Overlay networks with class [*]

G. Chiola
DISI, Università di Genova
via Dodecaneso 35, Genova, Italy
chiolag@acm.org

G. Cordasco, L. Gargano, A. Negro, and V. Scarano
DIA, Università di Salerno
Via S. Allende, Baronissi (Salerno), Italy.
{cordasco,lg,alberto,vitsca}@dia.unisa.it

Abstract

We define a family of Distributed Hash Table systems whose aim is to combine routing efficiency of the randomized networks — i.e. average path length $O(\log n/\log\log n)$ vs. the $O(\log n)$ average path length of the deterministic system — with the programmability and startup efficiency of a uniform system — that is a system in which the overlay network is transitive, and greedy routing is optimal. It is known that $\Omega(\log n)$ is a lower bound to the average path length for uniform systems with $O(\log n)$ degree.

The proposed family is parameterized with a positive integer c which measures the amount of randomness that is used. Indeed, edges are partitioned into c equivalence classes. *Varying the value c, the system goes from the deterministic case ($c = 1$) to an "almost uniform" system. Increasing c to relatively low values allows routing with optimal average path length while retaining most of the advantages of a uniform system, such as easy programmability and quick bootstrap of the nodes entering the system.*

We also provide a matching lower bound for the average path length of the family of routing schemes for any c. Moreover, we show how to extend the result to other overlay networks.

1 Introduction

Peer–to–Peer (P2P) file sharing applications have recently become very popular. Several of the proposed systems are completely distributed and use a scalable Distributed Hash Table (DHT) as a substrate. A DHT is a self–organizing overlay network that allows to add, delete, and lookup items in a hash table. Several proposals have recently been presented for systems whose hosts configure themselves into a structured network such that lookups require a small number of hops.

The greedy routing approach, in which any message is routed through the neighbor which is nearest to the target, has been used in most of the proposed P2P networks. These include [12, 3, 14, 1, 7]. Several reasons make the greedy strategy popular. In particular, greedy routing is very simple to implement and has some "implicit" fault-tolerance capabilities. It was, however, noticed that greedy routing usually produces paths of length larger than what would be required in a network of the given node degree.

In fact, randomization [6, 8] and de Bruijn graphs [4] were shown to produce networks with optimal average path length. Unfortunately, these algorithms are not greedy and present some other disadvantages as discussed in [8].

1.1 NoN routing on randomized networks: Optimal average path length.

By enlarging the set of nodes where the greedy choice is made, it was recently shown that $\Theta(\log n/\log\log n)$ hops can be obtained (on average) with the *Neighbors–of–Neighbors* (NoN) routing [8]: the greedy choice is made here from among all the nodes which are at most 2 hops distant from the node itself.

Latency can be optimally reduced in several well known topologies [8] (such as, e.g., Chord) provided that randomization is used to establish the neighbors of the nodes, and routing is implemented according to the NoN approach. Hence, the use of randomization inspired by the Small-World idea introduced by Kleinberg [5], together with the NoN routing allows one to keep, to some extent, the advantages of greedy routing

[*]This work was partially supported by the Italian FIRB project "WEB–MINDS" (Wide-scalE, Broadband MIddleware for Network Distributed Services), http://web-minds.consorzio-cini.it/.

while optimizing the latency.

Unfortunately, this result is obtained by trading off programmability and feasibility for efficiency. First of all, in NoN routing a certain amount of overhead is inherent and should not be underestimated. Randomization and NoN routing require the transmission to a node of the list of its neighbors of neighbors. While in [8] it is argued that this can be implemented without extra cost by using keep-alive TCP messages, we believe that because of abstraction requirements, transport protocols should not be tampered with by the application protocol, and that performance predictability of a NoN implementation can be seriously limited by underestimating this overhead in the analysis. Moreover, the randomization of the original network trades the decrement of the expected number of hops for the loss of *uniformity*, which would ensure both easy programmability and quick bootstrap.

1.2 Greedy routing on uniform networks: programmability and quick bootstrap.

Given a network, consider the set of node identifiers ordered on a virtual ring. The network is called *uniform* if, for each pair of nodes u and v, the (clockwise) distance between u and the node pointed by the i-th jump (e.g. outgoing edge) of u is equal to the distance between v and the node pointed by the i-th jump of v [13]. In particular, the Chord ring is uniform: if the identifiers are $0, \ldots, n-1$, x points to y iff $x + z \bmod n$ points to $y + z \bmod n$. A thorough study of uniform systems can be found in [13].

Uniformity is a crucial requirement, since it makes any system a good candidate for real implementations. Besides simplicity in the implementation and an optimal greedy routing algorithm without node congestion [13], it also offers easy and efficient join operations (i.e., a *quick bootstrap*).

The join operation in Chord protocol allows the DHT to support nodes that dynamically enter the system[1]. It is rather costly compared with the other operations. If n nodes are present in the network, it requires $O(\log^2 n)$ hops (since $O(\log n)$ lookup are needed) and the routing algorithm performances degrade gracefully even if only half of the nodes have the correct finger table. In general, on Chord-like systems the number of hops required by a join operation is $O((\text{finger table size})\times(\text{average path length}))$ w.h.p.

[1]The *leave* operation has no cost, with the exception of stabilization costs, where techniques similar to those applied to the join can be fruitfully used.

In practice, one wants to ensure a quick bootstrap to all the nodes entering the network and, in some cases, this is a strict requirement of the application. For example, DHT-based distributed file systems may want to give (rather) efficient access to all the files as soon as a node plugs itself into the system.

In uniform systems, quick bootstrap of a node u can easily be obtained by using the finger table of u's predecessor (that is, the node that precedes u on the ring) as a starting point to build an updated finger table. In fact, the i-th finger of u is efficiently obtained by asking u's predecessor for its i-th finger. Of course, uniformity of the network is crucial in this scenario. In this way, the number of hops becomes $O(\text{finger table size})$ on average, since the update for each finger is computed on a small portion of the ring, where only $O(1)$ nodes are present, on average.

It should be noticed that the use of randomized networks makes this approach unfeasible. This is due to the total lack of relation between the finger tables of any pair of nodes in the network. 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.

1.3 Our results

Our goal is to exploit the improvements provided by randomized networks (with NoN routing) while limiting the drawbacks in system complexity inherent to the randomization and the loss of uniformity of the system. The practical contribution of our approach is to combine ease of programmability, a quick bootstrap for the nodes joining the overlay network, and optimal routing efficiency.

On a very high level, the main idea is to have a mechanism that (randomly) partitions the nodes into a (predefined) number c of classes so that nodes in the same class satisfy the uniformity requirement. Our proposal represents a compromise to ensure the applicability of NoN routing in practice.

H-Chord [2] already removes the communication constraint of NoN information, as it is computed locally based on hash functions. Our new proposal goes further by guaranteeing quick bootstrap as well.

1.4 Related work

The Chord system was introduced in [12] to allow efficient lookup in a DHT. By using logarithmic size routing tables in each node, Chord allows one to find the node of a P2P system that is responsible for a given key in logarithmic n number of routing hops. Adding or removing a node is accomplished at a cost of $O(\log^2 n)$ messages.

Uniform P2P systems were formally defined in [13], where a thorough analysis of their parameters is given. Among other results, the paper shows that greedy routing is optimal for uniform systems and gives bounds on the average path length based on the node degree. In particular, any uniform system with degree $O(\log n)$ has average path length $\Omega(\log n)$.

Recently, some non-greedy routing algorithms were proposed that use De-Bruijn based DHT [10, 4]. The goal is to reach an optimal trade-off between degree and path length and, in particular, to allow routing in $O(\log n/\log\log n)$ with logarithmic degree.

The NoN greedy routing was studied in [8], which showed that it is possible to route greedily in $\Theta(\log n/\log\log n)$ on some classes of randomized networks having logarithmic degree.

1.5 Organization of the paper

Section 2 provides some initial definitions. Section 3 explains the ideas at the basis of this paper by focusing on the Chord system. Sections 3.1 and 3.2 report the routing performance estimates computed using Montecarlo simulation. Section 3.3 provides the lower bound on the diameter and on the average path length. Subsequently, in Section 4 we show how this idea applies to other networks. Finally, in Section 5 we conclude the paper with some final remarks.

2 Preliminary definitions

We consider a set of n nodes lying on a ring of 2^m identifiers. Identifiers are labeled from 0 to 2^m-1 in clockwise order and all the arithmetic operations on the identifiers are done $\bmod 2^m$. Each node x has an m-bit ID, $id(x)$, and is connected to its predecessor $P(x)$ and its successor $S(x)$ on the ring.[2]

- **Chord** [12] Each node x is connected by edges to the nodes $x+2^i$, for each $i=0,\ldots,m-1$.
- **R-Chord** [8] Each node x is connected by edges to the nodes $x+2^i+r(i)$, where $r(i)$ is an integer chosen by x uniformly at random in the interval $[0,2^i)$, for each $i=0,\ldots,m-1$
- **H-Chord** [2] Let $H()$ denote a hash function, that maps an id on the interval $[0,1)$. Each node x is connected by an edge to the node $x+2^i+\lfloor H(x)2^i\rfloor$, for $i=0,\ldots,m-1$.

We assume that each node holds not only its own routing table, but also its neighbors' routing tables. Let $d(x,y)$ be a metric for the nodes in the network.

The NoN–greedy routing algorithm:

1. Assume the message is currently at node $u\neq$target.
2. Let $V=\{v_1,\ldots,v_k\}$ be the neighbors of u. For each $1\leq i\leq k$, let $w_{i1},w_{i2},\ldots,w_{ik}$ be the neighbors of v_i and let $W=V\cup\{w_{ij}$ for all $1\leq i,j\leq k\}$.
3. Among these k^2+k nodes, assume that z is the closest to the target (with respect to metric d): If $z\in V$ (i.e. $z=v_{i'}$) route the message from u to $v_{i'}$ else $z=w_{i'j}$ and route from u to $v_{i'}$.

In section 4 we study Small–World networks like Hypercube and Symphony.

- **Hypercube** Each node x is connected by edges to the nodes y iff the two IDs have Hamming distance 1, i.e., they differ in exactly one bit position. Edges are undirected.
- **R-Hypercube** [8] For each $1,\ldots,m$ node x make a connection with node y_i defined as follows: The top $i-1$ bits of y_i are identical to those of x. The i^{th} bit is flipped. Each of the remaining $m-i$ bits is chosen uniformly at random. Edges are undirected.
- **Symphony*** [8] Let δ s.t. $\ln\delta=\frac{\ln 2^m}{k}$, where k is the node degree[3]. For each $i=1,\ldots,k$, let $I_i=(\delta^{i-1},\delta^i]$ and let ϕ^i denote a probability distribution over the integers in I_i, such that the probability of d is proportional to $\frac{1}{d}$. For $i=1,\ldots k$, an edge is established from node x to $x+a_i$, where a_i is an integer drawn from ϕ^i.

3 Almost uniform NoN routing with H_c-Chord

We propose a DHT system that is as efficient as the randomized version of Chord (i.e. the average path length is $O(\log n/\log\log n)$) and is as easy and efficient (programmability and quick bootstrap to the nodes) as a uniform system.

[2]Note that one has to deal with the case in which only some of all 2^m possible identifiers correspond to a node actually present in the network. Hence, $P(x)$ is the first node before x on the ring, i.e., no node identifier is present in the interval $[P(x)+1,id(x)-1]$; $S(x)$ is the first node following x on the ring, i.e., no node identifier is present in the interval $[id(x)+1,S(x)-1]$. Moreover, throughout this paper when we say that a node x is connected to id, we actually mean that x is connected to the first node equal or following id on the ring, i.e to the node corresponding to the identifier $id'\geq id$ such that no node identifier is present in the interval $[id,id'-1]$.

[3]Symphony* can have arbitrary degree k.

Let c be a given positive integer and H() be a cryptographic hash function (like, e.g., SHA-1 [11]) that maps an id on a sequence of m bits.

Definition 1 ***H_c-Chord:*** *For any x, with $0 \leq x < 2^m$, define*

$$c_x = \left\lfloor \left(\frac{H(x)}{2^m} \right) c \right\rfloor . \qquad (1)$$

Each node x is connected by an edge to the nodes $x + 2^i + \left\lfloor \frac{c_x 2^i}{c} \right\rfloor$, for $i = 0, \ldots, m-1$.

It is easy to see that for each node x, the integer c_x defined in (1) satisfies $c_x \in \{0, \ldots, c-1\}$. We refer to the integer c_x as the class of x and to the integers $0, 1, \ldots, c-1$ as the node classes. We also notice that each node chooses a class with probability $1/c$ independently. Moreover, the difference between the i-th jump of a node belonging to class c_x and the i-th jump of a node belonging to class $c_x + 1$ is at most $\left\lceil \frac{2^i}{c} \right\rceil$.

Lemma 1 *Let $1 < c \leq \log n / \log\log n$. The average path length is $O(\log_c n)$ hops for the NoN Greedy algorithm on H_c-Chord with $n = 2^m$ nodes.*

Proof. Consider a node s that wants to send a message to a node t at distance $d(s,t) = d$. Let p be the unique integer such that $2^p \leq d < 2^{p+1}$. There are two cases to take into consideration.
CASE 1: $p \leq c$. In this case $O(c)$ hops are sufficient to reach the destination, since the distance decreases at least by a factor of $3/4$ for each executed hop. Since $c \leq \log n / \log\log n$ we have that $O(\log_c n)$ hops suffice in this case.
CASE 2: $p > c$. Consider the interval of size $d' = \lceil d/c \rceil$ ending in the destination t.

$$I = (t - d', t].$$

In order to prove the lemma in this case, we first show that with constant probability we can reach the interval I in two hops. Let $s_1, \ldots, s_{c-1}$ denote the first $c-1$ neighbors of s, that is,

$$s_i = s + 2^i + \left\lfloor \frac{c_s 2^i}{c} \right\rfloor ,$$

for $i = 1, \ldots, c-1$; these are the neighbors of s in the interval $[s, s + 2^c)$. Moreover, let $s_0 = s$ and $S' = \{s_0, \ldots, s_{c-1}\}$. Finally, denote by $J_k(s_i)$ the k^{th} neighbor of s_i,

$$J_k(s_i) = s_i + 2^k + \left\lfloor \frac{c_{s_i} 2^k}{c} \right\rfloor .$$

From now on, by NoN of s, we mean the set

$$S = S' \bigcup \{J_k(s_i) \mid 1 \leq i < c, \, 0 \leq k < m\}.$$

We are investigating the probability of S having an outgoing edge entering the interval I, that is,

$$P = Pr\left[\exists\, 0 \leq i < c \text{ and } 0 \leq k < m \text{ s.t. } J_k(s_i) \in I\right].$$

We can prove the following:

Claim 1 *For any node $s_i \in S'$, the probability P' that it has an outgoing edge entering the interval I is at least $1/c$.*

Before proving the claim let's discuss its consequences. Each node has chosen its class and, hence, its neighborhood independently from the others. Therefore, the probability that none of these nodes has an outgoing edge reaching the interval I is

$$1 - P = (1 - P')^{|S'|} \leq \left(1 - \tfrac{1}{c}\right)^c \leq e^{-1}.$$

Hence, the probability that s can reach the interval in two hops is at least $1 - e^{-1}$. Thus, in $O(\log_c n)$ hops, the distance is decreased to $2^{p'}$, where $p' < c$, and we have reduced case two into case one. □

Proof. of Claim 1 Consider a node s_i at distance d_i from t. We are investigating the probability of s_i having an outgoing edge entering the interval I, i.e.,

$$P' = Pr\left[\exists\, 0 \leq k < m \text{ s.t. } J_k(s_i) \in I\right].$$

Let p_i be the unique integer ($\leq p$) such that $2^{p_i} \leq d_i < 2^{p_i+1}$. Two cases may arise:

1. $d_i - d' < 2^{p_i}$. In this case if s_i chooses the class 0 (i.e. $c_{s_i} = 0$) then $J_{p_i}(s_i)$ reaches the interval I (namely, $s_i + d_i - d' < J_{p_i}(s_i) = s_i + 2^{p_i} \leq s_i + d_i = t$). Hence, since each node chooses a class independently with probability $1/c$ we have that s_i reaches the interval I with probability at least $1/c$.
2. $d_i - d' \geq 2^{p_i}$. In this case the only jump that can reach the interval I is the p_i-th. We remember that the distance between the i-th jump of a node belonging to class c_x and the i-th jump of a node belonging to class $c_x + 1$ is at most $\left\lceil \frac{2^i}{c} \right\rceil$. Since the size of I is $d' = \left\lceil \frac{d}{c} \right\rceil$ and $2^{p_i} \leq 2^p \leq d$ we have that $|I| \geq \left\lceil \frac{2^{p_i}}{c} \right\rceil$. Then there is at least one class c' such that $s_i + 2^{p_i} + \left\lfloor \frac{c' 2^{p_i}}{c} \right\rfloor \in I$. Hence since each node independently chooses a class with probability $1/c$ we have that s_i reaches the interval I with probability at least $1/c$. □

We can generalize the results to also hold in a ring where not all nodes are present. Due to Chord constraints the n nodes are uniformly distributed[12].

Theorem 1 *Let $1 < c \leq \log n / \log \log n$, the average path length is $O(\log_c n)$ hops for the NoN Greedy algorithm on H$_c$-Chord in a ring of size 2^m where the number of nodes alive is $n \leq 2^m$.*

Proof. Consider a source that wants to send a message at distance d. Because of Lemma 1, it follows that diminishing the distance to size $2^m/n$ takes $O(\log_c n)$ hops. What is left to prove is that the number of nodes alive in an interval I of size $2^m/n$ is small. The expected number of nodes alive in the interval is

$$\begin{aligned} E[A] &= E\left[\sum_{i=1}^{n} x_i \in I\right] = \sum_{i=1}^{n} E[x_i \in I] \\ &= \sum_{i=1}^{n} Pr[x_i \in I] = \sum_{i=1}^{n} \frac{2^m}{n} \cdot \frac{1}{2^m} = 1. \end{aligned}$$

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 [9]. $\square$

3.1 Performances of H$_c$-Chord

Here, we summarize the results obtained by introducing a limited amount of "randomization" (with a hash function) into Chord: efficiency (as in non-uniform networks), programmability and quick bootstrap (as in uniform networks).

Corollary 1 *Let $c = \log n / \log \log n$; the average path length of the NoN Greedy algorithm on H$_c$-Chord is $O(\log n / \log \log n)$ hops in a ring of size 2^m, where the number of nodes alive is $n \leq 2^m$. Moreover, the number of hops for the completion of the join operation are $O(\log n \log c)$ (w.h.p.).*

Proof. The first part of the corollary is proved because of Theorem 1 when $c = \log n / \log \log n$.

In a uniform system, it suffices to use the finger table of u's predecessor (if u is entering the system) which means that its fingers can be off by at most $O(1)$ distance (on average) with respect to what u needs. In H$_c$-Chord u may need to go back to $O(c)$ nodes (w.h.p.) before it can find a node of the same class as u's. This means that its fingers may be off by $O(c)$ (w.h.p.) with respect to what u needs, and that $O(\log c)$ hops are needed to search for each right finger for u. Therefore, the bootstrap can be performed with $O(\text{(finger table size)} \times \log c)$ hops (w.h.p.). The result follows from the size of the finger table in Chord. $\square$

3.2 Montecarlo Simulation Results

In order to assess the effectiveness of our proposed routing algorithm from a practical point of view as well (in addition to the asymptotic complexity analysis), we used a Montecarlo simulation approach to compare the standard Chord, the H-Chord and the H$_c$-Chord routing performance. The H-Chord system [2] can be seen as the limit as c grows of H$_c$-Chord. H-Chord has been shown to have equivalent performances with the original NoN version presented in [8]. However, since this paper proves that for $c = \log n / \log \log n$ the theoretical bound is already reached, it makes no sense to increase the number of classes beyond $\log n / \log \log n$.

A virtual ring is constructed by randomly generating the IDs associated with the prescribed number of nodes. Then the finger list for each node is constructed according to the chosen distance function. Finally, a large number of uniform, independent, random routing requests are issued to the node with the lowest ID in the ring, and routed through the nodes simulating either the greedy (for Chord) or the NoN (for H- and H$_c$-Chord) routing protocol. The number of hops is counted for each route and statistics are collected. 99% confidence level intervals are estimated in order to ensure the precision of the simulated results. The simulation experiments are repeated for many randomly generated virtual rings, until the 99% confidence intervals drop to below 1% of the point estimates.

Figure 1 reports some of the simulation results we obtained by comparing the standard Chord, the H-Chord, and the H$_c$-Chord routing performance (the latter with $c = 2$ classes).

Both the average number of hops and 90^{th} percentile number of hops (i.e., the minimal number of hops that is never exceeded, with a 90% probability) are reported for each case (the H-Chord and H$_c$-Chord with $c = 2$ have the same 90^{th} percentile curve).

The diagrams clearly confirm the advantage of H-Chord over Chord, not only in terms of average hop count (ranging from an 11% reduction with only 100 nodes, to a 20% reduction with 1,000 nodes, up to a 27% reduction with 500,000 nodes) but also in terms of 90^{th} percentile. Moreover, the diagrams show that for small/medium size rings, H$_c$-Chord with only two classes behaves almost as well as H-Chord (with a less than 2% difference up to 5,000 nodes on average hop count, and identical 90^{th} percentile over the whole range of ring size we took into consideration). As

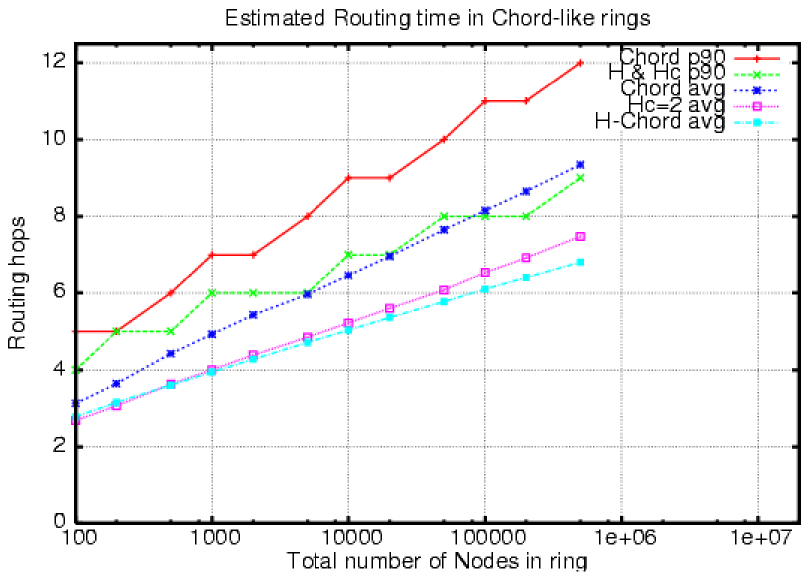

Figure 1: Number of hops for different routing algorithms as a function of the size of the ring: average values and 90 percentile.

ring size increases, the difference between H-Chord and H$_c$-Chord with $c = 2$ classes becomes more evident (3% with 10,000 nodes, 7% with 100,000 nodes, 10% with 500,000 nodes), but still provides a substantial advantage and no substantial disadvantages over Chord from a practical implementation point of view.

3.3 Lower Bound

The bound in Theorem 1 is optimal. In fact, it is possible to prove the following:

Theorem 2 *Both the diameter and the average (shortest) path length of a Small-World network with degree* $\log n$ *and* c *classes are* $\Omega(\log_c n)$.

A comment is needed regarding the choice of c. In fact, this value is, indeed, dynamic (i.e. depending on n) but, because of its slow growth, it reflects the changes in n on the number of classes needed by H$_c$-Chord very slowly. Therefore, standard techniques to estimate the number of nodes can easily be employed to estimate the number of classes that should be used. Moreover, implementation may relax the need to update the number of classes each time a new one must be added (i.e. when the number of nodes is squared), but only for fixed values of c.

4 Further networks with class

We briefly summarize the results for other Small–World networks. Most of the proof is similar to what is presented in the case of the Chord structure, thus for the sake of brevity, it has been omitted.

Definition 2 ***H$_c$-Symphony*:*** *Let* $H()$, c, c_x *and* δ *as defined above, for each* $i = 1, \ldots, m$, x *is connected by an edge to the nodes* $x + \delta^{i - \frac{c_x+1}{c}}$.

Theorem 3 *Let* $\frac{\log n}{k} < c \leq \frac{k}{\log c}$*, the average path length is* $O\left(\frac{\log^2 n}{k \log c}\right)$ *for the NoN greedy routing on H$_c$-Symphony*.*

Definition 3 ***H$_c$-Hypercube:*** *Let* $H()$, c *and* c_x *as defined above, for each* $i = 1, \ldots, m$*, node* x *is connected to the node* y_i *defined as follows: The top* $i-1$ *bits and the last* $m - i - \lfloor \log c \rfloor$ *bits of* y_i *are identical to those of* x*, the* i^{th} *bit is flipped, bits from* $i+1$ *to* $i + \lfloor \log c \rfloor$ *are identical to those of* v_x.

Theorem 4 *Let* $1 < c \leq \log n / \log\log n$*, the average path length is* $O(\log_c n)$ *hops for the NoN Greedy algorithm on H$_c$-Hypercube.*

5 Conclusions

Our Theorems 1 and 2 prove that H$_c$-Chord is a flexible and *optimal* family of routing schemes: when $c = 1$ the network is the original Chord, and by increasing the value of c we can reduce the expected number of hops up to the minimum possible (i.e. $O(\log n / \log\log n)$) that is reached for $c =$

$\log n / \log \log n$. The same improvement can also be obtained on other overlay structures such as Symphony and Hypercube.

Hence, when $c = \log n / \log \log n$ the network becomes as efficient as the randomized version with NoN routing. In a sense, we show that it is not necessary to use a completely randomized network to obtain the optimal value for the average path length. Indeed, the result can be obtained by using a limited amount of randomness, namely only $c = \log n / \log \log n$ classes.

Moreover, since our proposal is based on hash functions (see also [2]), it allows NoN routing to be realistically implemented by avoiding the additional overhead involved in an explicit transmission of the lists of the neighbors of neighbors. The implementation is easy and efficient at the same time, combining the advantages of the NoN routing (that can be implemented as a greedy routing applied to a larger set of nodes) with a quick bootstrap of nodes entering the system (which can easily be obtained thanks to the uniformity among nodes belonging to the same class).

In applications on wireless networks, join/leave operations are frequent and, therefore, quick bootstrap becomes crucial in this setting. Moreover, a node entering the ring can start routing right away based on the approximate finger table provided by its predecessor, even before updating it.

Stochastic simulation results have allowed us to show that the partition of nodes in a very small number of classes provides almost the same reduction of H-Chord in terms of the expected number of hops in case of small/medium size rings, and that even in case of rings with several hundred thousand nodes $c = 2$ classes may provide a substantial advantage compared with standard Chord with very limited overhead in finger table construction.

Finally, notice that, in practice, the values of c that allow us to obtain the optimal average path length for n at most 1 billion is upper limited by a small value, i.e. 5 but, as shown by our simulations, an even smaller value (i.e. 2) is sufficient for reasonable size networks.

Acknowledgements

This work was partially supported by the Italian FIRB project "WEB–MINDS" (Wide-scalE, Broadband MIddleware for Network Distributed Services), `http://web-minds.consorzio-cini.it/`

References

[1] J.Aspnes, G.Shah, "*Skip Graphs*". Proc. of Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, Jan 2003.

[2] G. Cordasco, L.Gargano, M.Hammar, and V. Scarano, "*Degree-Optimal Deterministic Routing for P2P Systems*". Proc. of 10th IEEE Symposium on Computers and Communications (ISCC 2005), La Manga del Mar Menor, Cartagena, Spain, Jun 2005. (see also Proc. of ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC '04) (Brief Announcement), Jul 2004).

[3] P. Druschel, A.Rowstron, "*Pastry: Scalable, distribute object location and routing for large-scale peer-to-peer systems*". Proc. of 18th IFIP/ACM Inter. Conf. on Distr. Sys. Plat., 2001.

[4] M.F. Kaashoek, D.R. Krager, "*Koorde: A simple degree-optimal distributed hash table*". Proc. of International Workshop on Peer-to-Peer Systems (IPTPS'03), Feb 2003.

[5] J. Kleinberg, "*The Small-World phenomenon: An algorithmic prospective*". Proc. of ACM Symposium on Theory of Computing (STOC '00), 2000.

[6] D. Malkhi, M. Naor and Ratajczak, "*Viceroy: A Scalable and Dynamic Emulation of the Butterfly*". In Proc. of ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC '02), Aug 2002.

[7] G.S.Manku, M.Bawa, P.Raghavan, "*Symphony: Distributed hashing in a Small World*". Proc. 4th USENIX Symposium on Internet Technologies and Systems (USITS '03), 2003.

[8] G.S. Manku, M. Naor, U. Wieder, "*Know thy Neighbor's Neighbor: The Power of Lookahead in Randomized P2P Networks*". Proc. of ACM Symposium on Theory of Computing (STOC '04), 2004.

[9] Rajeev Motwani and Prabhakar Raghavan, "Randomized Algorithms", Cambridge Iniversity Press, 1995.

[10] M. Naor and U. Wieder, "Novel architectures for P2P applications: the continous-discrete approach". In Proc. of 15th ACM Symp. on Parallel Algorithms and Architectures, 2003.

[11] National Institute of Standards and Technology, Secure Hash Standard, *http://www.itl.nist.gov/fipspubs/fip180-1.htm.*

[12] 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". IEEE/ACM Transactions on Networking, 2003.

[13] J. Xu, Abhishek Kumar, Xingxing Yu, "On the Fundamental Tradeoffs between Routing Table Size and Network Diameter in Peer-to-Peer Networks". IEEE Journ. on Selected Areas in Comm., vol. 22, no 1, Jan 2004.

[14] B. Y. Zhao, J. Kubiatowicz, and A. Joseph, "Tapestry: An infrastructure for fault-tolerant wide-area location and routing". Tech. Rep. UCB/CSD-01-1141, Univ. of California at Berkeley, Computer Science Department, 2001.