Theory Comput Syst (2009) 45: 43–63
DOI 10.1007/s00224-007-9074-x

Degree-Optimal Routing for P2P Systems

Giovanni Chiola · Gennaro Cordasco · Luisa Gargano · Mikael Hammar · Alberto Negro · Vittorio Scarano

Published online: 25 October 2007

Abstract We define a family of Distributed Hash Table systems whose aim is to combine the routing efficiency of randomized networks—e.g. optimal average path length $O(\log^2 n/\delta \log \delta)$ with δ degree—with the programmability and startup efficiency of a *uniform* overlay—that is, a deterministic system in which the *overlay network* is transitive and *greedy routing* is optimal. It is known that $\Omega(\log n)$ is a lower bound on the average path length for *uniform* overlays with $O(\log n)$ degree (Xu et al., IEEE J. Sel. Areas Commun. 22(1), 151–163, 2004).

Our work is inspired by *neighbor-of-neighbor (NoN)* routing, a recently introduced variation of *greedy routing* that allows us to achieve optimal average path length in randomized networks. The advantage of our proposal is that of allowing the NoN technique to be implemented without adding any overhead to the corresponding deterministic network.

We propose a family of networks parameterized with a positive integer c which measures the amount of randomness that is used. By 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 for routing with asymptotically 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.

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/.

G. Chiola
DISI, Università di Genova, via Dodecaneso 35, 16146 Genova, Italy

G. Cordasco (✉) · L. Gargano · A. Negro · V. Scarano
DIA, Università di Salerno, via Ponte don Melillo, 84084, Fisciano (SA), Italy
e-mail: cordasco@dia.unisa.it

M. Hammar
Research & Development, Apptus Technologies AB, IDEON, 223 70, Lund, Sweden

We also provide a matching lower bound for the average path length of the routing schemes for any c.

Keywords Peer-to-peer · Overlay network · Greedy routing

1 Introduction

Peer-to-peer (P2P) is a class of network applications that takes advantage of existing computing power, computer storage, and networking connectivity, which are available at the edges of the Internet. Hence P2P allows users to leverage their collective power to the benefit of all.

P2P systems have quickly become very popular in the recent years. Several applications exploit P2P networks. Without a doubt, file sharing systems are among the most popular. Sharing other resources, such as file storage and CPU time are also common.

One of the most important benefits provided by P2P systems is the *scalability*. In a P2P system, each consumer of resources also donates resources. Nevertheless, *scalability* has been recognized as the central challenge in designing such systems [18].

To design a scalable system, several P2P systems are based on the distributed hash table (DHT) schema [5, 17, 19, 21]. A DHT is a self-organizing *overlay network* that allows us to add, delete and look up hash table elements. 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 each node uses a routing schema to forward the request for data whose key does not belong to its own key space to the appropriate next-hop. Recently, several systems have been proposed whereby hosts configure themselves into a structured network in such a way that lookups only require a small number of hops.

The efficiency of routing algorithms in P2P systems represents a key factor for ensuring *scalability*. A fast algorithm not only makes the primitive operations (such as accessing or inserting data) efficient, but also impacts positively on maintenance, i.e., by keeping the DHT consistent and in working order in the face of both dynamic and unpredictable join/leave of peers in the network and of node failures.

1.1 Our Results: A Road-Map

In this paper, we propose a novel technique that allows us to retain the improvements provided by the *NoN routing* introduced in [13] over randomized networks, while eliminating the drawbacks in system overhead that is implied by this technique. Moreover, in order to evaluate the effectiveness of the proposed strategy, we apply our technique to two well known networks: Chord [19] and Symphony* [13].

In a sense, our results allow us to retain the programmability and quick bootstrap of uniform networks while leveraging randomization to ensure efficiency. In this section we provide a road-map to our results that can be expressed as obtaining a combination of the ease of management of uniform networks and efficiency of randomized networks.

A preliminary version of some results described in this paper were reported at the 8th International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN 2005) [2] and at the 10th IEEE Symposium on Computers and Communicatiuons (ISCC 2005) [4].

1.1.1 Greedy Routing on Uniform Networks: Programmability and Quick Bootstrap

The traditional measurements of routing efficiency (degree, average path length, etc.) must often be supplemented, in practice, with the consideration of how simple the implementation of the routing algorithms at each node is. In this scenario, the popularity of *uniform networks*, as well as the *greedy routing* approach, is easily explained. In fact, the first proposed DHT systems were *uniform networks* and exploited a *greedy routing* algorithm [5, 17, 19, 21].

Definition 1 (Uniform networks [20]) Consider a network U embedded into a circular metric space (ring) having 2^m identifiers labeled from 0 to $2^m - 1$ in clockwise order. U is *uniform* iff there exist $\delta \geq 1$ integers (a.k.a. jumps) $j_0, j_1, \ldots, j_{\delta-1}$ (where $j_i \in [1, 2^m)$ for each $i = 0, \ldots, \delta - 1$) such that each node $v \in U$ is connected by an edge to the nodes[1] $v + j_i$, for each $i = 0, \ldots, \delta - 1$ (i.e., all the connections are symmetric.)

An extensive study of *uniform systems* was carried out by Xu et al. in [20]. The most important property of *uniform networks* is that the *greedy routing* is optimal. Moreover, 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. if proximity between keys implies proximity between peers) [13]. *Greedy routing* is also very simple to implement and has an implicit fault tolerance capability. Indeed, at each step the *greedy routing* forwards the message to a peer that is closer to the destination. Therefore, if a faulty edge (peer) is found during the routing, the work that has been carried out is not lost. Indeed, each step that is performed independently gets the message closer to the destination. Furthermore, a *greedy routing* on a *uniform network* is peer-congestion-free [20] and ensures a quick bootstrap to all the peers entering the network.

On the down side, *uniform networks* produce paths of length that are larger than what would be required in a network of that given node degree. A typical example is Chord [19] that has degree $O(\log n)$ and in which the *greedy routing* produces an average path length $O(\log n)$, whereas the lower bound is $\Omega(\log n / \log\log n)$. If we limit our attention to *uniform networks*, Xu et al. [20] showed that Chord, as well as the other popular DHT systems like Pastry [5], CAN [17] and Tapestry [21], are indeed optimal.

The join operation in the Chord-like protocol allows the DHT to support nodes that dynamically enter the system. It is rather costly compared with the other operations. The number of hops required by a join operation is $O((\text{finger table size}) \times (\text{average path length}))$ w.h.p., since we need a lookup in order to fill each finger table

[1] All the arithmetic operations on the ring are done mod 2^m.

entry. The *leave* operation is cost-free, with the exception of stabilization costs. The latter can be reduced using techniques similar to those applied to the optimization of the join.

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 networks*, quick bootstrap of a node v can easily be obtained by using the finger table of v's predecessor (i.e., the node that precedes v on the ring) as a starting point. In fact, v and v's predecessor finger tables are quite similar, therefore, the i-th finger of v is efficiently obtained by asking v's predecessor for its i-th finger. Of course, uniformity of the network is crucial in this scenario. Thus, on the average, the number of hops required by a join operation becomes $O(\text{finger table size})$, since the lookup for each finger is computed on a small portion of the ring where only an average of $O(1)$ nodes are present. It must be pointed out that the use of randomized networks makes this approach unfeasible. This is due to the total lack of relationship between the finger tables of any pair of nodes in the network.

1.1.2 NoN Routing on Randomized Networks: Optimal Average Path Length

Recently, a novel approach for routing in DHTs which improves on *greedy routing* was proposed by Manku et al. in [12]. This approach, called *NoN* (Neighbors-of-Neighbors) or 1-*lookahead routing*, substantially consists in making the greedy choice by not only looking at the neighbors of a node, but by looking at all the nodes that are at most two hops away from the node itself.

By using NoN routing, latency can be optimally reduced in several well known topologies (such as Chord) provided that randomization is used to establish the neighbors of the nodes. Hence, the use of randomization together with the *NoN routing* allows us to keep the advantages of *greedy routing* to some extent, while optimizing the latency. For example, while it is known that Chord is not degree–optimal (it uses $\log n$ degree and has an average path length $(1/2)\log n$), [13] highlights that by inserting randomization in the choice of the neighbors of a node and using *NoN routing*, one can, with $\log n$ degree, make the latency (i.e. average path length) drop to $O(\log n/\log\log n)$. It must be pointed out that the *NoN* approach without randomization has proved to be [7] ineffective for Chord, since the average path length is $\Omega(\log n)$.

Unfortunately, the results in [13] are 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 list of its neighbors of neighbors be transmitted to a node. While in [13] Manku et al. argued that this can be implemented without extra cost by using keep-alive TCP messages, we believe that because of abstraction requirements, the application protocol should not tamper with the transport protocols, and that performance predictability of a *NoN* implementation can be seriously limited by underestimating this overhead in the analysis [3, 4]. We substantiate our claim in Sect. 5 by providing a lower bound to the communication cost that is incurred by the *NoN routing* on randomized networks. 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.

For the sake of completeness, it must also be said that several other proposals have been made that give up the *uniformity* constraint in order to improve the average path length of uniform networks. In fact, some research groups (independently) proposed optimal solutions (with non-*greedy routing* algorithms) to build non-*uniform overlay networks* based on the static butterfly networks or the de Bruijn graphs [6, 8, 10, 15]. However, such systems lack on feasibility due to their complex routing algorithms. Besides, Aspnes et al. [1] proposed a new concept of distributed data structure, which allows queries based on key ordering to be performed. Anyhow, the proposed systems, called Skip-graphs, do not address the issue of *load balancing* the number of resources per peer.

1.1.3 Our Proposal: The Best of Both Worlds

Our proposal combines the programmability and startup efficiency of a *uniform* system with the routing efficiency of randomized networks—e.g. optimal average path length $O(\log^2 n/\delta \log \delta)$, where n is the network size and δ is the degree of each node.

After preliminary definitions in Sect. 2, Sect. 3 introduces a family of networks H_c that are parameterized by the actual amount of randomness that is used. By changing the integer parameter c, the network exhibits a behavior that goes from the deterministic case ($c = 1$) to an "*almost uniform*" system.

In Sect. 4 we show that *NoN routing* on H_c networks based on Chord (Sect. 4.1) and Symphony* (Sect. 4.2), allows to obtain the optimal—to within constant factors—average path length for $c = \Omega(\log n/\log \delta)$.

In Sect. 5, we summarize the results of the previous section in order to allow a fair comparison with the traditional *NoN routing* [13], whose communication cost is explicitly bounded from below.

We complete our study by experimentally showing, in Sect. 6, that optimal efficiency can be obtained with relatively low values of c (for example between 2 and 5) for a significant size of the network (around 10^6 nodes). In particular, our approach leads to networks which both maintain the degree and the optimal average path length of the randomized versions of the network, and almost maintain the system overhead of the deterministic version. Our simulations, which were performed both at varying sizes of the network and at varying network loads (i.e. number of nodes with respect to number of available identifiers), validated the theoretical results.

Before the Conclusions section, we also prove a matching lower bound in Sect. 7 for $\delta = \log n$ and for any value of c.

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. All the operations are on the circular metric, e.g. $\bmod 2^m$.

Each node v has an m-bit identifier, and is connected to its predecessor $P(v)$ and its successor $S(v)$ on the ring. In general, one has to deal with the case in which only some of all 2^m possible identifiers correspond to a node that is actually present in the network. Hence, $P(v)$ is the first node before v on the ring, i.e., no node identifier is present in the ring interval $[P(v)+1, v-1]$; $S(v)$ is the first node following v on the ring, i.e., no node identifier is present in the ring interval $[v+1, S(v)-1]$. Moreover, throughout this paper, when we say that a node v is connected to id, we actually mean that v is connected to the first node with key greater than or equal to id on the ring, i.e., to the node corresponding to the identifier id' following id on the ring, such that no node identifier is present in the interval $[id, id'-1]$.

In this paper we will exploit our ideas by applying them to the two well known overlays of Chord and Symphony, whose definitions are reported below:

- Chord: [19] Each node v is connected by edges to the nodes $v+2^i$, for each $i=0,\ldots,m-1$. Notice that Chord is a *uniform network* (cf. Definition 1) where $\delta = m$ and $j_i = 2^i$ for each $i=0,\ldots,\delta-1$.
  R-Chord: [13] Each node v is connected by edges to the nodes $v+2^i+r(i)$, where $r(i)$ is an integer chosen by v at random with uniform probability in the range $[0,2^i)$, for each $i=0,\ldots,m-1$.
- Symphony*: [13] Let γ denote a real number satisfying $\log\gamma = \log\frac{2^m}{\delta}$, where δ is the node degree (Symphony* can have arbitrary degree $\delta \geq 1$). For each $i=1,\ldots,\delta$, let $I_i = [\gamma^{i-1}, \gamma^i)$ and let ϕ^i denote a probability distribution over the integers in I_i, such that the probability of $a \in I_i$ is proportional to $1/a$. For each $i=1,\ldots\delta$, an edge is established from node v to $v+a_i$, where a_i is an integer drawn from ϕ^i.
  We observe that Symphony* can be considered the randomized version of a *uniform network* having jumps $j_i = \gamma^i$ for each $i=0,\ldots,\delta-1$.

2.1 Neighbor of Neighbor Routing

The following definition of the *neighbor of neighbor (NoN) routing* was provided by [13]. We assume that each node holds its neighbors' routing tables as well as its own routing table.

Definition 2 ([13] Neighbor of neighbor routing) Let (V, E) be a graph embedded in a metric space and $d: V^2 \to \mathcal{R}^+$ be a metric for the nodes in the network. Neighbor of neighbor routing entails the following decision (where v is the current node and t is the target key):

1. Let $N(v) = \{u_1, u_2, \ldots, u_\delta\}$ be the neighbors of v.
2. For each $i=1,\ldots,\delta$, let $w_{i1}, w_{i2}, \ldots, w_{i\delta}$ be the neighbors of u_i and let $N^2(v) = \{w_{ij} | 1 \leq i, j \leq \delta\}$.
3. Among these $\delta^2+\delta$ nodes, assume that z is the one closest to the target (with respect to metric $d()$).
4. If $z \in N(v)$ route the message from v *to* z else $z = w_{ij}$, for some i and j, and route the message from v *via* u_i *to* z.

Remark We call the (standard) definition given above, which was analyzed in [13] from a theoretical point of view, 2-*phase NoN*. A more efficient definition, that we call 1-*phase NoN*, can be obtained by replacing Step 4 as follows:

4′. If $z \in N(v)$ route the message from v *to* z else $z = w_{ij}$, for some i and j and route the message from v *to* u_i.

Intuition can easily support the claim of efficiency of *1-phase NoN*: re-applying the whole algorithm at node u_i can only extend the choices. Furthermore, experiments have shown that 2-*phase NoN* is slower than 1-*phase NoN*.

In the *NoN routing* algorithm, u_i might not be the neighbor of v which is closest to the target (with respect to metric d). The algorithm could be viewed as a greedy algorithm on the square of the graph—a message gets routed to the best possible node among those at hop-distance two.

3 H_c-Networks

In this section we introduce the novel family of H_c-networks. We assume that nodes are partitioned into a (predefined) number c of classes so that nodes in the same class can be considered as satisfying the uniformity requirement. Namely, each node randomly chooses one of the classes to belong to; each node in class i, for $i = 0, \ldots, c-1$, chooses its neighbors depending on the parameter i.

Definition 3 (H_c-Network) Let U be a generic *uniform overlay network* (see Definition 1), having n nodes and δ jumps $j_0, j_1, \ldots, j_{\delta-1}$. Let c be any given positive integer in the interval $[1, 2^m]$ and $H()$ be a cryptographic hash function (like, e.g., SHA-1 [16]), that maps an identifier id on the interval $[0, 1)$.

Consider c real numbers $\lambda_0, \lambda_1, \ldots, \lambda_{c-1}$ such that $0 = \lambda_0 < \lambda_1 < \cdots < \lambda_{c-1} < 1$. The network H_c-U is obtained from U as follows: For any $v = 0, \ldots, 2^m - 1$, node v is connected by an edge to the nodes

$$v + \lfloor j_i + \lambda_{c_v}(j_{i+1} - j_i) \rfloor, \tag{1}$$

where $i = 0, \ldots, \delta - 1$, $j_\delta = 2^m$ and $c_v = \lfloor cH(v) \rfloor$.

We refer to the integers $0, 1, \ldots, c-1$ as the node classes and to the integer c_v as the class of v. Easily, c_v satisfies $c_v \in \{0, \ldots, c-1\}$. It is clear that the following property holds.

Property 1 Each node chooses its class independently from among the set $\{0, 1, \ldots, c-1\}$ with probability $1/c$.

We can also observe that an H_c-Network is fully described showing the generating network U and the values λ_ℓ for $\ell = 0, \ldots, c-1$.

In this paper we shall analyze the H_c-Networks that are originated from two significant *overlay networks:* Chord and Symphony*.

- Considering the Chord network and defining $\lambda_\ell = \ell/c$, for each $\ell = 0, \ldots, c-1$, we obtain the H_c-Chord network:

  H_c-Chord: Each node v is connected by an edge to the nodes $v + 2^i + \lfloor \frac{c_v 2^i}{c} \rfloor$, for each $i = 0, \ldots, m-1$.
- We recall that Symphony* can be seen as a randomized version of the deterministic *uniform network* having jumps $j_i = \gamma^i$ for each $i = 0, \ldots, \delta - 1$. By defining $\lambda_\ell = \frac{\gamma^{\ell/c}-1}{\gamma-1}$, for each $\ell = 0, \ldots, c-1$, we obtain the H_c-Symphony* network:

  H_c-Symphony*: Each node v is connected by an edge to the nodes $v + \lfloor \gamma^{i+\frac{c_v}{c}} \rfloor$, for each $i = 0, \ldots, \delta - 1$.

3.1 A Particular Case: H-Networks

By varying the value c, the network H_c-U goes from the underlying *uniform network* U when $c = 1$, i.e., all nodes are in the same class, to a randomized network when $c = 2^m$. In this latter case, each node can choose any value in $[j_i, j_{i+1})$ as its i-th network's jump; we will denote the resulting network by H-U.

In particular, in case of Chord and Symphony we get:

H-Chord: Each node v is connected by an edge to the nodes $v + 2^i + \lfloor H(v) 2^i \rfloor$, for each $i = 0, \ldots, m-1$.
H-Symphony*: Each node v is connected by an edge to the nodes $v + \lfloor \gamma^{i+H(v)} \rfloor$, for each $i = 0, \ldots, \delta - 1$.

We will show that H-Networks are characterized by the fact that by simply using the hash-function defined on the node identifiers, they allow us to preserve the $O(\log n / \log \log n)$ average number of hops for the NoN routing (vs. the need for $\log n$ random numbers needed in the randomized overlay, cfr. R-Chord and Symphony* [13]).

4 Routing in H_c-Networks

4.1 Routing in H_c-Chord

In this section we present the upper bounds we obtained for routing in H_c-Chord networks.

Lemma 1 *Consider $c > 1$. The average path length is $O(\log_c n + \frac{\log n}{\log \log n})$ hops for the NoN routing 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 $2^p \leq d < 2^{p+1}$ and $q = \min\{c, \frac{\log n}{\log \log n}\}$. There are two cases to be taken into consideration.

Case 1: $p \leq q$. In this case, $O(q)$ hops suffice to reach the destination, since the distance decreases at least by a factor of 3/4 for each executed hop (cf. [11]).

Case 2: $p > q$. Consider the interval of size $d' = \lceil d/q \rceil$ 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 $s_1, \ldots, s_{q-1}$ denote the first $q-1$ neighbors of s, that is,

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

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

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

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

$$S = S' \cup \{J_\ell(s_i) \mid 1 \le i < q,\ 0 \le \ell < m\}.$$

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

$$P = \Pr[\exists\ 0 \le i < q \text{ and } 0 \le \ell < m \text{ s.t. } J_\ell(s_i) \in I].$$

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 $\frac{1}{4q}$.*

Before proving the claim let us look at its consequences. Let $\mathcal{S} \subseteq S'$ be a maximal subset of nodes which belong to different classes. Easily, $|\mathcal{S}| \ge \frac{q}{\alpha}$, for some constant $\alpha \ge 1$, with probability not less than $1 - \frac{1}{\alpha^q}$, because $q \le c$. Since nodes in $\mathcal{S}$ belong to different classes, the cases in which one of them has a link in the interval I are independent from one node to another.

Therefore, the probability that none of these nodes has an outgoing edge reaching the interval I is

$$P \ge 1 - (1 - P')^{|\mathcal{S}|} \ge 1 - \left(1 - \frac{1}{4q}\right)^{|\mathcal{S}|} \ge \frac{1}{8\alpha}.$$

Hence, the expected number of nodes encountered before a successful event occurs is $O(1)$. Thus, in $O(\log_q n)$ hops, the distance drops to $2^{p'}$, where $p' < q$, and we have reduced case two to case one.

Accordingly, we need $O(q + \log_q n)$ hops to reach the target t. Hence, one can find the desired value of the average path length by observing that $q = \min\{c, \frac{\log n}{\log \log n}\}$.

We now complete the proof by showing 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.,

$$P' = \Pr[\exists\, 0 \le \ell < m \text{ s.t. } J_\ell(s_i) \in I].$$

Let p_i be the unique integer ($\le p$) such that $2^{p_i} \le d_i < 2^{p_i+1}$, there are two cases to be taken into consideration.

1. $d_i - d' \ge 2^{p_i}$. In this case the only jump that can reach the interval I is the p_i-th. Let C' be the set of classes that allows s_i to reach the interval I (i.e. $\forall c' \in C', s_i + 2^{p_i} + \lfloor \frac{c' 2^{p_i}}{c} \rfloor \in I$). Since each node independently chooses a class with probability $1/c$ we have that s_i reaches the interval I with probability at least $\frac{|C'|}{c}$. Moreover, the distance between the i-th jump of a node belonging to class a and the i-th jump of a node belonging to class $a + r$ is at most $\lceil r\frac{2^i}{c} \rceil$. Hence, $|C'| \ge \lfloor \frac{|I|}{\lceil \frac{2^{p_i}}{c} \rceil} \rfloor$.
   If $q \ge c/4$ we have

$$\frac{|C'|}{c} \ge \frac{\left\lfloor \frac{d'}{\lceil \frac{2^{p_i}}{c} \rceil} \right\rfloor}{c} \ge \frac{\left\lfloor \frac{\lceil \frac{2^p}{q} \rceil}{\lceil \frac{2^p}{c} \rceil} \right\rfloor}{c} \ge \frac{1}{c} \ge \frac{1}{4q}.$$

   Otherwise

$$\frac{|C'|}{c} \ge \frac{\left\lfloor \frac{d'-1}{\frac{2^{p_i}}{c}} \right\rfloor}{c} \ge \frac{\left\lfloor \frac{\frac{2^p}{q}-1}{\frac{2^p}{c}} \right\rfloor}{c} \ge \frac{1}{q} - \frac{1}{2^p} - \frac{1}{c} > \frac{1}{4q}.$$

2. $d_i - d' < 2^{p_i}$. Consider the possibility that $q \ge c/4$. 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, $s_i + d_i - d' < J_{p_i}(s_i) = s_i + 2^{p_i} \le s_i + d_i = t$). 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 $1/c \ge 1/4q$.
   Otherwise, define $A = (d_i - d', 2^{p_i})$, $B = [2^{p_i}, d_i]$, $C = (2d_i - 2d', 2^{p_i+1})$. The probability that one of the hops will reach the interval I is equal to $\Pr[J_{p_i-1}(s_i) \in A$ or $J_{p_i}(s_i) \in B]$.
   We can observe that $J_p(s_i) \in C$ implies that $J_{p-1}(s_i) \in A$.
   Observe that for any $n > 16$, the intervals $(2d_i - 2d', 2^{p_i+1})$ and $[2^{p_i}, d_i]$ do not overlap. Thus, we have:

$$\frac{|C'|}{c} \ge \frac{\left\lfloor \frac{|B|-1}{\frac{2^{p_i}}{c}} \right\rfloor + \left\lfloor \frac{|C|-1}{\frac{2^{p_i}}{c}} \right\rfloor}{c} \ge \frac{\left\lfloor \frac{d'-2}{\frac{2^{p_i}}{c}} \right\rfloor - 1}{c}$$

$$\ge \frac{\frac{\frac{2^p}{q}-2}{\frac{2^p}{c}} - 2}{c} \ge \frac{1}{q} - \frac{2}{2^p} - \frac{2}{c} > \frac{1}{4q}.$$

We can generalize the results so that they also hold in a ring that does not contain all nodes. Due to consistent hashing constraints the n nodes can be assumed to be uniformly distributed [9].

Theorem 1 *Assume $c > 1$, then the average path length is $O(\log_c n + \frac{\log n}{\log\log n})$ hops for the NoN routing algorithm on H_c-Chord in a ring of size 2^m where the number of live nodes 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 $O(\log_c n + \frac{\log n}{\log\log n})$ hops. What is left to prove is that the number of live nodes in an interval I of size $2^m/n$ is small. The expected number of live nodes in the interval is:

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

Furthermore, using the *Chernoff bound*, the probability that more than $1.5 \ln n / \ln\ln n$ nodes lie in an interval I of size $2^m/n$ is less than n^{-2} (see Example 4.4 in [14] for more details). Hence, with probability greater than $1 - 1/n^2$, the number of nodes that lie in the same interval is $O(\log n / \log\log n)$. □

Corollary 1 *The average path length is $O(\frac{\log n}{\log\log n})$ hops for the NoN routing algorithm on* H*-Chord in a ring of size 2^m where the number of live nodes is $n \leq 2^m$.*

4.2 Routing in H_c-Symphony*

In this section we analyze *greedy* and *NoN routing* on H_c-Symphony* networks. The following preliminary result will be a tool in the analysis of the performances of both routing strategies.

Lemma 2 *Let $q > 1$ be an integer and let Φ denote the probability that a node of an H_c-Symphony* network, with $n = 2^m$ nodes (where $1 \leq \delta \leq \log n$) with one hop is able to diminish the distance to a target node from d to, at most, d/q, then $\Phi \geq \frac{\delta}{4q \log n}$ if $c \geq \frac{q \log n}{\delta}$.*

Proof Without loss of generality, assume that the message is at node $v = 0$ and the target is $t = d$, we want to route the message to a node w with $d - w \leq d/q$. in one hop. We denote the smallest integer such that $\gamma^{\ell^*+1} > d$ by ℓ^*. Similarly, we denote the smallest integer such that $\gamma^{\ell^*+\frac{a^*}{c}} > d$ by a^*.

Let I and H respectively denote the intervals $(d(1 - 1/q), d]$ and $[\gamma^{\ell^*+\frac{a^*-1}{c}}, \gamma^{\ell^*+\frac{a^*}{c}}]$. Since $c \geq \frac{q \log n}{\delta}$ we have that $|H| \leq |I|$. Hence, at least one point of the form $\gamma^{\ell^*+\frac{a^*-\alpha}{c}}$ for some integer $\alpha > 0$, belongs to I.

Since each node independently chooses a class with probability $1/c$, we have that each point of the form $\gamma^{\ell^*+\frac{a^*-\alpha}{c}}$ for some integer $\alpha > 0$ is a jump of v with probability $1/c$.

Let $b = \lfloor \frac{c}{q \log \gamma} \rfloor$ (i.e. $bq \log \gamma \leq c < (b+1)q \log \gamma$). Since $c \geq \frac{q \log n}{\delta} = q \log \gamma$, we have that $b \geq 1$. Define $B = \{\gamma^{\ell^* + \frac{a^*-1}{c}}, \gamma^{\ell^* + \frac{a^*-2}{c}}, \ldots, \gamma^{\ell^* + \frac{a^*-b}{c}}\}$. We are able to show that each element of B belongs to I. By contradiction, we assume that $\gamma^{\ell^* + \frac{a^*-b}{c}}$ does not belong to I. We know that $\gamma^{\ell^* + \frac{a^*}{c}} > d$ and since $\gamma^{\ell^* + \frac{a^*-b}{c}} \notin I$, we have that $\gamma^{\ell^* + \frac{a^*-b}{c}} < d(1 - 1/q)$. Hence,

$$\gamma^{\frac{b}{c}} = \frac{\gamma^{\ell^* + \frac{a^*}{c}}}{\gamma^{\ell^* + \frac{a^*-b}{c}}} > \frac{d}{d - d/q} = \frac{q}{q-1}. \tag{2}$$

From (2) we have that $c < \frac{b \log \gamma}{\log(\frac{q}{q-1})}$. Since $q \log(\frac{q}{q-1}) > 1$ then $c < bq \log \gamma$ and we have a contradiction because $c \geq bq \log \gamma$.

Since each element of B could be a jump of v with probability $\frac{1}{c}$, then the probability Φ that a jump of v diminishes the distance from d to at most d/q is

$$\Phi = 1 - \left(1 - \frac{1}{c}\right)^{|B|} \geq \frac{b}{2c} > \frac{b}{2(b+1)q \log \gamma} \geq \frac{\delta}{4q \log n}. \qquad \square$$

Lemma 3 *Consider $c > \frac{2 \log n}{\delta}$, then the average path length is $O(\frac{\log^2 n}{\delta})$ for greedy routing on H_c-Symphony* having $n = 2^m$ nodes, when $1 \leq \delta \leq \log n$.*

Proof Consider node v that holds a message destined for node t at distance d.

Let Φ denote the probability that a v's outgoing link diminishes the distance by at least half, then by using Lemma 2 (with $q = 2$) we have $\Phi \geq \frac{\delta}{8 \log n}$.

Therefore, the expected number of nodes encountered along the route before we come across a distance-halving link is $O(\frac{\log n}{\delta})$. Since the initial distance is d, the maximum number of times the remaining distance could possibly be halved is $\log d$. Since $d < n$ it follows that the average path length is $O(\frac{\log^2 n}{\delta})$. $\square$

Lemma 4 *Assume $c > \frac{\log n}{\log \delta}$, then the average path length is $O(\frac{\log^2 n}{\delta \log \delta})$ for the NoN routing on H_c-Symphony*, with $n = 2^m$ nodes, when $1 < \delta \leq \log n$.*

Proof Consider node v that holds a message destined for node t at distance d and let $q = \frac{\delta}{\log \delta}$. There are two cases to take into consideration.

Case 1: $\log d \leq \frac{q \log n}{\delta}$, then the remaining distance can be covered using *greedy routing* (cf. Lemma 3) in $O(\frac{\log^2 n}{\delta \log \delta})$ hops.

Case 2: $\log d > \frac{q \log n}{\delta}$. Let ψ denote the event that the current node is able to diminish the remaining distance from d to, at most, d/q in (at most) two hops.

Let Ψ denote the probability that event ψ occurs, we will show that

$$\Psi \geq \frac{\delta}{8\alpha \log n}. \tag{3}$$

In order to prove (3), assume $I = (d(1-1/q), d]$. Define Q as the set of nodes that are reachable from v in zero or one hop and that are at distance less than or equal to d from v. We have that $|Q| \geq \log_\gamma d = \frac{\log d}{\log \gamma} > \frac{\frac{q \log n}{\delta}}{\log \gamma} = q$. Let Φ denote the probability that such a node has a link in I. Using Lemma 2 we know that $\Phi \geq \frac{\delta}{4q \log n}$. Let $\mathcal{Q} \subseteq Q$ be a maximal subset of nodes which belongs to different classes. Easily, $|\mathcal{Q}| \geq \frac{q}{\alpha}$, for some constant $\alpha \geq 1$, with probability not less than $1 - \frac{1}{\alpha^q}$, because $q \leq c$. Since nodes in $\mathcal{Q}$ belong to different classes, the cases in which one of them has a link in the interval I are independent from one node to another. Therefore, the probability that none of these nodes has an outgoing edge reaching the interval I is

$$\Psi \geq 1 - (1-\Phi)^{|\mathcal{Q}|} \geq \left(\frac{\delta}{8q \log n}\right)\left(\frac{q}{\alpha}\right) = \frac{\delta}{8\alpha \log n}.$$

Hence (3) holds.

The expected number of nodes encountered before a successful event ψ occurs is at most $O(\frac{\log n}{\delta})$. Since the initial distance is d, the maximum number of times the remaining distance could possibly be diminished is $\log_q d$. Therefore since $d < n$ it follows that the average path length is $O(\frac{\log^2 n}{\delta \log q})$. Thus, in $O(\frac{\log^2 n}{\delta \log q})$ hops, the distance is decreased to d^*, where $\log d^* \leq \frac{q \log n}{\delta}$, and we have reduced case two to case one. Summing the two, the total number of hops is $O(\frac{\log^2 n}{\delta \log q} + \frac{\log^2 n}{\delta \log \delta})$. Hence one can find the desired value of the average path length by observing that $q = \frac{\delta}{\log \delta}$. □

Using the same argument as Theorem 1, i.e., by observing that the number of nodes that lie on the same interval of size $2^m/n$ is small, one can easily prove the following theorem.

Theorem 2 *Assume $c > \frac{\log n}{\log \delta}$, then the average path length is $O(\frac{\log^2 n}{\delta \log \delta})$ (when $1 < \delta \leq \log n$) for the NoN routing on H_c-Symphony* in a ring of size 2^m where the number of live nodes is $n \leq 2^m$.*

5 A Comparison of H_c-Networks with Traditional NoN Routing

Here, we would like to provide a complete comparison of our results with the traditional NoN networks proposed by Manku et al. in [13].

Our results were obtained by introducing a limited amount of "randomization" (with a hash function) into a deterministic network: efficiency (as in non-*uniform* networks), programmability and quick bootstrap (as in *uniform* networks).

Corollary 2 *Assume $1 < \delta \leq \log n$ and $c = \log n / \log \delta$; then the average path length of the NoN routing algorithm on an H_c-Network is $O(\log^2 n / \delta \log \delta)$ hops in a ring of size 2^m, where the number of live nodes is $n \leq 2^m$. Moreover, the number of hops for the completion of the join operation is $O(\delta \times \log \log n)$ (w.h.p.).*

Proof The first part of the corollary is proved because of Theorems 1 and 2 when $c = \log n / \log \delta$.

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

Our results are obtained without the significant maintenance overhead that is incurred by the NoN networks. Without loss of generality, we restrict our discussion to the Chord-like systems, although the same line of arguments are also suited for Symphony* networks.

First of all, note that in order to exploit the *NoN greedy* routing, an R-Chord network uses $\Omega(\log^2 n)$ words of memory, since it keeps track of all the NoN information, whereas the H_c-Chord network only needs a storage of $O(\log n)$.

As explained by Manku et al. [13], R-Chord uses the same network update algorithms as Chord. In addition, the neighbors of neighbors information is communicated during basic network maintenance.

We already commented on the feasibility of this suggestion in Sect. 1. Here, we try to estimate the overhead caused by this communication. The careful reader will have already noticed that H_c-Chord uses the same network maintenance protocol as Chord, i.e., without any extra overhead.

As a matter of fact, the hash-function—exploited by the H_c networks to find the neighbors' neighbors—merely provides us with identifiers. The real neighbors' neighbors are the nodes which immediately follow these identifiers on the ring. This means that the exact neighbors of neighbors information is not available in an H_c network, since the identifier suggested by the hash-function might not correspond to a live node. However, this information suffices to perform efficient lookups: indeed each identifier provided by the hash-function allows us to evaluate the distance to the neighbor in question. Hence, we can use this distance to estimate the search progress that could be made in each step. It follows that by using the estimated distances in Theorem 1 we get an upper bound on the number of hops needed to reach the target.

Below we describe the additional communication costs of R-Chord. Since Manku et al. do not provide exact algorithms, we only give lower bounds on the communication costs and try to comment on the expected overhead in practice.

Theorem 3 *The communication complexity, using R-Chord together with the 1-lookahead protocol for maintaining the network structure is* $\Omega(\log n)$ *messages for each node for and each round if no failures, joins, or leaves occur.*

Proof For each node v_i we need to check whether its neighbors have changed and also whether its neighbors' neighbors have changed. If no changes have taken place

an acknowledgment bit is transmitted from each of the neighbors. Hence, to maintain the ring during a period in which there are no failures or leaves we need to send $\Omega(\log n)$ one bit messages per round. □

In practice, since the probability of neighborhood-updates is high, it is more likely that the protocol directly checks which neighbors of the neighbor have changed by using a $\log n$ bit word. Hence, the communication complexity will be $\Omega(\log^2 n)$ bits per node and round.

Theorem 4 *Every* join, leave *or* failure *incurs a message overhead of* $\Omega(\log^2 n)$ *messages and* $\Omega(\log^3 n)$ *bits*.

Proof We will only prove the theorem in case of a *leave*. Consider a node n_ℓ leaving the system. The leaving node has, on the average, $k = \Omega(\log n)$ incoming edges. Hence, n_ℓ is in the neighborhood list of k nodes, call them $n_1, \ldots, n_k$. Each node n_i has received a new representative n_i' for n_ℓ by the stabilizing algorithm of the Chord system. Due to the topological change, n_i' must communicate its neighbors to n_i. The number of these neighbors is, on the average, $\Omega(\log n)$. Hence, $\Omega(\log n)$ messages of size $\Omega(\log^2 n)$ need to be communicated. Furthermore, each neighbor n_i needs to communicate its updated neighborhood information to all of its own neighbors. Hence, additional $\Omega(\log^2 n)$ messages of size $\Omega(\log n)$ bits each need to be communicated. □

6 Monte Carlo 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 Monte Carlo simulation approach to compare the standard Chord, the H-Chord and the H_c-Chord routing performance. The performance of H-Chord was found to be equivalent to the original NoN version that was presented in [13]. 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 identifiers 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 identifier 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 only $c = 2$ classes).

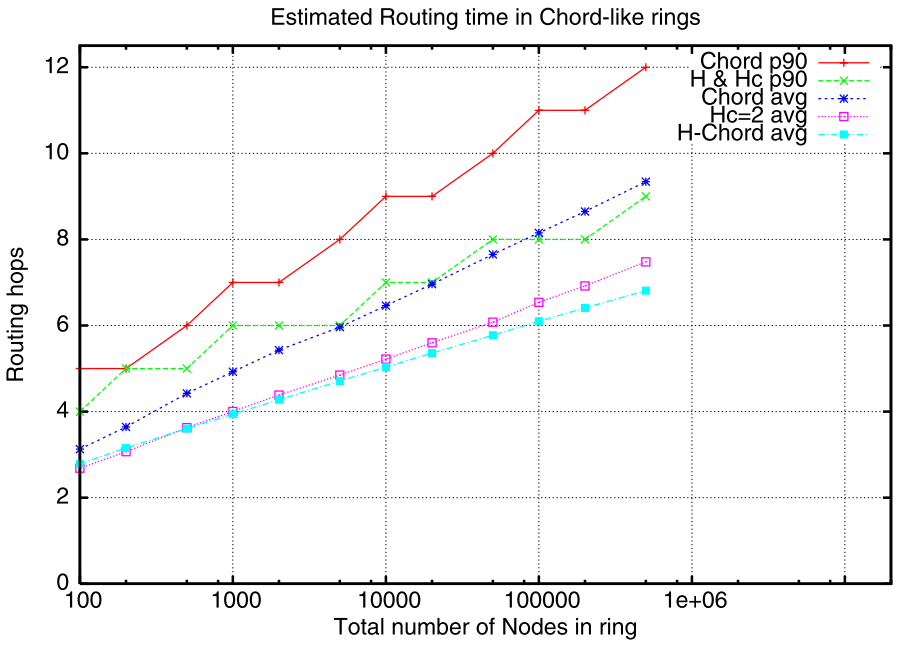

Fig. 1 Monte-Carlo Simulation Results. Number of hops for different routing algorithms as a function of the size of the ring: average values and 90th percentile

Both the average number of hops and 90th percentile number of hops (i.e., the minimal number of hops having a 90% probability of never being exceeded) are reported for each case (the H-Chord and H_c-Chord with $c = 2$ have the same 90th 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 90th 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 90th percentile over the whole range of ring size we took into consideration). As 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 over Chord without substantial disadvantages (in terms of efficiency of the implementation) compared with the uniform version of Chord.

A comment is needed regarding the choice of the number of classes c when the number of peers n varies. Indeed, c is dynamic, but due to its slow growth it very slowly reflects the changes in n. Therefore, standard techniques to estimate the number of nodes, such as considering the distribution of a node's successors [8], can easily be employed to estimate the number of classes that should be used. Moreover, our strategy does not require each node in the network to take on the same value of

c. Accordingly, the value of c can be easily changed on the fly with no harm to the efficiency of the system.

7 Lower Bound

In this section we present a lower bound on the diameter and the average path length of any H_c-Network based on the degree of the node in the network. By degree we mean the number of jumps from each node in the network.

We first prove the following preliminary lemmas.

Lemma 5 *Let $c(n)$ be any function such that $2 \le c(n) \le \log n$. The diameter of any uniform network having degree $O(c(n)\log n)$ is $\Omega(\log_{c(n)} n)$.*

Proof Consider any uniform network U. Since the degree of the network is $O(c(n)\log n)$ then we can assume that the degree of the network (that is, the number of jumps from each node in U) is at most $a \cdot c(n)\log n$, where $a > 2$ and $n > N$ for some $N > 0$. Then choose $n > N$. For the sake of simplicity, since n is fixed, we hide the dependency from n of the various parameters. Hence, denote the degree of the network with n peers by δ and its diameter by d. Also assume a fixed number of classes $c = c(n)$.

In order to prove the lemma, we will show that

$$d \ge \frac{1}{4 \cdot a} \log_c n. \tag{4}$$

Assume, on the contrary that (4) does not hold, that is, $d < \frac{1}{4 \cdot a} \log_c n$ and consider the binomial $\binom{\delta+d}{d}$. By using Stirling's formula[2]

$$\binom{\delta+d}{d} = \frac{(\delta+d)!}{\delta! d!} < \frac{2\sqrt{2\pi(\delta+d)}\left(\frac{(\delta+d)}{e}\right)^{(\delta+d)}}{\sqrt{2\pi\delta}\left(\frac{\delta}{e}\right)^{\delta}\sqrt{2\pi d}\left(\frac{d}{e}\right)^{d}} < \frac{(\delta+d)^{(\delta+d)}}{\delta^{\delta} d^{d}}. \tag{5}$$

By denoting $\alpha = a \cdot c$ and $\beta = \frac{1}{4 \cdot a \log c}$, we can write $\delta \le \alpha \log n,\ d < \beta \log n$. Since $\frac{(\delta+d)^{(\delta+d)}}{\delta^{\delta} d^{d}}$ is an increasing function of both δ and d, we have

$$\binom{\delta+d}{d} < \frac{(\alpha\log n + \beta \log n)^{(\alpha \log n + \beta \log n)}}{(\alpha \log n)^{\alpha \log n}(\beta \log n)^{\beta \log n}} = \left(\left(\frac{\alpha+\beta}{\alpha}\right)^{\alpha}\left(\frac{\alpha+\beta}{\beta}\right)^{\beta}\right)^{\log n}. \tag{6}$$

By noticing that $(\frac{\alpha+\beta}{\alpha})^{\alpha}(\frac{\alpha+\beta}{\beta})^{\beta}$ is an increasing function of β and that $\beta = \frac{1}{4 \cdot a \log c} \le \frac{1}{4 \log \alpha}$, we have

$$\log\left[\left(\frac{\alpha+\beta}{\alpha}\right)^{\alpha}\left(\frac{\alpha+\beta}{\beta}\right)^{\beta}\right] \le \alpha \log\left(1 + \frac{1}{4\alpha \log \alpha}\right) + \frac{1}{4 \log \alpha} \log(1 + 4\alpha \log \alpha)$$

[2] $\sqrt{2\pi x}(x/e)^x < x! < 2\sqrt{2\pi x}(x/e)^x$.

$$\overset{\alpha\geq 4}{<} \frac{1+\log\alpha+\log\log\alpha+\log 5}{4\log\alpha} < 1.$$

Therefore,

$$\left(\frac{\alpha+\beta}{\alpha}\right)^{\alpha}\left(\frac{\alpha+\beta}{\beta}\right)^{\beta} < 2.$$

From (6) we have

$$\binom{\delta+d}{d} < 2^{\log n} = n.$$

However, the network is uniform and, by a result in [20], we know that

$$\binom{\delta+d}{d} \geq n. \tag{7}$$

Informally, the authors in [20] use elementary combinatorics to show that the number of different pathways (in the sense that the paths end up in different destinations) that can be obtained on a uniform network having degree δ and diameter d is equal to $\binom{\delta+d}{d}$. Since each node must be able to reach all other nodes, the inequality (7) holds.

Hence, there is a contradiction and we can conclude that (4) holds. □

Lemma 6 *The average path length of any uniform network having degree $\delta(n)$ and diameter $d(n) = O(\delta(n))$ is $\Omega(d(n))$.*

Proof Let U be a uniform network with n peers and let v be a node in U. Denote by $r(i)$ the number of nodes at distance i from v. By using the same argument as [20] it is easy to show that for each $i = 0, 1, \ldots, d(n)$,

$$r(i) \leq \binom{\delta(n)+i-1}{i}.$$

Since the average path length is $a(n) = \frac{1}{n}\sum_{i=0}^{d(n)} i \cdot r(i)$, we have that the smallest possible average path length is obtained when each $r(i)$ is as large as possible, that is,

$$r(i) = \begin{cases} \binom{\delta(n)+i-1}{i} & \text{if } 1 \leq i < d(n), \\ n - \sum_{i=0}^{d(n)-1} r(i) & \text{if } i = d(n). \end{cases}$$

In such a case, we have that,[3]

$$n \geq \sum_{i=0}^{d(n)-1} r(i) = \sum_{i=0}^{d(n)-1} \binom{\delta(n)+i-1}{i}$$

[3]Remember that, given two positive integers a and b, $\sum_{i=0}^{b}\binom{a+i-1}{i} = \binom{a+b}{b}$ which implies $\sum_{i=0}^{b}\binom{a+i-1}{i} = \frac{a+b}{b}\binom{a+b-1}{b-1} = \frac{a+b}{b}\sum_{i=0}^{b-1}\binom{a+i-1}{i}$.

$$= \frac{\delta(n)+d(n)-1}{d(n)-1} \sum_{i=0}^{d(n)-2} \binom{\delta(n)+i-1}{i}$$

$$= \frac{\delta(n)+d(n)-1}{d(n)-1}[n-(r(d(n)+r(d(n)-1))]. \tag{8}$$

We assume that $d(n) = O(\delta(n))$, that is, $d(n) \leq \alpha \cdot \delta(n)$ for some $\alpha > 0$ and a sufficiently large n. This implies that for sufficiently large n, from (8) we have

$$r(d(n)+r(d(n)-1) \geq \left(n\frac{\delta(n)+d(n)-1}{d(n)-1} - n\right)\left(\frac{d(n)-1}{\delta(n)+d(n)-1}\right)$$

$$= \frac{n\delta(n)}{\delta(n)+d(n)-1} > \frac{n}{\alpha+1}.$$

Hence,

$$a(n) = \frac{\sum_{i=0}^{d(n)} i \cdot r(i)}{n} > \frac{\sum_{i=d(n)-1}^{d(n)} i \cdot r(i)}{n}$$

$$> \frac{(d(n)-1)(r(d(n))+r(d(n)-1))}{n}$$

$$> \frac{(d(n)-1)\frac{n}{\alpha+1}}{n} = \frac{d(n)-1}{\alpha+1} = \Omega(d(n)). \qquad \square$$

Theorem 5 *Both the diameter and the average (shortest) path length of an* H_c*-Network with degree* $O(\log n)$ *and* $c \geq 2$ *classes are* $\Omega(\log_c n + \frac{\log n}{\log\log n})$.

Proof We consider a generic H_c-Network H with n peers, $c(n) \geq 2$ classes and degree $\delta'(n)$. It is known that the lower bound $\Omega(\frac{\log n}{\log\log n})$ holds [20]. We need to show the lower bound $\Omega(\log_c n)$.

Let $\overline{\mathrm{H}}$ be the network obtained by augmenting H in such a way that each node in $\overline{\mathrm{H}}$ maintains $c\delta'(n)$ connections. Namely, each node v has all the $\delta'(n)$ connections $v + \lfloor j_i + \lambda_\ell(j_{i+1} - j_i)\rfloor$ for $i = 0, \ldots, \delta - 1$, corresponding to its membership to class ℓ, for each $\ell = 0, \ldots, c-1$.

We denote by $\delta(n)$ and $d(n)$, respectively, the degree and the diameter of $\overline{\mathrm{H}}$. Obviously $\delta(n) = c\delta'(n)$.

We can observe that: (a) $\overline{\mathrm{H}}$ is a *uniform network* (all the nodes maintain $\delta(n)$ connections of the same length); (b) the diameter and average path length of $\overline{\mathrm{H}}$ is smaller than H ($\overline{\mathrm{H}}$ is in fact obtained by adding some connection to H). Hence, in order to bound the diameter of H (with degree $\delta'(n)$), we can apply the lower bound given in Lemma 5 to $\overline{\mathrm{H}}$ (with degree $\delta(n)$). Moreover, by Lemma 6, we can see that the same bound holds on the average path length of H. $\square$

8 Conclusions

We propose routing schemes that optimize the average number of hops for lookup requests in Peer-to-Peer systems. Unlike other proposed systems, our scheme does not

add any overhead to the system. A recently introduced variation of *greedy routing*, called *NoN routing*, allows us to get optimal average path length with respect to the degree; higher overhead is paid compared to previous systems due to the additional network maintenance [13]. Our proposal has the advantage of "limiting" randomization to such an extent that neighborhood information can be encoded within the hash-value of the node identifier. This enables us to use *NoN* lookup routing without any additional overhead.

Our Theorems 1, 2 and 5 prove that H_c-Networks make up a flexible and *asymptotically optimal* family of routing schemes. When $c = 1$ the network is *uniform*, and by increasing the value of c we can reduce the expected number of hops down to the minimum possible (i.e. $O(\log^2 n/\delta \log \delta)$) that is reached for $c = \log n/\log \delta$.

Implementation is easy and efficient at the same time and combines 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).

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

Acknowledgements The authors would like to thank the anonymous referees whose very helpful comments allowed to significantly improve the presentation of their work.

References

1. Aspnes, J., Shah, G.: Skip graphs. In: Proc. of 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'03), pp. 384–393 (Jan. 2003)
2. Chiola, G., Cordasco, G., Gargano, L., Negro, A., Scarano, V.: Overlay networks with class. In: Proc. of 8th International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN'05), Las Vegas, Nevada, USA, pp. 241–247. IEEE Computer Society (Dec. 2005)
3. Cordasco, G., Gargano, L., Hammar, M., Scarano, V.: Brief announcement: degree-optimal deterministic routing for P2P systems. In: Proc. of 23rd Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC'04) (Brief Announcement), St. John's, Newfoundland, Canada, p. 395. ACM Press, New York (Jul. 2004)
4. Cordasco, G., Gargano, L., Hammar, M., Scarano, V.: Degree-optimal deterministic routing for P2P systems. In: Proc. of 10th IEEE Symposium on Computers and Communications (ISCC '05), La Manga del Mar Menor, Cartagena, Spain, pp. 158–163. IEEE Computer Society (Jun. 2005)
5. Druschel, P., Rowstron, A.: Pastry: scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In: Proc. of the 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware'01), Heidelberg, Germany, pp. 329–350. Springer, New York (Nov. 2001)
6. Fraigniaud, P., Gauron, P.: D2B: a de Bruijn based content-addressable network. Int. J. Theor. Comput. Sci. **355**(1), 65–79 (2006)
7. Ganesan, P., Manku, G.S.: Optimal routing in chord. In: Proc. of 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '04), New Orleans, LA, USA, pp. 176–185. Springer, New York (Jan. 2004)
8. Kaashoek, M.F., Karger, D.R.: Koorde: a simple degree-optimal distributed hash table. In: Proceedings of 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03), Berkeley, CA, USA. Lecture Notes in Computer Science, pp. 98–107 (Feb. 2003)

9. Karger, D.R., Lehman, E., Leighton, F.T., Panigrahy, R., Levine, M.S., Lewin, D.: Consistent hashing and random trees: distributed caching protocols for relieving hot spots on the world wide web. In: Proc. of the 29th Annual ACM Symposium on Theory of Computing (STOC'97), El Paso, TX, USA, pp. 654–663. ACM Press, New York (May 1997)
10. Kumar, A., Merugu, S., Xu, J.(J.), Yu, X.: Ulysses: a robust, low-diameter, low-latency peer-to-peer network. In: Proc. of 11th IEEE International Conference on Network Protocols (ICNP '03), pp. 258–267. IEEE, New York (Nov. 2003)
11. Manku, G.S.: The power of lookahead in small-world routing networks. Technical Report, Computer Science Department, Stanford University, CA, USA (Nov. 2003)
12. Manku, G.S., Bawa, M., Raghavan, P.: Symphony: distributed hashing in a small world. In: Proc. of 4th USENIX Symposium on Internet Technologies and Systems (USITS'03) (Mar. 2003)
13. Manku, G.S., Naor, M., Wieder, U.: Know thy neighbor's neighbor: the power of lookahead in randomized P2P networks. In: Proc. of 36th Annual ACM Symposium on Theory of Computing (STOC '04), Chicago, IL, USA, pp. 54–63 (Jun. 2004)
14. Motwani, R., Raghavan, P.: Randomized Algorithms. Cambridge University Press, Cambridge (1995)
15. Naor, M., Wieder, U.: Novel architectures for P2P applications: the continuous-discrete approach. In: Proc. of 15th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA '03), San Diego, CA, USA, pp. 50–59 (Jun. 2003)
16. National Institute of Standards and Technology, Secure Hash Standard (NIST). http://www.itl.nist.gov/fipspubs/fip180-1.htm
17. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Shenker, S.: A scalable content-addressable network. In: Proc. of ACM Special Interest Group on Data Communication (ACM SIGCOMM'01), San Diego, CA, USA, pp. 161–172 (Aug. 2001)
18. Ratnasamy, S., Shenker, S., Stoica, I.: Routing algorithms for DHTs: some open questions. In: Proc. of 1st International Workshop on Peer-to-Peer Systems (IPTPS'02), Cambridge, CA, USA, pp. 45–52 (Mar. 2002)
19. Stoica, I., Morris, R., Liben-Nowell, D., Karger, D.R., Kaashoek, M.F., Dabek, F., Balakrishnan, H.: Chord: a scalable peer-to-peer lookup protocol for Internet applications. IEEE/ACM Trans. Netw. (TON) **11**(1), 17–32 (Feb. 2003)
20. Xu, J., Kumar, A., Yu, X.: On the fundamental tradeoffs between routing table size and network diameter in peer-to-peer networks. IEEE J. Sel. Areas Commun. **22**(1), 151–163 (Jan. 2004) (A preliminary version appeared in the Proc. of IEEE INFOCOM'03)
21. Zhao, B.Y., Kubiatowicz, J.D., Joseph, A.D.: Tapestry: an infrastructure for fault-tolerant wide-area location and routing. Tech. Report No. UCB/CSD-01-1141, Computer Science Division (EECS), University of California at Berkeley, CA, USA (Apr. 2001)