed, if the destination is one of the first $\mathrm{Fib}(m-1)$ nodes in the interval, no jumps of size $\mathrm{Fib}(m-1)$ are required. On the other hand, if the destination is one of the last $\mathrm{Fib}(m-2)$ nodes, to go from the first part of the interval to the second one the routing algorithm performs a jump of size $\mathrm{Fib}(m-1) \neq \mathrm{Fib}(i)$. That jump is hence not counted in $L_1(i,m)$.

Moreover, for $L_{1/2}$ it can be proved that:

4. $L_{1/2}(m-1,m) = \mathrm{Fib}(m-2)$;
5. $L_{1/2}(m-2,m) = \mathrm{Fib}(m-1) + \mathrm{Fib}(m-3)$;
6. $L_{1/2}(i,m) = L_{1/2}(i,m-1) + L_{1/2}(i,m-2)$ for $i < m-2$.

By using properties **1–6**, we can prove the following lemma.

Lemma 1. *It holds that:*

a) $L_1(i,m) = \mathrm{Fib}(i-1)\mathrm{Fib}(m-i)$ *for* $2 \le i \le m-2$.
b) $L_{1/2}(2i,m) = \mathrm{Fib}(2i-1)\mathrm{Fib}(m-2i) + \mathrm{Fib}(2i+1)\mathrm{Fib}(m-2i-1)$ *for* $2 \le 2i \le m-1$.

Proof. We show here the proof of the property **(a)**. However, the proof of the property **(b)** is quite similar. We prove, by induction on $m-i$, that $L_1(i,m) = \mathrm{Fib}(i-1)\mathrm{Fib}(m-i)$ for $2 \le i \le m-2$.

Let the base case be established for $m-i=2$:

$$\begin{aligned} L_1(m-2,m) &= L_1(m-2,m-1) + L_1(m-2,m-2) \\ &= \mathrm{Fib}(m-3) = \mathrm{Fib}(m-3)\mathrm{Fib}(2). \end{aligned}$$

Our inductive step is when $m-i=t$:

$$\begin{aligned} L_1(m-t,m) &= L_1(m-t,m-1) + L_1(m-t,m-2) \\ &= \mathrm{Fib}(t-1)\mathrm{Fib}(m-t-1) + \mathrm{Fib}(t-2) \\ &\quad \times \mathrm{Fib}(m-t-1) \\ &= \mathrm{Fib}(t)\mathrm{Fib}(m-t-1) = \mathrm{Fib}(i-1) \\ &\quad \times \mathrm{Fib}(m-i). \end{aligned}$$

∎

Theorem 2. *The average path length of the F-Chord(α) scheme, is upper-bounded by*

$$0.39812 \log N + (1-\alpha)0.24805 \log N + 1. \qquad (1)$$

Proof. For sake of simplicity, we assume that $N = \mathrm{Fib}(m)$. The sum of the total link load in the network, denoted by $S_\alpha(m)$, equals the sum of the lengths of each path.

$$\begin{aligned} S_\alpha(m) &= \sum_{i=1}^{\lfloor (1-\alpha)(m-2)\rfloor} L_{1/2}(2i,m) + \sum_{i=2\lfloor (1-\alpha)(m-2)\rfloor+2}^{m-1} L_1(i,m) \\ &= \sum_{i=1}^{\lfloor (1-\alpha)(m-2)\rfloor} \mathrm{Fib}(2i-1)\mathrm{Fib}(m-2i) + \mathrm{Fib}(2i+1) \\ &\quad \times \mathrm{Fib}(m-2i-1) \\ &\quad + \sum_{i=2\lfloor (1-\alpha)(m-2)\rfloor+2}^{m-1} \mathrm{Fib}(i-1)\mathrm{Fib}(m-i) \\ &= \sum_{i=1}^{m-2} \mathrm{Fib}(i)\mathrm{Fib}(m-i-1) \\ &\quad + \sum_{i=1}^{\lfloor (1-\alpha)(m-2)\rfloor} \mathrm{Fib}(2i-1)\mathrm{Fib}(m-2i-1) \end{aligned}$$

By using the property

$$\sum_{i=0}^{p} \mathrm{Fib}(i) \cdot \mathrm{Fib}(p-i) = \frac{1}{5}\{p[\mathrm{Fib}(p+1)+\mathrm{Fib}(p-1)] - \mathrm{Fib}(p)\},$$

we have

$$\begin{aligned} S_\alpha(m) &= \sum_{i=1}^{m-2} \mathrm{Fib}(i)\mathrm{Fib}(m-i-1) \\ &\quad + \sum_{i=1}^{\lfloor (1-\alpha)(m-2)\rfloor} \mathrm{Fib}(2i-1)\mathrm{Fib}(m-2i-1) \\ &= \frac{1}{5}[(m-1)(\mathrm{Fib}(m) + \mathrm{Fib}(m-2)) - \mathrm{Fib}(m-1)] \\ &\quad + \sum_{i=1}^{\lfloor (1-\alpha)(m-2)\rfloor} \mathrm{Fib}(2i-1)\mathrm{Fib}(m-2i-1) \\ &= \frac{1}{5}[(m-1)(\mathrm{Fib}(m) + \mathrm{Fib}(m-2)) - \mathrm{Fib}(m-1)] \\ &\quad + \sum_{i=3}^{\lfloor (1-\alpha)(m-2)\rfloor} \mathrm{Fib}(2i-1)\mathrm{Fib}(m-2i-1) \\ &\quad + \mathrm{Fib}(m-3) + 2\mathrm{Fib}(m-5). \end{aligned}$$

Hence, one can find the desired value of the average path length by observing that:

$$\begin{aligned} \frac{S_\alpha(m)}{\mathrm{Fib}(m)} &< \frac{1}{5}\left[(m-1)\left(1+\frac{1}{\phi^2}\right) - \frac{1}{\phi}\right] \\ &\quad + \frac{(1-\alpha)(m-2)\mathrm{Fib}(5)\mathrm{Fib}(m-7)}{\mathrm{Fib}(m)} + 1 \end{aligned}$$

∎

The following are immediate corollaries of the above Theorems 1 and 2.

Corollary 1. *The F-Chord(1) scheme (= Fib-Chord) has degree* $1.44042 \log N - 2$*, diameter equal to* $0.72021 \log N$*, and the average path length is* $0.39812 \log N$.

Corollary 2. *The F-Chord*$(1/2)$ *scheme has degree and diameter both equal to* $0.72021 \log N$ *and the average path length is* $0.52215 \log N + 1$.

Corollary 3. *For each* $\alpha \in [0.58929, 0.69424]$*, the F-Chord*$(\alpha)$ *schemes improve on Chord in all parameters (degree, diameter, and average path length).*

Edge Congestion. We remind the reader that, by using a uniform scheme, there is no node congestion [17]. Therefore, here we focus on edge load, that is defined as the number of times it is used by all routes from every node to every other node.

Analysis of $L_1(i,m)$ and $L_{1/2}(i,m)$ shows that $L_{1/2}(i,m)$ is decreasing with $i > 1$, whereas $L_1(i,m)$ decreases until it reaches its minimum at $i = \lceil m/2 \rceil$, after which it is increasing. Hence, the maximum and the minimum loads are given by

$$\begin{aligned} max\{L_1(i,m), L_{1/2}(i,m)\} &= L_{1/2}(2,m) = \mathrm{Fib}(m-1) \\ &\quad + \mathrm{Fib}(m-3) \\ min\{L_1(i,m), L_{1/2}(i,m)\} &= L_1(\lceil m/2 \rceil, m) \\ &= \mathrm{Fib}(\lceil m/2 \rceil - 1)\mathrm{Fib}(\lceil m/2 \rceil), \end{aligned}$$

and the ratio between the two is bounded by $9/4$, which can be viewed as the worst possible congestion. On the other hand, in the literature congestion is sometimes defined as the maximum load divided by the average load. With this definition the congestion g_α for the F-Chord(α) scheme lies between $g_{1/2} \leq g_\alpha \leq g_1$, and the boundaries are swiftly calculated to $g_{1/2} = 1.18034$ and $g_1 = 1.38197$.

4.1. A comment on scalability

In Section 3 we obtain the F-Chord routing scheme by pruning Fib-Chord starting from smaller size jumps. Now we present another family of routing schemes that is obtained by taking Fib-Chord and pruning them, starting from larger size jumps. Formally:

Definition 2 (F_b-Chord). *Let* $N \in (\mathrm{Fib}(m-1), \mathrm{Fib}(m)]$ *and* $\alpha \in [1/2, 1]$. *The* F_b*-Chord*(α) *scheme uses the* $\lceil \alpha(m-2) \rceil$ *jumps*

$$\mathrm{Fib}(i), \text{ for } i = 2, \ldots, m - 2\lfloor (1-\alpha)(m-2) \rfloor$$

and

$$\mathrm{Fib}(2i), \text{ for } i = \left\lceil \frac{m - 2\lfloor (1-\alpha)(m-2) \rfloor}{2} \right\rceil + 1, \ldots, \lfloor (m-1)/2 \rfloor.$$

Our analysis (cfr. Theorems 1 and 2) has been carried out by considering the N-size identifier space. We use a standard technique (see [16] for technical details) to manage the situation when $n < N$ nodes are in the network. In this case, in fact, we assume that the n nodes are uniformly distributed at random on the N identifiers, therefore we can consider a ring of n intervals each with N/n identifiers.

Since in each interval there are at most $O(\log n)$ nodes w.h.p., any deterministic result on the diameter in terms of N can be easily translated in the same result in terms of n, with high probability. The same argument can be used for the degree. In fact, the distance between two nodes is at least N/n^2 w.h.p. If $\delta(N)$ is the degree when all the N identifiers are used in a ring of size N, then the number of jumps that are useful (and consequently stored) when only $n < N$ nodes are present is at most $2\delta(n)$, w.h.p.

In the case of F-Chord(1) and F-Chord(1/2), with $n \leq N$ nodes in the network, results can be easily rewritten in terms of n with high probability, as already noticed in [16, 17]. When choosing a parameter $\alpha \in (1/2, 1)$, the presence of $n < N$ nodes will automatically tune (due to the fact that the shortest jumps will point to the same node): to $\alpha(n) > \alpha$ if one is using F-Chord(α); to $\alpha_b(n) < \alpha$ if one is using F_b-Chord(α). Thus, it is possible to choose which member of the family is most suitable for the requirements of the application. If a lower average path length is desired one can choose F-Chord(α) (here, fixing α assures an upper bound of $\lceil \alpha(m-2) \rceil$ on the node degree for large values of n) while if a low degree is the main goal one can choose F_b-Chord(α) (in this case fixing α assures an upper bound on the average path length as in Theorem 2).

5. THE LOWER BOUND

In this section, we furnish a lower bound of $1.44042 \log N$ on the sum of the degree and the diameter in any P2P network using uniform routing on N identifiers. Namely, we prove the following theorem.

Theorem 3. *Let* $N(\delta, d)$ *denote the maximum number of consecutive identifiers obtainable through a uniform algorithm using up to* δ *jumps (i.e. degree* δ*) and diameter* d*. For any* $\delta \geq 0$*, and* $d \geq 0$*, it holds that*

$$\begin{aligned} &\textit{i) if } |\delta - d| \leq 1 \textit{ then } N(\delta,d) \leq \mathrm{Fib}(\delta + d + 1) \quad (2) \\ &\textit{ii) if } |\delta - d| > 1 \textit{ then } N(\delta,d) < \mathrm{Fib}(\delta + d + 1) \end{aligned}$$

Proof. We first prove i). We proceed by induction on the sum $\delta + d$. Trivially, $N(0,0) = N(1,0) = N(0,1) = 1 = \mathrm{Fib}(1) = \mathrm{Fib}(2)$. Assume that (2) holds for any δ and d with $\delta + d < x$. We will show that for any degree y and diameter z with $y + z = x$ it holds that $N(y,z) \leq \mathrm{Fib}(x+1)$. We distinguish three cases on the number of times the first (i.e. the biggest) jump is repeated. We recall that the assumption of a uniform algorithm implies that jumps can be used to build paths in a greedy manner with respect to their size, that is, on each path jumps of the same size can be assumed to be consecutive.

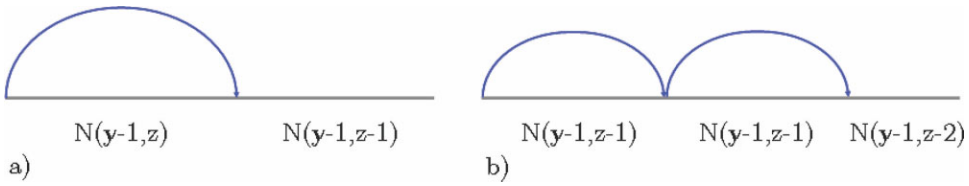

FIG. 4. (a) Case 1: Each jump appears at most once in each path. (b) Case 2: Each jump appears at most twice in each path. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

Case 1. The first jump appears at most once on each path. In this case, each path, either starting with the given jump or not, will never use this jump again. This implies that (cfr. Fig. 4a)

$$\begin{aligned} N_1(y,z) &\le N(y-1,z) + N(y-1,z-1) \\ &\le \mathrm{Fib}(y+z) + \mathrm{Fib}(y+z-1) = \mathrm{Fib}(x+1). \end{aligned} \quad (3)$$

Case 2. The first jump appears at most twice on each path. Since we have two equal jumps, the size of the jump cannot exceed the maximum number of nodes reachable using degree $y-1$ and diameter $z-1$ remaining after the first jump has been used (cfr. Fig. 4b). The part remaining after the second jump cannot exceed the maximum number of nodes reachable using the remaining degree $y-1$ and diameter $z-2$. Hence

$$\begin{aligned} N_2(y,z) &\le 2N(y-1,z-1) + N(y-1,z-2) \\ &\le 2\mathrm{Fib}(x-1) + \mathrm{Fib}(x-2) = \mathrm{Fib}(x+1). \end{aligned} \quad (4)$$

Case 3. The first jump appears at most ℓ times on each path for some $2 < \ell \le z$. As in Case 2, we can deduce that (cfr. Fig. 5)

$$\begin{aligned} N_3(y,z) &\le \ell N(y-1,z-\ell+1) + N(y-1,z-\ell) \\ &\le \ell\mathrm{Fib}(x-\ell+1) + \mathrm{Fib}(x-\ell). \end{aligned}$$

Since for each $2 < m \le z$

$$\begin{aligned} &m\mathrm{Fib}(x-m+1) + \mathrm{Fib}(x-m) \\ &\le (m-1)\mathrm{Fib}(x-m+1) + \mathrm{Fib}(x-m) + \mathrm{Fib}(x-m+1) \\ &< (m-1)(\mathrm{Fib}(x-m+1) + \mathrm{Fib}(x-m)) + \mathrm{Fib}(x-m+1) \\ &= (m-1)\mathrm{Fib}(x-m+2) + \mathrm{Fib}(x-m+1). \end{aligned} \quad (5)$$

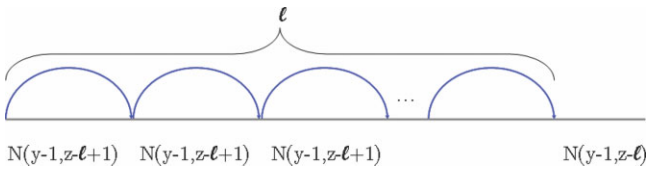

FIG. 5. Case 3: Each jump appears at most ℓ-times for each path. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

Hence

$$\begin{aligned} N_3(y,z) &\le \ell\mathrm{Fib}(x-\ell+1) + \mathrm{Fib}(x-\ell) \\ &< (\ell-1)\mathrm{Fib}(x-\ell+2) + \mathrm{Fib}(x-\ell+1) \\ &< (\ell-2)\mathrm{Fib}(x-\ell+3) + \mathrm{Fib}(x-\ell+2) \\ &< \dots \\ &< 2\mathrm{Fib}(x-1) + \mathrm{Fib}(x-2) = \mathrm{Fib}(x+1) \end{aligned} \quad (6)$$

where for each row we use the inequality (5). From inequalities (3), (4) and(6), we get

$$N(y,z) \le \max\{N_1(y,z), N_2(y,z), N_3(y,z)\} \le \mathrm{Fib}(x+1).$$

The same inductive proof as i) with base ($\delta = 2$, $d = 0$) and ($\delta = 0$, $d = 2$) proves ii). ∎

Because of Corollary 2 and Theorem 3 we obtain the following corollary.

Corollary 4. *The F-Chord(1/2) scheme is optimal with respect to diameter and degree.*

6. CONCLUSIONS AND OPEN PROBLEMS

We have described a family of simple algorithms that (1) improves uniform routing on Chord (see Fig. 6) and (2) is of practical interest. In fact, the designer can choose which member of the family is most suitable for the requirements of the application. For example, in a distributed file-system application one may want to prefer lower average path length over worst case (and thus choose F-Chord(1)), while

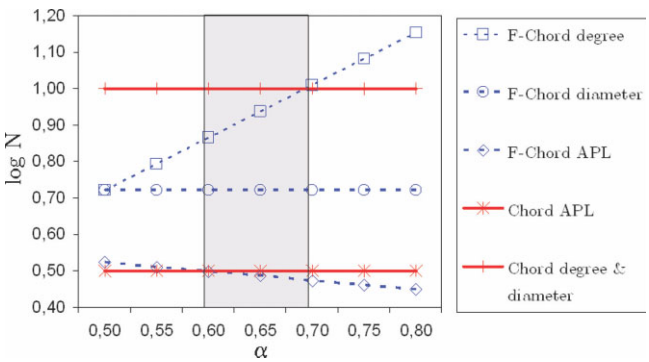

FIG. 6. Summary of results showed in Section 4 (APL denotes average path length). [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

in an application where fast delivery is paramount, one would choose a faster worst case over the average (i.e. F-Chord(1/2)) with a whole range of intermediate choices by using appropriate values of α.

Since any greedy routing requires $\Omega(\log N)$ hops when the degree is logarithmic [10], we believe that it is meaningful to improve the multiplicative constants in front of the $\log N$ since the results obtained by deterministic and uniform algorithms must be compared, practically, to the more theoretically appealing $O(\log N / \log\log N)$ that, for some values of N could have performances comparable to the deterministic and uniform algorithms (that are much much simpler to realize and deploy).

It eluded us to find an optimal average path length routing algorithm (once the node degree and the uniformity requirements are fixed). The search schemes used in F-Chord is close to optimal but not for all values of N.

Acknowledgments

The authors would like to thank the anonymous reviewers for their constructive comments to improve the quality of this paper.

REFERENCES

[1] G. Chiola, G. Cordasco, L. Gargano, M. Hammar, A. Negro, and V. Scarano, Degree-optimal routing for P2P systems, Theory of computing system, Springer, New York, in press.

[2] G. Cordasco, L. Gargano, M. Hammar, A. Negro, and V. Scarano, Non-uniform deterministic routing on F-Chord(α), Proc 1st Int Workshop Hot Topics in Peer-to-Peer Systems (HOT-P2P, in conjunction with MASCOTS 2004), IEEE Computer Society Press, Los Alamitos, CA, 2004, pp. 16–21.

[3] P. Druschel and A. Rowstron, Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems, Proc 18th IFIP/ACM Int Conf Distrib Systems Platforms, LNCS 2218, Springer Verlag, New York, 2001, pp. 329–350.

[4] D.-Z. Du and F.K. Hwang, Combinatorial group testing and its applications, World Scientific, Hackensack, NJ, 2000.

[5] S. Kapoor and E.M. Reingold, Optimum lopsided binary trees, J ACM 36 (1989), 573–590.

[6] M.F. Kaashoek and D.R. Krager, Koorde: A simple degree-optimal distributed hash table, Proc 2nd Int Workshop Peer-to-Peer Systems (IPTPS), LNCS 2735, Springer Verlag, New York, 2003, pp. 98–107.

[7] P. Ganesan and G.S. Manku, Optimal routing in chord, Proc ACM SIAM Symp Distrib Algorithms (SODA), SIAM, Philadelphia, PA, 2004, pp. 176–185.

[8] R.L. Graham, D.E. Knuth, and O. Patashnik, Concrete mathematics: A foundation for computer science, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1994.

[9] A. Kumar, S. Merugu, J. Xu, and X. Yu, Ulysses: A robust, low-diameter, low-latency peer-to-peer network, Proc IEEE Int Conf Network Protocols (ICNP), IEEE Computer Society Press, Los Alamitos, CA, 2003, pp. 258–267.

[10] G.S. Manku, M. Naor, and U. Wieder, Know thy neighbor's neighbor: The power of lookahead in randomized P2P networks, Proc 36th ACM Symp Theory Comput (STOC), ACM, New York, 2004, pp. 54–63.

[11] D. Malkhi, M. Naor, and D. Ratajczak, Viceroy: A scalable and dynamic emulation of the butterfly, Proc 21st ACM Symp Principles Distrib Comput (PODC), ACM, New York, 2002, pp. 183–192.

[12] M. Naor and U. Wieder, A simple fault tolerant distributed hash table, Proc 2nd Int Workshop Peer-to-Peer Systems (IPTPS), LNCS 2735, Springer Verlag, New York, 2003, pp. 88–97.

[13] M. Naor and U. Wieder, Novel architectures for p2p applications: The continous-discrete approach, Proc 15th ACM Symp Parallel Algorithms Architectures (SPAA), ACM, New York, 2003, pp. 50–59.

[14] S. Ratnasamy, S. Shenker, and I. Stoica, Routing algorithms for DHTs: Some open questions, Proc 1st Int Workshop Peer-to-Peer Systems (IPTPS), LNCS 2429, Springer Verlag, New York, 2004, pp. 45–52.

[15] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, A scalable content-addressable network, Proc ACM SIGCOMM, ACM, New York, 2001, pp. 161–172.

[16] I. Stoica, R. Morris, D. Liben-Nowell, D.R. Karger, M.F. Kaashoek, F. Dabek, and H. Balakrishnan, Chord: A scalable peer-to-peer lookup protocol for internet applications, IEEE/ACM Trans Networking 11 (2003), 17–32.

[17] J. Xu, A. Kumar, and X. Yu, On the fundamental tradeoffs between routing table size and network diameter in peer-to-peer networks, IEEE J Sel Areas Commun 22 (2004), 151–163.

[18] B.Y. Zhao, L. Huang, J. Stribling, S.C. Rhea, A.D. Joseph, and J. Kubiatowicz, Tapestry: A resilient global-scale overlay for service deployment, IEEE J Sel Areas Commun 22 (2004), 41–53.