Degree–Optimal Deterministic Routing for P2P Systems*

G. Cordasco, L. Gargano†, M. Hammar, V. Scarano
Dipartimento di Informatica ed Applicazioni, Università di Salerno, 84081, Baronissi, Italy

Abstract

We propose routing schemes that optimize the average number of hops for lookup requests in Peer–to–Peer (P2P) systems without adding any overhead to the system. 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. Our proposal has the advantage of first "limiting" and then "eliminating" the use of randomization. As a consequence, the NoN technique can be implemented with our schemes without adding any overhead. Analyzed networks include several popular topologies: Chord, Hypercube based networks, Symphony, Skip-Graphs. Theoretical results and extensive simulations show that the proposed simplifications (while maintaining the original node degree) do not increase the average path length of the networks, which is often improved in practice. The improvement is obtained with no harm to the operational efficiency (e.g stability, easy of programming, scalability, fault–tolerance) of the considered systems.

Keywords: Peer–to–Peer, Overlay networks, Greedy routing

Number of pages: 3

Introduction. Peer–to–Peer file sharing applications quickly became very popular in the recent years. Several of the recently 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 look up hash tables. Several proposals have been done recently of systems where hosts configure themselves into a structured network such that lookups require a small number of hops.

The greedy routing approach, in which the message is routed through the neighbor which is nearest to the target, has been used in most of the proposed P2P networks. These include [SML+02, DR01, ZKJ01, AS03, MBR03]. Several reasons make greedy a popular strategy. In particular, one of the main advantages is that greedy routing is very simple to implement and has some "implicit" fault-tolerance capability. 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. As an example some popular topologies like Chord have degree $O(\log n)$ and the greedy routing produces an average path length $O(\log n)$ whereas the lower bound is $\Omega(\log n/\log\log n)$. The use of randomization allowed to exhibit networks with optimal average path length [MNR00]. Recently, constructions based on de Bruijn graphs exhibited optimal trade–offs between degree and latency with deterministic routing; these algorithms are not greedy and present some other disadvantages as discussed in [NW04].

Recently a novel approach for routing in DHTs which improves on greedy routing has been proposed [MBR03, NW04, Man04]. This approach, called *NoN* (Neighbors–of–Neighbors) or *1-lookahead routing*, substantially consists in making the greedy choice by looking not only at the neighbors of a node but at all the nodes at distance at most 2 from the node itself. The NoN approach together with the use of randomization in establishing the neighbors of the nodes which are present in the network, can optimally reduce the latency in several well known topologies [NW04, Man04, MNW04]. Hence the use of randomization, inspired to the Small-world idea introduced by Kleinberg [Kle00], together with the NoN routing allows to maintain, to some extent, the advantages of greedy routing while optimizing the latency. 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 [NW04, Man04] that inserting randomization in the choice of each neighbor of a node and using NoN routing one can, with $\log n$ degree, make the latency (i.e. average path length) drop to $O(\log n/\log\log n)$.

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

†Contact author: Dipartimento di Informatica ed Applicazioni "R.M. Capocelli", Università di Salerno, Via S. Allende, 84081 Baronissi (Salerno) Italy. Phone: +39-089965331, Fax: +39-089965272. E-mail: lg@dia.unisa.it

Our results. Our goal is to retain the improvements given by the NoN routing over randomized networks, while eliminating the drawback in system overhead implied by this technique. In fact, randomization and NoN routing require the transmission to a node of its neighbors of neighbors. While the authors in [NW04, Man04] argue that this can be done without extra cost by using keep-alive TCP messages, we eliminate the extra-communication at all and, similarly, eliminate the need of storing in each node its neighbors of neighbors. To this aim we need to eliminate the random factor in establishing each neighbor of a node. In fact, determinism allows each node to calculate locally the neighbors of its neighbors. Randomness is eliminated through two "simplification" steps. First, we reduce the use of random values as much as possible, so that, in a second step, we can argue that we might as well use any good hash function (such as, e.g., the Secure Hash Algorithm (SHA)) instead of the random number, thus having a completely deterministic network (in which the NoN do not need any more to be transmitted and stored). Indeed hashing can be done on (for example) node's ID so that another node, knowing the ID x, also knows x's neighbors. Analyzed networks include several popular topologies: Chord, Hypercube based networks, Symphony, Skip-graphs. Theoretical results and extensive simulations show that the proposed simplifications (while maintaining the original node degree) do not increase the average path length of the networks, which is often improved in practice. It follows that our approach lead to "deterministic" overlay networks which maintain the original degree and the optimal average path length of the corresponding randomized versions, while improving on the system overhead. Simulations are done both at varying size of the network and at varying network load (i.e. number of nodes with respect to number of available identifiers).

Due to space limits we summarize the theoretical results for the various considered topologies and use Chord as a paradigmatic example for our validating simulations.

Summary of theoretical results. We first show that the average path length is $O(\log n/\log\log n)$ hops for performing look ups on the proposed modified versions of Chord, Symphony[1], and Hypercube based networks. The routing table size is $O(\log n)$.

As in [Man04], we use the R-Chord defined as follows: for each node v in the network, jump i is chosen as $(v+2^i+r(i)) \bmod n$, where $r(i)$ is chosen uniformly at random in $[0, 2^i)$, $i = 0, \ldots, \log n - 1$. 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. Then, we observe that we can use any good hashing function (such as SHA, MD5, etc.) on node IDs to get a deterministic algorithm. We call such a system H-Chord (where H stands for Hashing).

We can generalize our results to hold also in a ring where not all nodes are present. In this case, in fact, node y knows the minimum length of the hops made by x and, therefore, its estimate can only reach further (thus improving the efficiency). By giving a tighter balls and bin argument than that of [SML+02] we can, thus, get the average case upper bound $O(\log n/\log\log n)$ also in this case.

The same technique works for modifying Hypercube based networks (call them H-Hypercube) and Symphony (call it H-Symphony). Summarizing we get the following result.

Theorem 1 *The average path length for performing look ups using the greedy NoN algorithm is $O(\log n/\log\log n)$ hops on H-Chord, H-Hypercube and H-Symphony networks, where n is the number of nodes present in the network. The routing table size is $O(\log n)$ in all cases.*

The modification to obtain H-Skip-graphs is slightly different and deserves some explanation. In a sense, Skip-graphs are already *single-random* by definition, but the randomness cannot be utilized in a lookahead search, since the source needs the random number generated by the destination as a key to do the prefix search. By using a deterministic hash function, the key value of the destination becomes available to the source, and a more efficient search is now possible as also confirmed by simulations.

Summary of experimental results. We report, here, only few of the results of our validating simulations. We ran simulations to compare the performances of the greedy algorithms, the NoN greedy algorithms, and our algorithms. We report the results obtained for the average path length for Chord: *(1)* with a number of ID and nodes up to 2^{18}, and *(2)* with a ring of 2^{32} IDs where the number of nodes varies from 2 to 2^{18}; in *(3)* we show simulation results for average path length for Skip-graphs

[1] Actually Symphony can have arbitrary degree k, in this case the latency is $O(\log^2 n/k\log k)$ hops on average.

with a number of nodes up to 2^{17} (for Skip-graphs there is no bound on size). All simulations have been carried on by using SHA as hash function.

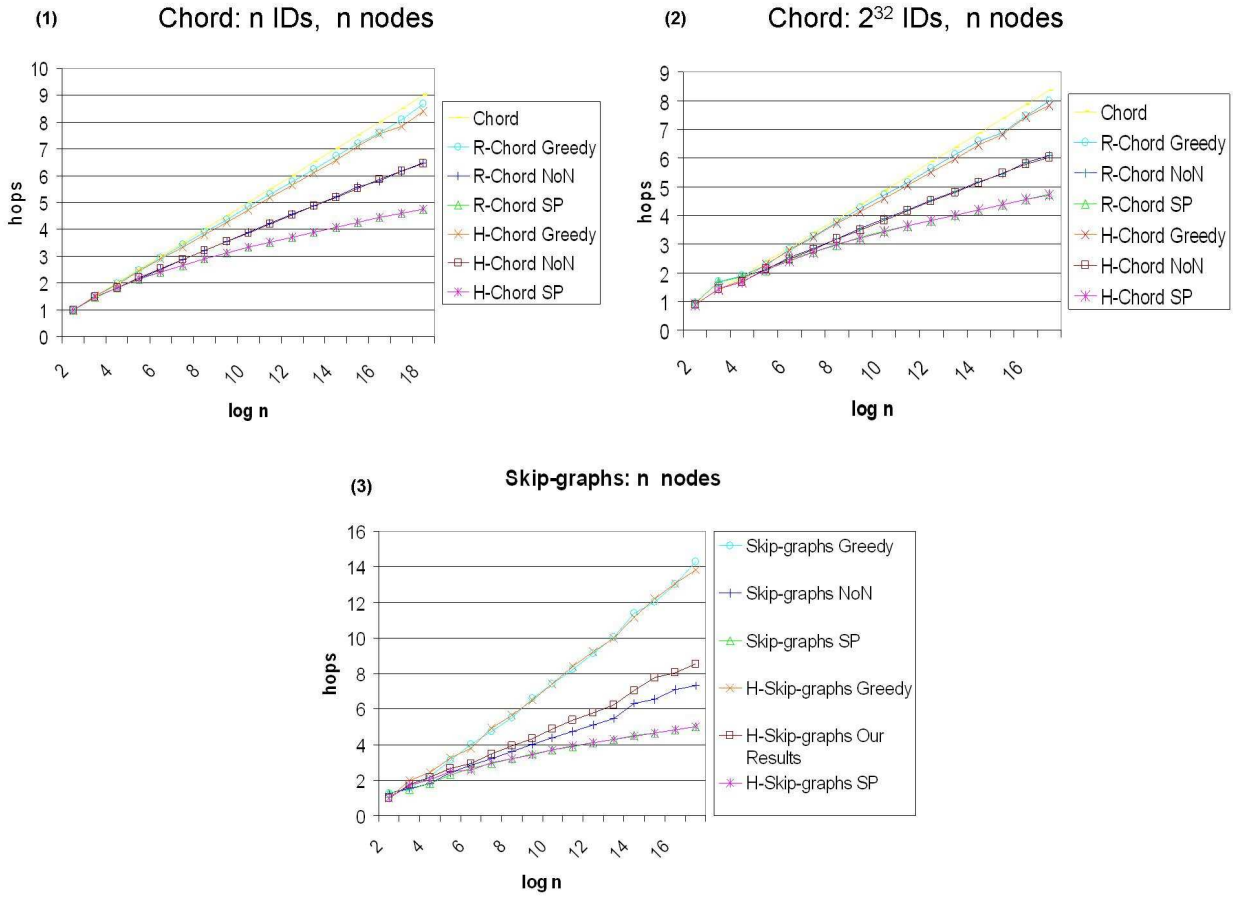

Figure 1: Three routing algorithms are compared here: *SP*, chooses the optimal shortest path between each source and destination; *Greedy* and *NoN*. In (1) and (2) the reader can easily check that R-Chord and our H-Chord have (practically) identical efficiency (average path length) with all the three algorithms. (3) shows that H-Skip-graphs have the same behavior as Skip-graphs when SP and Greedy routing is used, while our routing scheme shows moderate degradation in performances with respect to NoN, but it does not require extra-communication overhead.

References

[AS03] J.Aspnes, G.Shah, "*Skip Graphs*", Proc. of SODA '03.

[DR01] 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. (Middleware '01), Nov 2001.

[KK03] M.F. Kaashoek, D.R. Krager, "*Koorde: A simple degree-optimal distributed hash table*", Proc. IPTPS, Feb '03.

[Kle00] J. Kleinberg, "*The Small-world phenomenon: An algorithmic prospective*", Proc. of STOC '00.

[MNR00] D. Malkhi, M. Naor and Ratajczak, "*Viceroy: A Scalable and Dynamic Emulation of the Butterfly*", Proc. PODC '02, Aug 2002.

[MBR03] G.S.Manku, M.Bawa, P.Raghavan, "*Symphony: Distributed hashing in a Small World*", Proc. of USITS'03.

[MNW04] G.S. Manku, M. Naor, U. Wieder, "*Know thy Neighbor's Neighbor: The Power of Lookahead in Randomized P2P Networks*", Proc. of STOC '04, to appear.

[Man04] G.S. Manku, "*The Power of Lookahead in Small-World Routing Networks*", manuscript, 2004.

[NW04] M.Naor, U.Wieder,"*Know thy Neighbor's Neighbor: Better Routing for Skip-Graphs and Small Worlds*", Proc. of IPTPS '04.

[SML+02] 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 Trans. Networking, '03.

[ZKJ01] 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, '01.