Non-uniform deterministic routing on F-Chord(α)

Gennaro Cordasco, Luisa Gargano, Mikael Hammar, Alberto Negro, Vittorio Scarano
Dipartimento di Informatica ed Applicazioni "R.M. Capocelli"
Università di Salerno, 84081, Baronissi (SA) – Italy
E-mail: {cordasco,lg,hammar,alberto,vitsca}@dia.unisa.it

Abstract—**In this paper, we present a family of novel P2P routing schemes based on Chord [26] (and its variation F-Chord(α) [2]) that trades off uniformity with efficiency without using any additional overhead. We prove that H-F-Chord(α)'s routing is more efficient than in F-Chord(α) in terms of its average path length that is $O(\log n/\log\log n)$. We also show, by simulations, that H-F-Chord(α) is more efficient than the corresponding F-Chord(α) by a percentage that goes from 15% to 22% even for small n.**

I. Introduction

Peer-to-peer (P2P) systems are a class of network applications in which each node 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 (like storage and CPU cycles) is also common.

Maybe even more crucial than for other distributed systems, scalability is a central challenge in designing such systems. In fact, the amount of resources available to users is proportional to the number of users and, therefore, a successful P2P system must host a huge number of participants. In order to tackle the issue efficiently, a few P2P systems are based on Distributed Hash Table (DHT) schemes [6], [25], [26], [28]. 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.

Work partially supported by EU RTN project ARACNE and by Italian FIRB project WebMinds.

The efficiency of routing algorithms in P2P systems represents a key factor to ensure the scalability. A quick algorithm makes efficient not only the primitive operations (such as accessing or inserting data) but impacts also positively on maintenance, i.e., keeping the DHT consistent and in working order in spite of dynamic and unpredictable join/leave in the network as well as nodes failures. Of course, in practice, the traditional measures of routing efficiency (path length, average path length, etc.) must be supplemented with the consideration of how simple is the implementation of the routing algorithms in each node.

In this scenario, it is easily explained the popularity ([1], [6], [26], [16], [28]) of the greedy routing approach, where at each step message is routed through the neighbor which is nearest to the target. In fact, the greedy routing is very simple to implement and has some "implicit" fault-tolerance capability. Moreover, routing occurs between the portion of ring that is delimited by source and destination (locality in the key space) and, therefore, any eventual semantics present in the keys is not lost (e.g. if proximity between keys implies proximity between nodes) [22]. In general, however, this choice has a price. In fact, usually greedy routing produces paths of length larger than what would be required in a network of the given node degree. A typical example is Chord that has degree $O(\log n)$ and the greedy routing produces an average path length $O(\log n)$ whereas the lower bound is $\Omega(\log n/\log\log n)$.

Optimal tradeoffs between degree and latency with deterministic routing can be obtained, for example constructions based on de Bruijn graphs (see [9], [21]) but these algorithms are not greedy and do not present locality since keys manipulation is necessary.

Recently, some of Chord's results were improved by introducing both randomization and a technique, called NoN (Neighborsof Neighbors) or 1-lookahead routing. This technique consists in increasing the set where the greedy choice is picked so that it includes also the nodes at distance two from the source. The result is that it is possible (see [15], [22], [14]) to route "greedily" in $\Theta(\log N/\log\log N)$ with logarithmic degree.

In a sense, the NoN approach, with the use of randomization in establishing the nodes' neighbors[1], allows to maintain the advantages of greedy routing while optimizing the latency. It should be noticed that the NoN approach without randomization has been proved [10] ineffective for Chord, since the average path length is $\Omega(\log n)$.

As an example, while it is known that Chord is not degree-optimal (it uses $\log n$ degree and has average path length $(1/2)\log n$) it is emphasized in [22] that inserting randomization in the choice of each neighbor of a node and using NoN routing one can, with $\log n$ degree, make the average path length drop to $O(\log n/\log\log n)$.

Of course, in using the NoN approach, one must carefully analyze the additional overhead to store and update the neighbors of neighbors. If the argument in [14], [15], [22] about the storage is convincing (a node effectively stores $\log^2 n$ nodes but only updates $\log n$ of them (its neighbors)) possibly more problems can be hidden in the operation of propagating a change in its own list of neighbors to the neighbors. In fact, the only suggestions to tackle the issue is given in [15] and consists in using the TCP Keep-alive messages which looks, at the very least, not practical.

In this paper, we apply randomization and NoN approach to the F-Chord(α) P2P systems and show that, following the guidelines in [4], it is possible to reach the degree optimality of the network (thereby ensuring that the average path length is $O(\log n/\log\log n)$) while retaining simplicity (i.e., no hidden overhead), determinism, the "greedy" approach and, last but not least, an experimentally proved improved efficiency with respect to Chord and F-Chord(α).

[1] Randomization is inspired by the Small-world idea introduced by Kleinberg [11].

II. Preliminaries

A. Fibonacci Numbers

First, we, briefly, recall here some basic facts on Fibonacci numbers which will be used in the rest of the paper. Let $Fib(i)$ denote the i-th Fibonacci number. They are defined as $Fib(0) = 0, Fib(1) = 1$ and, for each $i>1$,

$$Fib(i) = Fib(i-1) + Fib(i-2).$$

For each index i, it holds

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

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

We consider a set N of n nodes lying on a ring of $Fib(m)$ identifiers (labeled from 0 to $Fib(m)-1$ in clockwise order). Each node x has an ID $id(x)$ and is connected with its predecessor $P(x)$ and its successor $S(x)$ on the ring. For each ID i, we denote by $R(i)$ the first (existing) node that is found clockwise from the ID i (i.e. $R(i) = x \Leftrightarrow \forall y \in N$ where $x \neq y, y \notin [i, id(x)]$).

B. The family of routing schemes F-Chord(α)

Our family F-Chord(α) is defined below:

F-Chord(α): Let $\alpha \in [1/2, 1]$, the F-Chord(α) schemes uses $\lceil \alpha(m-2) \rceil$ jumps of size

$$Fib(2i) \text{ for each } 1 \leq i \leq \lfloor (1-\alpha)(m-2) \rfloor$$

and

$$Fib(i) \text{ for each } 2\lfloor (1-\alpha)(m-2) \rfloor + 2 \leq i \leq m-1.$$

In particular if we denote by $j_1, j_2, \ldots, j_{\lceil \alpha(m-2) \rceil}$ all the jumps of our schemes (ordered by their size), for each $i \in [1, \lceil \alpha(m-2) \rceil]$ node x is connected by edges to the nodes[2] $R(id(x) + j_i)$.

F-Chord(α) is a Chord-like scheme [2] that retains all the characteristics that made Chord a popular topology for

[2] All the arithmetic operation on the ring are done $\bmod Fib(m)$.

routing in P2P networks but also improves on the maximum/average number of hops for lookups and the routing table size per node.

For $\alpha = 1$, our scheme (that we also call Fib-Chord) uses $m-2$ jumps of size $Fib(i)$ for each $1 \leq i \leq m-1$ (i.e., all the Fibonacci numbers up to n); for $\alpha = 1/2$, our scheme (Fib$\left(\frac{1}{2}\right)$-Chord) uses $\frac{m-2}{2}$ jumps of size Fib(2i) for each $2 \leq i \leq \left\lfloor \frac{m-1}{2} \right\rfloor$, that is, all the even-index Fibonacci numbers.

F-Chord(α) holds some interesting properties [2]: the diameter of F-Chord(α) is $0.72021 \log n$ for each value of α, while the degree (i.e. the number of jumps) and the average path length are respectively $1.44042\alpha \log n$ and $(0.39812 + (1-\alpha)0.24805) \log n$. It particular for each $\alpha \in [0.58929, 0.69424]$, the F-Chord($\alpha$) schemes improve on Chord with respect to the number of jumps, diameter and average path length. The F-Chord(1/2) scheme is optimal with respect diameter and degree. Furthermore F-Chord(α) schemes allowing a trade–off between efficiency/faulty (i.e. average path length) and memory requirements/maintenance (i.e. degree). The improvements are obtained without introducing routing algorithm and keeping the routing congestion-free.

C. The NoN approach on F-Chord(α)

As noticed before, the NoN approach without randomization is known not to be effective. As a consequence, in order to use the NoN approach over a P2P scheme one has, first, to create the randomized version of the network and, then, use effectively the NoN-Greedy protocol.

In our case, the first step consists in create a randomized version of F-Chord(α).

R-F-Chord(α):
We remind the reader that we denote by $j_1, j_2, \ldots, j_{\lceil \alpha(m-2) \rceil}$ all the jumps of our schemes (ordered by their size). Let $\alpha \in [1/2, 1]$, for each $i \in [1, \lceil \alpha(m-2) \rceil]$, let $r(i)$ denote an integer chosen uniformly at random from the interval $[0, j_{i+1} - j_i)$ (where $j_{\lceil \alpha(m-2) \rceil]+1} = n$), node x is connected by edges to the nodes $R(id(x) + j_i + r(i))$.

Now, let us define formally the NoN-Greedy protocol. We assume that each node holds its own routing table, and on top of that holds its neighbors routing tables. Let $d(x, y)$ be a metric for the nodes in the network. Here is the description of the NoN-Greedy routing protocol:

1. Assume the message is currently at node $u \neq$target.
2. Let $N = \{v_1, v_2, \ldots, v_k\}$ be the neighbors of u.
3. For each $1 \leq i \leq k$, let $w_{i1}, w_{i2}, \ldots, w_{ik}$ be the neighbors of v_i and let $N' = \{w_{ij} \ \forall \ 1 \leq i, j \leq k\}$.
4. Among these $k^2 + k$ nodes, assume that z is the one closest to the target (with respect to metric d).
5. If $z \in N$ route the message from u to z else $z = w_{ij}$, for some i and j, and we route the message from u **via** v_i to z.

Remark: the (standard) definition given above is called the *2-phases NoN* and it is the protocol that has been theoretically analyzed. A more efficient definition, call it *1-phase NoN*, can be obtained by replacing the step 5 as follows:

5'. If $z \in N$ route the message from u to z else $z = w_{ij}$, for some i and j and we route the message from u **to** v_i.

Intuition can easily support the claim of improved efficiency of 1-phase NoN with respect to 2-phases NoN: reapplying the whole algorithm at node v_i can only extend the choices. Furthermore, experiments showed that 2-phase NoN is slower than 1-phase NoN. Notice, finally, that while in our proofs we analyze 2-phase NoN, in our experiments, in Sec. IV we used (for all networks) the 1-phase NoN.

Following the guidelines of the proofs in [15], it is easy to prove that the average path length is $O(\log n / \log \log n)$ hops for the NoN-Greedy algorithm on R-F-Chord(α) in a ring of size $Fib(m)$ where the number of nodes alive is $n < Fib(m)$.

III. H-F-Chord(α)

In this section we describe the novel version of F-Chord(α) and show that the average path length is $O(\log n / \log \log n)$ hops for performing lookups, without adding any additional overhead to know the neighbors of neighbors. The routing table size is $O(\log n)$.

The key observation in our results is that generating one single random number for each node in the network is enough to preserve the $O(\log n/\log\log n)$ hops (on average). Routing occurs through canonical NoN. Observe that we can use any good hashing function (such as SHA, MD5, etc.) on node IDs to get a deterministic algorithm with the same properties.

By using hashing together with the new network we get the same results as the NoN protocol for R-F-Chord(α), but with a lower message complexity for maintaining the peer to peer network intact. In fact, by using a hashing function each node can evaluate the neighbors of its neighbors with no need to transmit additional information.

The network is defined as follows.

H-F-Chord(α):

Let $\alpha \in [1/2, 1]$ and H() denote a good hash function, that maps an *id* on a sequence of $\lceil \log Fib(m) \rceil$ bits. For each $i \in [1, \lceil \alpha(m-2) \rceil]$, node x is connected by edges to the nodes $R\left(id(x) + j_i + \left\lfloor \frac{H(id(x))}{2^{\lceil \log Fib(m) \rceil}} * (j_{i+1} - j_i) \right\rfloor\right)$.

We analyze the protocol as if the hash function generates an integer r taken uniformly at random from the interval $[0, Fib(m))$.

We first consider $n = Fib(m)$, then any n.

Lemma 1: For each value of $\alpha \in [0, 1/2]$, the average path length is $O(\log n/\log\log n)$ hops for the NoN Greedy algorithm on H-F-Chord(α) with $n = Fib(m)$ nodes.

Proof: We show the proof for H-F-Chord(1/2). In fact, by adding more jumps the average path length can only decrease.

We consider a source node v that wants to send a message a distance d. Let $I = [d - d', d]$, where $d' = \lceil d \log\log n/\log n \rceil$. Let p be the unique number with $Fib(p) \le d < Fib(p+2)$. There are two cases to consider. In the first case $p \le \frac{2\Phi \log n}{\log\log n}$. In this case $O(p)$ hops are sufficient to reach the destination, since, like in R-F-Chord(1/2), the distance decreases at least with a factor of $1 - 1/\Phi^4 \approx 0.854$ for each jump executed.

In the second case $p > \frac{2\Phi \log n}{\log\log n}$. Let us assume that, for any node $u \in [v, v+d]$, the probability that it has an outgoing edge entering the interval I is $d'/Fib(p+1)$. Now, v has at least $p/2$ neighbors $n_1, \ldots, n_{p/2}$ in the interval $[v, v+d]$. Each of these nodes have chosen its neighborhood independently from the others, hence, the probability that none of these nodes have an outgoing edge reaching the interval I is

$$\begin{aligned}\left(1 - \frac{d'}{Fib(p+1)}\right)^{p/2} &\le \left(1 - \frac{d \log\log n}{Fib(p+1)\log n}\right)^{p/2} \\ &\le \left(1 - \frac{Fib(p)\log\log n}{Fib(p+1)\log n}\right)^{p/2} \\ &\le \left(1 - \frac{\log\log n}{\Phi \log n}\right)^{\frac{\Phi \log n}{\log\log n}} \le e^{-1},\end{aligned}$$

since $p > \frac{2\Phi \log n}{\log\log n}$. Hence, the probability that v can reach the interval in two jumps is at least $1 - e^{-1}$. Thus, in $O(\log d/\log\log n)$ hops, the distance is decreased to $Fib(p')$, where $p \le \frac{2\Phi \log n}{\log\log n}$, and we have reduced case two into case one. Left is to prove the assumption.

Consider a node v at most a distance d from I. We are investigating the probability of v having an outgoing edge entering the interval I, i.e., $Pr[\exists j_i \in I]$. There are two cases to consider.

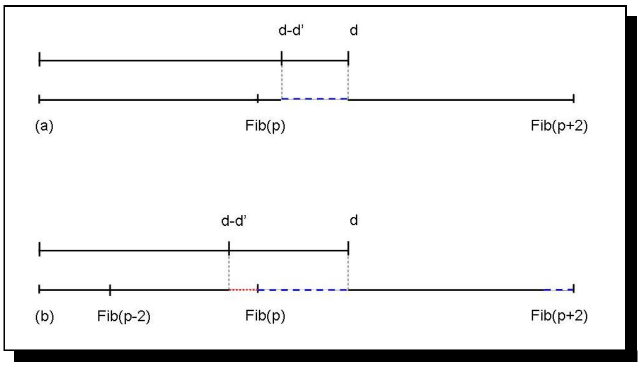

Fig. 1. (a) Case 1 ($d - d' > Fib(p)$) in Lemma 1; (b) Case 2 ($d - d' < Fib(p)$) in Lemma 1

Case 1: $d - d' > Fib(p)$ (see fig. 1 (a)). In this case the probability that $j_{p/2}$ reaches the interval is equal to $d'/Fib(p+1)$.

Case 2: $d - d' < Fib(p)$ (see fig. 1 (b)). The probability that one of the jumps reaches the interval I is equal to

$$Pr[j_{p/2-1} \in [d - d', Fib(p)) \cup j_{p/2} \in [Fib(p), d]].$$

We can observe that

$$j_{p/2} \in [d - d' + Fib(p+1), Fib(p+2)) \Rightarrow j_{p/2-1} \in [d - d', Fib(p)).$$

Hence,

$$\begin{aligned}Pr[\exists j_i \in I] &\ge Pr\Big[j_{p/2-1} \in [d - d', Fib(p)) \cup j_{p/2} \in [Fib(p), d]\Big] \\ &\ge Pr\Big[j_{p/2} \in [d - d' + Fib(p+1),\end{aligned}$$

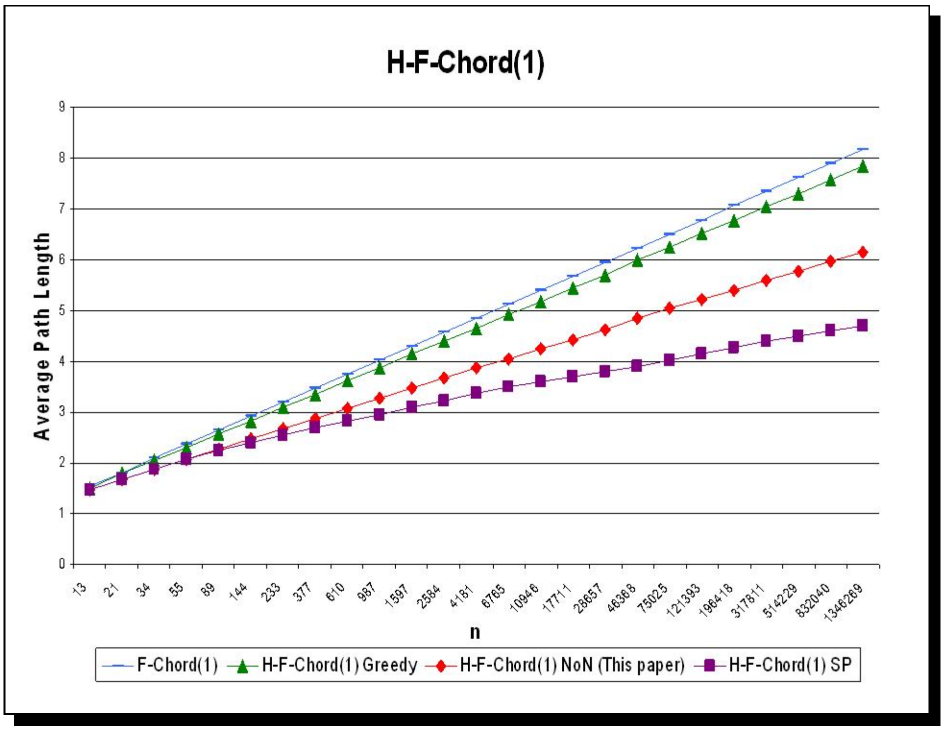

Fig. 2. Comparison between F-Chord(1) and H-F-Chord(1) (lower is better).

$$
\begin{aligned}
& Fib(p+2)) \cup j_{p/2} \in [Fib(p), d]\Big] \\
&= \frac{Fib(p+2) - d + d' - Fib(p+1) + d - Fib(p)}{Fib(p+1)} \\
&= \frac{d'}{Fib(p+1)}.
\end{aligned}
$$

■

By following a standard argument, the result can be generalized to hold in a ring where not all the nodes are present. In fact, since IDs are obtained by consistent hashing, the nodes present in the ring can be assumed to be uniformly distributed. By applying the same technique described in [4], H-F-Chord(α) uses the same network maintenance protocol as Chord, but without the extra overhead. In fact, information about the *exact* neighbors of neighbors is not available in H-F-Chord(α), since the nodes suggested by the hash-function might not be alive. However, the hash-function provides us with the node IDs of the neighbors neighbors and this information is sufficient to perform efficient lookups. The node ID is namely a lower bound on the distance to the neighbor in question. Therefore, we can state the following

Theorem 1: The average path length is $O(\log n/\log\log n)$ hops for the NoN Greedy algorithm on H-F-Chord(α) in a ring of size $Fib(m)$ where the number of nodes alive is $n < Fib(m)$.

Remark: *Before proving the theorem, it should be said that the result stated here is complemented by the results in Section IV where it is shown (1) that the improvement over the average path length in F-Chord(α) (that is $O(\log n)$) is substantial and there is no hidden constant to limit the relevance of the improvement only to unrealistically large n; (2) that the constant hidden in the Big-Oh notation for H-F-Chord(α) (for infinitely many α) is smaller than the corresponding constant of the same results (in magnitude) for the average path length in H-Chord.*

Proof: Consider a source sending a message a distance d. From Lemma 1 it follows that diminishing the distance to size $Fib(m)/n$ takes $O(\log n/\log\log n)$ hops. Left is to prove that the number of nodes alive in an interval I of size $Fib(m)/n$ is small. The expected number of nodes alive in the interval is

$$E[A] = E\left[\sum_{i=1}^{n} x_i \in I\right] = \sum_{i=1}^{n} E[x_i \in I]$$

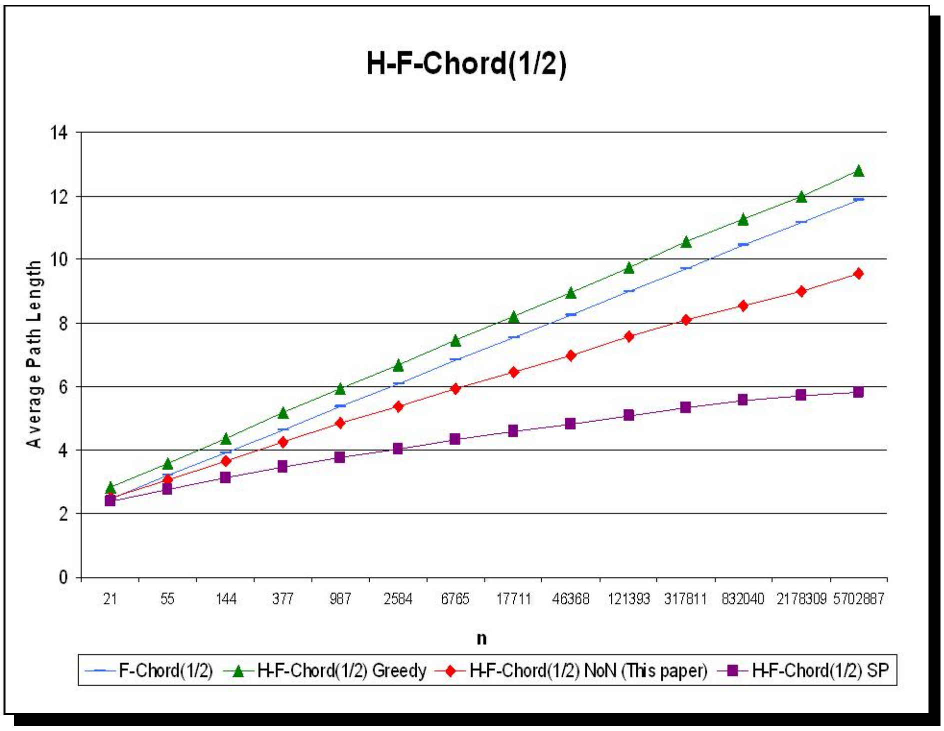

Fig. 3. Comparison between F-Chord(1/2) and H-F-Chord(1/2) (lower is better).

$$= \sum_{i=1}^{n} Pr[x_i \in I] = \sum_{i=1}^{n} \frac{Fib(m)}{n} \cdot \frac{1}{Fib(m)} = 1.$$

Furthermore, with probability larger than $1 - 1/n^2$, the number of nodes that lies in the same interval is $O(\log n / \log\log n)$, see Example 4.4 in [19]. ■

IV. Experimental results

In this section, our objective is to show that the improvement on the average path length that has been previously shown can be helpful in improving the performances even for small values of n. In fact, the simulations that we show here support the evidence that the H-F-Chord(α) is practically more efficient than F-Chord(α).

We report, here, some results of our validating simulations. We ran simulations to compare the performances of F-Chord(α) with respect H-F-Chord(α) either by greedy routing or by NoN routing. The performance are measured in terms of *average path length.* Furthermore, we evaluated also the performances of H-F-Chord(α) with an optimal routing scheme that routes messages along the shortest paths (SP).

Our goal was to show that no hidden constant (in the big-Oh notation) in the theoretical results of the previous sections could limit the significance of the deterministic networks that we propose here.

The choices of parameter α to be used in the experiments is based on the need to propose a comprehensive view along all the range. Therefore, we picked $\alpha = 1$ in order to show the behavior on a scheme that uses all the jumps based on Fibonacci numbers. Then, at the other end, we chose $\alpha = 1/2$ as an example of a scheme that uses fewer jumps and is cheaper to store and maintain dynamically. Between the two extremes, we finally chose $\alpha = 0.69424$ since it is the value that allows F-Chord(α) to be more efficient than Chord [2] with respect to average path length and diameter but with the same degree.

The experiments were conducted evaluating the average path length on all pairs of nodes in the network with sizes less than or equal to $Fib(30) = 832040$. For larger n, because of efficiency, we estimated the average path length for n nodes by evaluating the average path length on $\log n$ randomly chosen subsets of $\log n$ size: for each node in the subset we considered the paths to all the $n-1$ nodes. Our restriction has been validated for smaller sizes and it gives the same results as the exhaustive evaluation. The hash function used is the well-known SHA-1 [23].

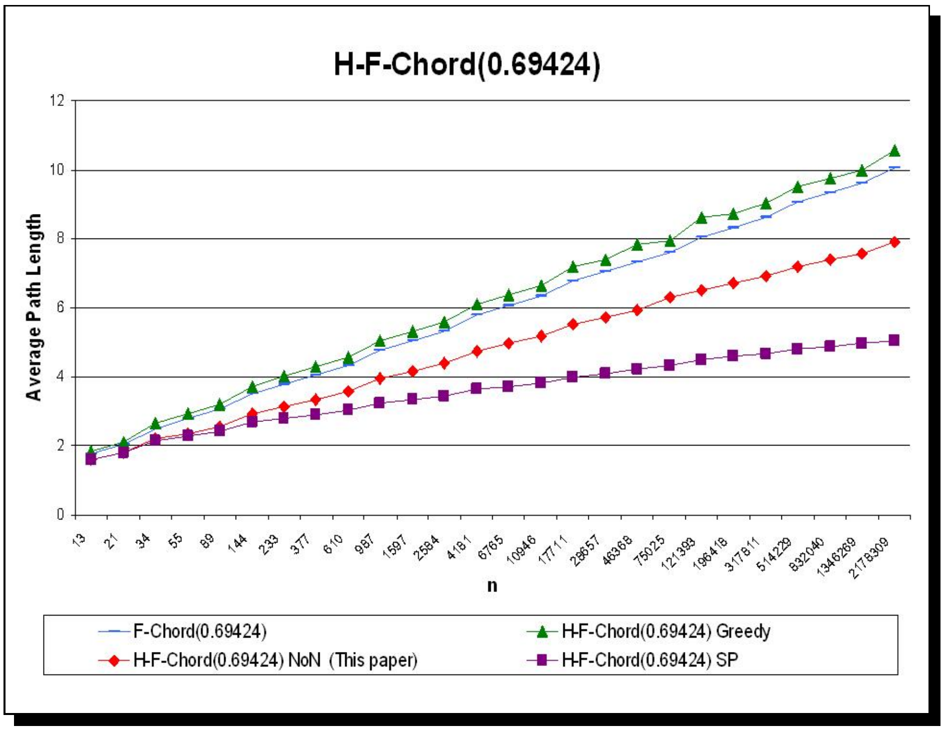

Fig. 4. Comparison between F-Chord(0.69424) and H-F-Chord(0.69424) (lower is better).

In Fig. 2, 3 and 4 we show, for α equals, respectively, to 1, 1/2 and 0.69424 the average path length of H-F-Chord(α), by using a greedy routing, the NoN routing with a hash function (as described in this paper), and an optimal, off-line shortest path (SP) routing algorithm (shown here only to compare to the most efficient routing that can be obtained). Since F-Chord(α) is a deterministic and uniform scheme, both greedy routing, NoN routing and SP routing obtain the same performance (as proved in [27]) and, therefore, we only show in the diagrams F-Chord(α) with the optimal (greedy) routing.

In general, as one can see in Fig. 6, the improvements on the average path length by our schemes are larger when α is larger.

It should be noticed, in Fig. 3, that because of the structure of F-Chord(1/2), the experiments only tested sizes equals to even index Fibonacci numbers.

Finally, a comment is due for Figs. 4, 5 and 6 in order to explain some anomalies in the (otherwise) regular shape of the curve of H-F-Chord(0.69424). In fact, the apparent irregularities in the performances derive mainly by the necessary approximation (to integers) of the degree. In some cases, in fact, enlarging the size from $Fib(m)$ to $Fib(m+1)$ does not really add a new available jump, because of the arithmetics. To wit, for $Fib(11)$ nodes and $Fib(12)$ nodes, our schemes use the same number of jumps (i.e., respectively, $1, 3, 8, 13, 21, 34, 55$ and $1, 3, 8, 21, 34, 55, 89$) on rings of different dimensions.

V. Conclusions

We propose a family routing schemes that optimize the average number of hops for lookup requests in Peer-to-Peer systems, allowing a trade–off between efficiency and maintenance), without adding any overhead to the system. As graphically summarized by Fig. 6, the improvements of average path length over F-Chord(α) vary between 6% and 16% even for very small values of n (i.e. larger than 100) and that 10% improvement is guaranteed for n larger than 1000.

Our work is inspired by the recently introduced variation of greedy routing, called neighbor-of-neighbor (NoN), which allows to get optimal average path length with respect to the degree, paying a small overhead compared to previous systems due to additional network maintenance.

Our proposal has the advantage of "limiting" randomization to such an extent that neighborhood information can

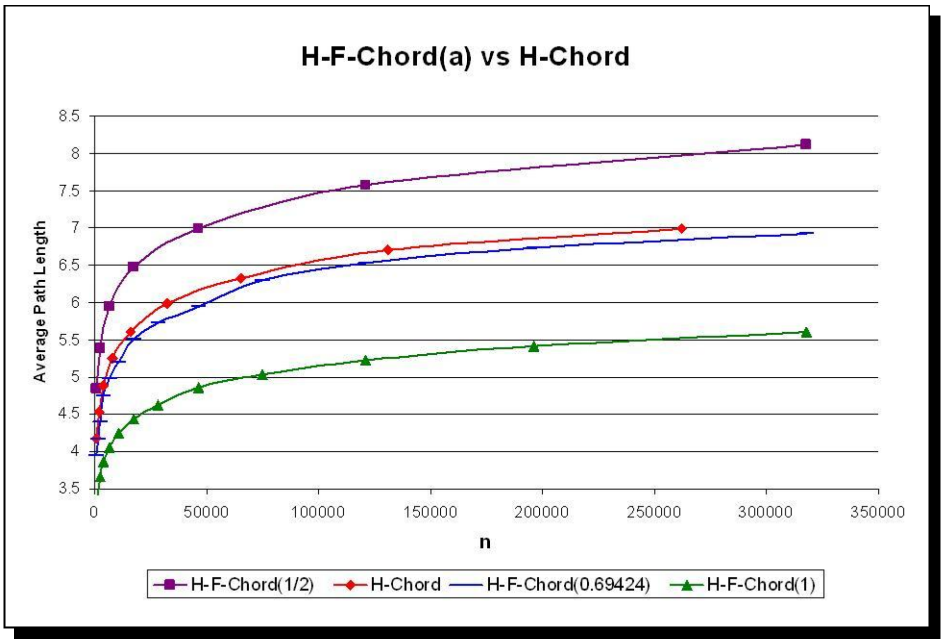

Fig. 5. Comparing our technique as applied to Chord (thus obtaining the H-Chord scheme described in [4]) with respect to the results obtained by H-F-Chord(α) (lower is better).

be encoded within the hash–value of the node ID. This enables us to use NoN lookup routing without the additional overhead that was present in their original formulation. In our opinion, the overhead to keep the list of neighbors of neighbors updated is not to be easily dismissed in practice. As a matter of fact, the implementation of additional update mechanisms (however smart they can be, in order to minimize the amount of exchanged data [22]) does require additional coding and testing with respect to our solution. Then, the cost of exchanging updates on the neighbors of neighbors has to be (somewhere) taken into account into the stabilization protocol, since the suggestion of hiding such updates messages on TCP Keep-alive messages (by piggybacking) seems unrealistic [15].

Another consideration about our technique is based on the original "six degrees of separation" experiment by Milgram [18]. The sociological experiment relied on social networks to transmit a letter from a person to unfamiliar targets by passing the letter only via acquaintances. The work by Milgram (who originated the term *small world*) found that a surprising small number of steps was needed (around 6). In computer network setting, this is just a plain greedy algorithm. Recent work by [5], shows that, in the first steps the message was forwarded to a person P by using a guess on who P knew or, in our words, on his/her neighbors (like the NoN approach). It is interesting to notice that our flavor of Neighbor of Neighbor greedy routing, based on a hash function, looks more genuinely close to the sociological experiment since a node A (sending the message to B) does not know exactly which are B's neighbors but has a way (via the hashing function) of evaluate approximately which neighbors the node B can have.

About the performances, our approach, basically, keeps deterministic the path between two nodes (in terms of IDs traveled through). Therefore, one can fruitfully include in our scheme some caching mechanisms to improve performances. For example, one might keep copies of the data along the path, and move the copies with the corresponding IDs when new nodes appear in the network.

Finally, it can be argued that the additional overhead in traditional NoN routing is too small to be a nuisance. However, there is a problem that these P2P-systems all have in common. As stressed in [26]:

"*In practice, a Chord ring will never be in a stable state; instead, joins and departures will occur continuously, interleaved with the stabilization algorithm. The ring will not have time to stabilize*

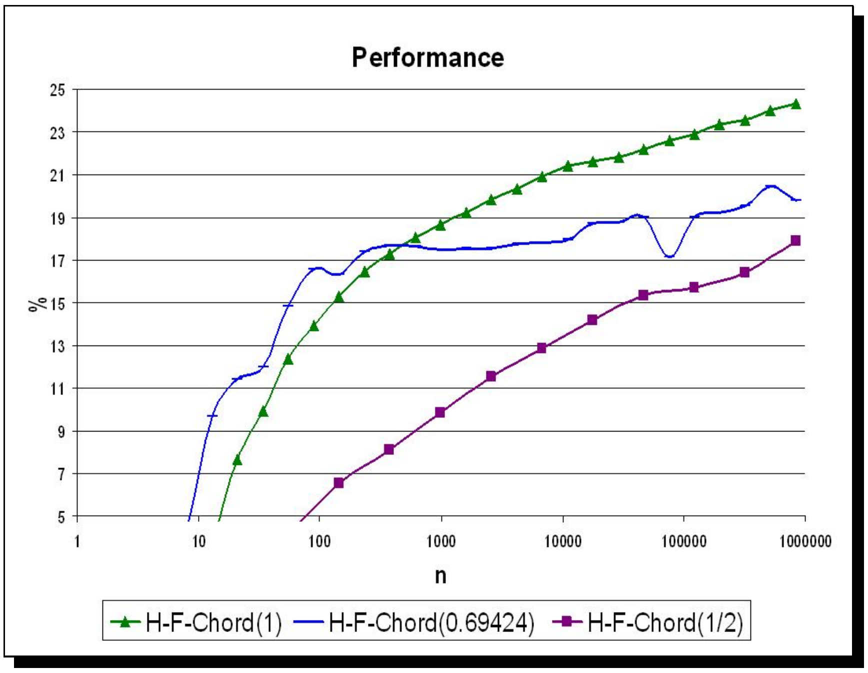

Fig. 6. Performance improvements (in percentage) of H-F-Chord(α) average path length with respect to F-Chord(α) average path length (logarithmic scale).

before new changes happen."

This leaves R-Chord (or any other randomized Chord-like system) in an inconsistent state that in the end will incur failures in choosing lookup paths. Furthermore, this problem is relevant also for the other P2P-systems considered in [15]. We believe that our technique resolves this precarious situation in an elegant and, most important, efficient way.

References

[1] J. Aspnes, G.S hah, "Skip Graphs", in Proceedings of ACM Symposium on Discrete Algorithms (SODA 03).

[2] G. Cordasco, L.Gargano, M. Hammar, A. Negro, V. Scarano, "F-Chord: Improved Uniform Routing on Chord", in Proceedings of 11th Colloquium on Structural Information and Communication Complexity (SIROCCO 2004) June 21-23, 2004, Smolenice Castle, Slovakia. Springer-Verlag, Lecture Notes in Computer Science, Vol. 3104.

[3] G. Cordasco, L.Gargano, M. Hammar, V. Scarano, "Brief-announcement: Degree-Optimal Deterministic Routing for P2P Systems", in Proceedings of Twenty-Third Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC 2004) July 25-28, 2004, St. John's, Newfoundland, Canada.

[4] G. Cordasco, L.Gargano, M. Hammar, V. Scarano, "Degree-Optimal Deterministic Routing for P2P Systems", submitted for publication (September 2004).

[5] P. S. Dodds, M. Roby, D. J. Watts. "An experimental study of search in global social networks". Science, 301:827829, 2003.

[6] P. Druschel and A. Rowstron, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems" in Proceedings of the 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001), Nov. 2001.

[7] D. Du, F. K. Hwang. "Combinatorial Group testing and its applications". World Scientific, 2000.

[8] S. Kapoor, E.M.Reingold, "Optimum Lopsided Binary Trees", Journal of ACM, Vol. 36, No.3, July 1989, pp. 573–590.

[9] M. F. Kaashoek and D. R. Krager, "Koorde: A simple degree-optimal distributed hash table", in Proceedings of 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03), Feb. 20-21, 2003, Berkeley, CA (USA).

[10] P. Ganesan, G. S. Manku, "Optimal Routing in Chord", in Proceedings of 15th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2004, p 169-178, Jan 2004.

[11] J. Kleinberg, "The Small-world phenomenon: An algorithmic prospective", in Proceedings of ACM Symposium on Theory of Computing 2000 (STOC 00).

[12] R. Graham, O. Patashnik, D.E. Knuth "Concrete Mathematics", Addison-Wesley, 1994.

[13] A. Kumar, S. Merugu, J. Xu and X. Yu, "Ulysses: A Robust, Low-Diameter, Low-Latency Peer-to-peer Network", in Proceedings of IEEE International Conference on Network Protocols (ICNP 2003), Atlanta, Nov. 2003.

[14] G. S. Manku, "The Power of Lookahead in Small-World Routing Network", Technical Report, CS Department, Stanford University, Nov. 2003.

[15] G. S. Manku, M. Naor, U. Wieder. "Know thy Neighbor's Neighbor: The Power of Lookahead in Randomized P2P Networks", in Proceedings of 36th ACM Symposium on Theory of Computing (STOC) 2004, 54-63, June 2004.

[16] G.S.Manku, M.Bawa, P.Raghavan, "Symphony: Distributed

hashing in a Small World", in Proceedings of USITS 03.

[17] D. Malkhi, M. Naor and Ratajczak, "Viceroy: A Scalable and Dynamic Emulation of the Butterfly", in Proceedings of the 21st ACM Symposium on Principles of Distributed Computing (PODC 2002), Aug. 2002.

[18] S. Milgram. "The small world problem." Psychology Today, 67(1):6067, May 1967.

[19] Rajeev Motwani and Prabhakar Raghavan, "Randomized Algorithms", Cambridge University Press, 1995.

[20] M. Naor and U. Wieder, "A Simple Fault Tolerant Distributed Hash Table," in Proceedings of 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03), Feb. 20-21, 2003, Berkeley, CA (USA).

[21] M. Naor and U. Wieder, "Novel architectures for p2p applications: the continous-discrete approach." in Proceedings of 15th ACM Symposium on Parallel Algorithms and Architectures (SPAA), 2003.

[22] M. Naor, U. Wieder. "Know thy Neighbor's Neighbor: Better Routing for Skip-Graphs and Small Worlds", in 3rd International Workshop on Peer-to-Peer Systems (IPTPS'04), Feb. 26-27, 2004, San Diego, CA (USA). Post-proceedings to be published in Springer-Verlag Lecture Notes in Computer Science, Hot Topics series.

[23] National Institute of Standards and Technology, Secure Hash Standard, www.itl.nist.govfipspubsfip180-1.htm.

[24] S. Ratnasamy, S. Shenker, and I. Stoica, "Routing Algorithms for DHTs: Some Open Questions", in Proceedings for the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02), Mar. 7-8 2002, Cambridge, MA (USA), Springer-Verlag Lecture Notes in Computer Science (LNCS) 2429.

[25] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S.Shenker, "A scalable content-addressable network", in Proceedings of ACM SIGCOMM, Aug. 2001.

[26] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, H. Balakrishnan, "Chord: A Scalable Peer-to-peer Lookup Protocol for Internet Applications", in IEEE/ACM Trans. on Networking, 2003.

[27] Xu, J., Kumar, A., and Yu, X., "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, Jan 2004. A preliminary version appeared in the Proceedings of IEEE INFOCOM 2003, May 2003.

[28] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, J. Kubiatowicz, "Tapestry: A Resilient Global-scale Overlay for Service Deployment", in IEEE Journal on Selected Areas in Communications, Jan. 2004, Vol. 22, No. 1.