Know thy Neighbor's Neighbor: the Power of Lookahead in Small Worlds and Randomized P2P Networks*†

Gurmeet Singh Manku‡ Moni Naor§¶ Udi Wieder¶

August 2, 2005

Abstract

Several peer-to-peer networks are based upon randomized graph topologies that permit efficient GREEDY routing, e.g., randomized hypercubes, randomized Chord, skip-graphs and constructions based upon small-world networks. In each of these networks, a node has out-degree $O(\log n)$, where n denotes the total number of nodes, and GREEDY routing is known to take $O(\log n)$ hops on average. Our contribution in this paper is twofold. First we investigate the limitations of GREEDY routing and establish lower-bounds for GREEDY routing for these networks. The main contribution of the paper is the analysis of the *Neighbor-of-Neighbor* (NoN)-GREEDY routing. The idea behind NoN, as the name suggests, is to take a neighbor's neighbors into account for making better routing decisions.

The following picture emerges: Deterministic routing networks such as hypercubes and Chord have diameter $\Theta(\log n)$. This means that GREEDY routing is optimal in the sense that its routing distance is at most (approximately) the diameter, yet networks with average degree of $O(\log n)$ may have diameter $O(\frac{\log n}{\log\log n})$. Randomized routing networks such as skip-graphs, randomized hypercubes, randomized Chord, and constructions based upon small-world percolation networks, have diameter $\Theta(\log n/\log\log n)$ with high probability. In all of these networks, GREEDY routing fails to find short routes, requiring $\Omega(\log n)$ hops with high probability. Surprisingly, the NoN-GREEDY routing algorithm is able to diminish route-lengths to $\Theta(\log n/\log\log n)$ hops, which is asymptotically optimal.

*A preliminary version of the paper appeared in [29, 34]

†Research supported in part by the RAND/APX grant from the EU Program IST

‡Google Inc. Research done while at Stanford. manku@cs.stanford.edu

§Incumbent of the Judith Kleeman Professorial Chair.

¶The Weizmann Institute of Science. {moni.naor,udi.wieder}@weizmann.ac.il

1 Introduction

Randomized network constructions that model the *Small-World Phenomenon* have recently received considerable attention. A widely-held belief pertaining to social networks is that any two people in the world are connected via a chain of six acquaintances (*six-degrees of separation*)[1]. The behavior of such networks has been investigated extensively by researchers from diverse set of disciplines including: the social sciences, physics, computer science and webologists. These investigations consist of either checking the existence of the phenomenon in various setting or coming up with models to explain it. The quantitative study of the phenomenon started with Milgram's [31] experiments in 1960's, asking people to send letters to unfamiliar targets only through acquaintances. Milgram's experiments and the work by Pool and Kochen [12] confirmed that often random pairs of individuals are indeed connected by short chains.

The study of the algorithmic or routing perspective of this phenomenon was initiated by Kleinberg [23],[22], who pointed out that the small world experiments showed not only that short paths exist, but that *people can find such paths based on local information*. To model the routing aspects of the Small-World Phenomenon, Kleinberg considered a family of random graphs. The graphs not only have small diameter (to model the "six degrees of separation") but also allow short routes to be discovered on the basis of local information alone (to model Milgram's observation that messages can be "routed to unknown individuals efficiently"). In particular, Kleinberg considered a two dimensional $n \times n$ grid with n^2 nodes. Each node is equipped with a small set of "local" contacts and one "long-range" contact drawn from a harmonic distribution, i.e, the probability of establishing an edge (x, y) is proportional to $||x - y||^{-2}$, where $||x - y||$ stands for the grid's L_1 distance. With GREEDY routing, the path-length between any pair of nodes is $O(\log^2 n)$ hops, w.h.p. Local knowledge available to a node suffices for GREEDY routing – a message is forwarded along that out-going link which takes it *closest* to the destination. Barrière *et al* [7] showed that GREEDY routing requires $\Omega(\log^2 n)$ hops for Kleinberg's construction. The diameter of small world graphs is shorter and is $\Theta(\log n)$ on expectation [30]. Thus, GREEDY routing is sub-optimal and it is desirable to find routing schemes that route along shorter paths.

Kleinberg's results can have various interpretations: it could be thought of as an explanation of how people in the chain letter experiments behaved (this is a descriptive approach). Alternatively, it could be seen as suggesting a routing strategy. While sending letters to unknown targets is not the most useful activity, the problem is related to routing in peer-to-peer networks, i.e. networks where nodes join and leave the system dynamically. In various P2P systems nodes are assigned labels that are interpreted as points on some $d-$dimensional space; links are added to close neighbors and some links are added to far away ones. Hence the hope is that lessons learned for the small world graphs may be applicable in the peer-to-peer environment. Indeed the works of Aspnes *et al* [4] and Manku *et al* [28] apply intuitions from small worlds into peer-to-peer constructions. We shall show further adaptations in this work.

Peer-to-Peer Networks

P2P routing networks have witnessed a flurry of research activity recently. Broadly, the topology of these networks can be classified into two categories – deterministic and randomized. In deterministic

[1]According to Barabási [6] this idea may have its origins in a short story "Chains" by the Hungarian writer Frigyes Karinthy from 1929; this idea has been retold and recast many times since then, in the literature, popular press as well as scientific studies.

P2P networks the topology is a function of the id's of the nodes. Typically they are based upon classical parallel inter-connection networks, such as the hypercube and its variants, butterflies or De-Bruijn graphs, e.g [37, 42, 39, 1]. In randomized networks, as the name suggests, randomization is used to determine the topology of the network. Natural examples include skip-graphs [5, 20], the randomization of hypercubes [17, 9] and the randomization of Chord [41, 17]. All of which have a node degree of $O(\log n)$. Other examples are networks based on Kleinberg's construction such as Symphony [28] [4]. In these networks the out-degree of each node is bounded by a constant. Among the various P2P routing networks, skip-graphs are unique in that node identifiers (or "keys" associated with nodes) can be drawn from an arbitrary ordered domain, e.g., the set of character strings. This property makes skip-graphs the only P2P routing network that naturally supports prefix-*search*. Other P2P routing networks assume that nodes are assigned identifiers that are drawn uniformly from the unit interval $[0, 1)$.

Many P2P networks share structural similarities with a network in which nodes are associated with a $d-$dimensional torus, and an edge (i, j) is established with probability $\frac{1}{||i-j||^d}$, independently of all other edges. We call this network a *small-world percolation network*. The small-world percolation network has its antecedents in classical "long range percolation" models. We outline a brief history at the beginning of Section 2.

The routing scheme suggested for all the above networks is GREEDY; i.e., each node routes the message to its neighbor which is closest to the target. The GREEDY algorithm is appealing to use since it is based on local information only and is very simple (both conceptually and to implement). The main disadvantage of GREEDY is that often it routes along paths that are much longer than the shortest paths in the network. The Neighbor-of-Neighbor (NoN) greedy algorithm is meant to overcome this problem. The idea underlying NoN is to allow a node to gain knowledge of its neighbor's neighbors for assistance in making better routing decisions. . Our work addresses two questions:

(a) When does GREEDY routing route along (approximately) shortest paths?

(b) What is the role of look-ahead (or Neighbor of Neighbor) upon GREEDY routing?

1.1 Our Contributions

In a network with k out-going links per node, the average length of shortest paths is $\Omega(\log n/\log k)$. Therefore, with $O(\log n)$ links per node, it *might* be possible to route in $O(\log n/\log\log n)$ hops, and it *might* be possible to route in $O(\log n)$ hops in Kleinberg's construction. The known upper bounds for GREEDY routing are sub optimal in this sense. The main contribution of this work is to show that in many cases GREEDY routing is indeed asymptotically sub-optimal, while the NoN-Greedy algorithm which uses just one level of look-ahead is asymptotically optimal. In particular we show the following:

Upper bounds: We show that NoN-GREEDY routing, which fixes two hops of a route (by taking the neighbors of neighbors of a node into account), is optimal for the small-world percolation networks and requires $\Theta(\log n/\log\log n)$ hops, w.h.p. (Section 2). The same upper bound is established for randomized-hypercubes and randomized-Chord (Section 3) and for skip graphs (Section 4). Thus skip-graphs are the only degree-optimal P2P network that supports *prefix search*. In

Section 3 we also analyze Kleingberg's construction (Symphony) and show that the NoN algorithm is asymptotically better than GREEDY yet not optimal.

The asymptotical analysis is accompanied by simulations which show that for network sizes ranging from 2^{12} to 2^{24} nodes, NoN-GREEDY routes are 40% to 48% shorter than GREEDY routes in all of these topologies (Section 6).

Lower bounds In Section 5 we show that GREEDY routing requires $\Omega(\log n)$ hops on average in small world percolation graphs and in each of the following randomized P2P networks: skip-graphs, randomized-Chord, randomized-hypercube, and Symphony with $k = \Theta(\log n)$ per node.

1.2 Related Work

The tradeoff between the average path length and the out-degree of nodes is of fundamental interest to designers of P2P routing networks. Hypercubes and Chord offer average paths of length $\Theta(\log n)$ with $\Theta(\log n)$ links per node with GREEDY routing (optimal routes in Chord were identified by Ganesan and Manku [16]). Skip graphs, Randomized-hypercubes and randomized-Chord were known to offer routes of length $O(\log n)$ with GREEDY routing. Among the randomized P2P networks, Viceroy [26] offers routes of length $\Theta(\log n)$ w.h.p. with only $O(1)$ links per node. A randomized construction in [27] combines ideas from Viceroy with Kleinberg's construction to arrive at a network that routes in $\Theta(\log n/\log k)$ hops w.h.p., with k links per node.

Networks based on De-Bruijn graphs [33, 21, 14] offer an optimal tradeoff between degree and path length, in particular for $O(\log n)$ links per node the routes are of length $O(\log n/\log\log n)$. The De-Bruijn networks are significantly simpler than Viceroy and the construction in [27], yet the routing protocol in De-Bruijn graphs is not GREEDY – it is based on numeric computations on labels of nodes. Recently Abraham *et al* [2] presented a graph based on the Butterfly network in which when the degree is d, GREEDY routes along paths of length $O(\log n/\log d)$.

Overall, two classes of networks are known to have optimal route lengths with respect to the degree, for instance route in $\Theta(\log n/\log\log n)$ hops with $\Theta(\log n)$ links per node: De-Bruijn networks and butterfly networks. The P2P implementation of these networks requires that keys are *random*, thus unlike skip-graphs there is no natural way for keys to carry *semantic* meaning. The results of this paper add a third class – "randomized small-world networks". We hope that our results inspire further investigations into the general properties of these networks.

The basic idea of the NoN-GREEDY approach is drawn from two sources. A paper by Coppersmith *et al* [11] uses the neighbors-of-neighbors approach, though not in an algorithmic perspective. They use the idea to establish that the diameter of small-world percolation networks on n nodes is $O(\frac{\log n}{\log\log n})$ w.h.p. NoN-GREEDY *routing* was first used (under the name "GREEDY with 1-LOOKAHEAD") by Manku *et al* [28] as a heuristic for Symphony, a randomized P2P network. Fraigniaud *et al* [15] recently analyzed other variants of GREEDY algorithms in Kleinberg's model, when each node is aware of the long-range contacts of the $\log n$ nodes which are closest to it. They show that a variant of GREEDY which routes in expected $\Theta(\log^{1+\frac{1}{d}} n)$ hops (when d is the dimension of the mesh). Aspnes *et al* [4] established lower bounds for GREEDY over a general family of randomized networks under the assumption that each "long-range" link is drawn from the same probability distribution. Lebhar and Schabanel [25] present a routing algorithm which is not greedy and improves over the simple greedy algorithm.

1.3 The NoN-GREEDY Routing Algorithm

We introduce the main object of our investigation, the NoN-GREEDY Routing Algorithm, in Figure 1. We assume the existence of a metric on the labels of nodes.

Algorithm for routing a message to node t.

1. Assume the message is currently at node $u \neq t$. Let $w_1, w_2, \ldots, w_k$ be the neighbors of u.
2. For each w_i, $1 \leq i \leq k$, find z_i - the closest neighbor to t. Let j be such that z_j is the closest to t among $z_1, z_2, \ldots, z_k$.
3. Route the message from u via w_j to z_j.

Figure 1: The NoN-GREEDY Algorithm. Some metric over the labels of nodes is assumed.

In the NoN-GREEDY algorithm, w_j may not be the neighbor of u which is closest to t. 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 distance two.

2 Small-World Percolation Graphs

Definition 2.1. *A "small-world percolation network" of dimension d is a finite graph whose vertex set is associated with the $d-$dimensional mesh. The probability that (u, v) is an edge is $\frac{1}{||u-v||^d}$ and is independent from all other edges, and $||u - v||$ stands for the mesh L_1 distance between u and v.*

Small-world percolation networks originate from a classical percolation model called "long range percolation". In that model, nodes lie on an infinite grid and an edge is put between a pair of nodes with some positive probability. The question of existence of infinite components was considered by Schulman [38], Aizenman and Newman [3] and Newman and Schulman [35], where the one dimensional grid $\mathcal{Z}$ is studied and edges (i, j) are selected with probability $\beta/\|i - j\|^s$ for some values β, s.

Benjamini and Berger [8] proposed and studied a finite percolation model: a cycle graph over n nodes where an edge between nodes i and j exists with probability 1 if $\|i - j\| = 1$, otherwise, it exists with probability $\exp(-\beta/\|i-j\|^s)$, for some values β, s. Coppersmith *et al* [11] extended the model to multiple dimensions: a $d-$dimensional mesh where an edge (u, v) is selected independently with probability $1/\|u - v\|^d$. Coppersmith *et al* [11] established that the *diameter* of the resulting graph is $\Theta(\log n/ \log\log n)$ w.h.p. Their proof used the neighbor-of-neighbor approach for part of the way, and a non-constructive argument for the rest of the way. We now show that Non-GREEDY routing results in paths of length $\Theta(\log n/ \log\log n)$ w.h.p.

Theorem 2.2. *Given two nodes s, t in a $d-$dimensional small-world percolation network over n nodes, with probability at least $1 - \frac{1}{n^3}$ the NoN-GREEDY algorithm routes a message from s to t in $O(\frac{\log n}{\log\log n})$ hops. The probability is taken over the configuration of the graph.*

Note that the high probability bound of Theorem 2.2 implies that with high probability the NoN algorithm finds short paths between *all* pairs of nodes.

Proof of Theorem 2.2. The L_1 distance between any two nodes is at most n. So we assume the worst case - that the distance between the source and target is n. We partition the routing into two phases. In the first phase, the message is routed so that the remaining distance to the target diminishes to $e^{\sqrt{\log n}}$ or less. In the second phase, the message covers the remaining distance. We show that each phase takes $O(\log n/\log\log n)$ w.h.p., thus proving the theorem. The first phase was handled in Lemma (6.1) from [11].

Lemma 2.3 ([11]). *If $m \geq c\log n/\log\log n$, for some constant c which depends only on the dimension, then after m NoN-*GREEDY *routing steps, the message would reach a node that lies at distance $e^{\sqrt{\log n}}$ or less from the destination, with probability at least $1-\frac{1}{n^3}$.*

The second phase of the routing could in fact be performed by plain GREEDY routing.

Lemma 2.4. *Given that the current location of the message from the source is at distance at most $e^{\sqrt{\log n}}$ from its destination, then with probability at least $1-\frac{1}{n^3}$, the message would reach its destination within $O(\log n/\log\log n)$* GREEDY *steps.*

First we show the effect of a single a NoN hop:

Claim 2.5. *Let δ and δ' denote the distance from the destination before and after performing a single NoN* GREEDY *hop. There is an $\epsilon = \epsilon(d) > 0$ such that for any sequence of hops leading to the current node and for all $k > 0$*

$$\Pr[\delta' \leq \lceil (1-\frac{1}{k})\delta\rceil] \geq 1-\frac{1}{k^\epsilon}.$$

Proof. The proof will show that the Claim holds even for a single *greedy* hop. A single NoN hop is always longer than a single greedy hop. Assume the message is at node $\vec{0}$, and the target note t is such that $||t||_1 = \delta$. For each integer k define B_k to be all nodes with distance at most $(1-\frac{1}{k})\delta$ from t (for notational convenience we remove the ceilings and floors). We calculate the probability there is an edge from $\vec{0}$ to the ball B_k. Define ℓ_i to be the number of vertices x such that $||x|| = i$ and x is in B_k. We have:

$$\Pr[\vec{0} \text{ is not connected to } B_k] = \prod_{i=\delta/k}^{\delta}(1-i^{-d})^{\ell_i} \leq \prod_{i=2\delta/k}^{\delta}(1-i^{-d})^{\ell_i} \leq \prod_{i=2\delta/k}^{\delta} e^{-\ell_i/i^d} = \exp(-\sum_{i=2\delta/k}^{\delta}\frac{\ell_i}{i^d})$$

Now assuming that ℓ_i is $\Theta(i^{d-1})$ for $\frac{2\delta}{k} \leq i \leq \delta$, for some constant ϵ it holds that

$$\exp(-\sum_{i=2\delta/k}^{n}\frac{\ell_i}{i^d}) \leq \exp(-\Theta(\sum_{i=2\delta/k}^{n}\frac{1}{i})) \leq \frac{1}{k^\epsilon}$$

which proves the claim. It remains to show that indeed $\ell_i = \Theta(i^{d-1})$ for $\frac{2\delta}{k} \leq i \leq \delta$. There are $\Theta(i^{d-1})$ nodes at distance i from $\vec{0}$, we need to show that a constant fraction of them are in B_k, i.e with distance at most $(1-\frac{1}{k})\delta$ from t. Let i take some value $2\delta/k \leq i \leq \delta$ and let x be a point on a shortest path from $\vec{0}$ to t such that $||x|| = \lceil\frac{1}{2}(\delta/k+i)\rceil$. Note that $x \in B_k$, furthermore, all the points in distance $i - ||x|| = \lfloor\frac{1}{2}(i-\delta/k)\rfloor$ from x are also in B_k. Note that $\lfloor\frac{1}{2}(i-\delta/k)\rfloor$ is $\Theta(i)$ therefore there are $\Theta(i^{d-1})$ points at distance $\lfloor\frac{1}{2}(i-\delta/k)\rfloor$ from x. How many of them are

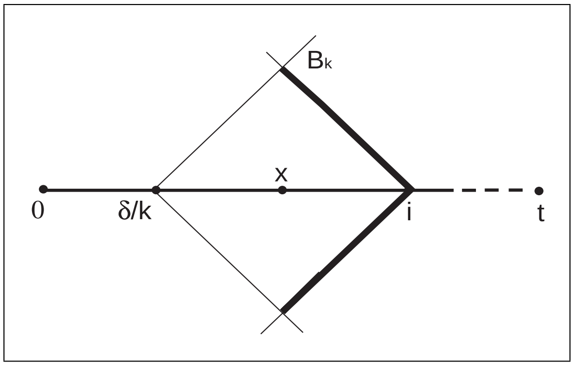

Figure 2: The bold line indicates points which are in B_k and are exactly distance i from $\vec{0}$.

of distance i from $\vec{0}$? All the points of equal distance from x are evenly divided between the 2^d quadrants of the ball around x. It follows that a $2^{-d} = \Theta(1)$ fraction of them are at distance i from $\vec{0}$, which concludes the proof of Claim 2.5. Figure 2 illustrates these calculations. □

Proof of Lemma 2.4. Claim 2.5 analyzed the case of a single NoN hop. Each hop (whether NoN or Greedy) examines edges towards the target, and eventually takes the edge (or two edges) which covers the longest distance. Therefore the portion of the graph that was encountered on previous hops is disjoint from the portion of the graph encountered in the current hop. In other words, the length of each hop is a random variable which depends only on the distance from the target and is independent from previous hops. Therefore we can use Claim 2.5 iteratively: set $k = \log^{1/4} n$, the probability the distance is reduced by a factor of $1 - \frac{1}{(\log n)^{1/4}}$ is $1 - \frac{1}{\log^{\epsilon} n}$. This means that $o(\log n / \log\log n)$ steps, each reduces the distance by $1 - \frac{1}{(\log n)^{1/4}}$, would route the message to the destination. We prove this occurs with probability $1 - \frac{1}{n^3}$ using the following argument: Let X_i be the random Bernoulli variable indicating whether the i^{th} NoN-hop have failed in reducing the distance by a factor of $1 - \frac{1}{(\log n)^{1/4}}$. We know that $\Pr[X_i = 1] \leq \frac{1}{\log^{\epsilon} n}$. Now assume that the variable X_i is simulated by tossing $\epsilon \log\log n$ fair coins and setting $X_i = 1$ if all coins turned up to be 1. Now we have $c \log n$ fair coins, and if less than $\frac{3}{4}$ of the coins turned up to be 1 the algorithm will not fail. The standard Chernoff bound [10] shows there is a constant c such that this happens with probability at least $1 - \frac{1}{n^3}$. □

The proof of Theorem 2.2 is now completed by combining Lemma 2.3 which handled the first phase of the routing, with Lemma 2.4 which handled the second phase of the routing. □

Do People Use the NoN-GREEDY Algorithm in Social Networks?

Since the original motivation of analyzing small-world graphs was the modeling of social networks, it is interesting to check whether people use the NoN-greedy algorithm when they navigate in a social network. Recently Dodds *et al* [13] repeated the famous experiment of Milgram [31] in which letters were passed between random nodes on a social network where edges corresponds to say, an acquaintance known by first name. In the Dodds *et al* experiment participants were given a target and were asked to forward an email to some person they were acquainted with. The goal of forwarding was to ensure that the email would reach its destination quickly. The participants were also asked to explain *why* they chose the person from among their set of acquaintances. It appears that in the first two steps of the "routing", which are most meaningful, about 25% of the people sent the message to a recipient for one of the following reasons:

1. The recipient was known to have traveled to the target's geographical region.
2. The recipient's family was known to have originated from the target's geographical region.

Both reasons suggest that the recipient received the message based on who his/her (possible) acquaintances were, and not on the individual characteristics of just the recipient. Other reasons, such as – "the recipient has the same education as the target" – could be viewed both as greedy and NoN-GREEDY steps. We can conclude that at least some of the time, the NoN-GREEDY algorithm was used.

3 Small-World P2P Networks

In this section, we analyze NoN-GREEDY routing for various randomized P2P routing networks which are related to the small world model and the small world percolation model discussed in the previous Section. Skip Graphs, which are of a different flavor, are analyzed in Section 4. We begin by defining these networks formally. For each of the following we assume there are $n = 2^\ell$ nodes arranged on a circle.

- **Randomized-Hypercube** [9, 17]: The out-degree of each node is ℓ. For each $1 \le i \le \ell$, node $\mathbf{x}$ makes a connection with node $\mathbf{y}$ defined as follows: The top $i-1$ bits of $\mathbf{y}$ are identical to those of $\mathbf{x}$. The i^{th} bit is flipped. Each of the remaining $\ell - i$ bits is chosen uniformly at random. Edges are directed. Out-degree is $\ell = \log n$.
- **Randomized-Chord** [41, 17]: Node $\mathbf{x}$ makes ℓ connections as follows: Let $r(i)$ denote an integer chosen uniformly at random from the interval $[0, 2^i)$. Then for each $0 \le i < \ell$, node $\mathbf{x}$ creates an edge with node $(\mathbf{x} + 2^i + r(i)) \bmod n$. Edges are directed. Each node has out-degree $\ell = \log n$.
- **Symphony** [28, 4]: Node $\mathbf{x}$ establishes a *short-distance* edge with node $(\mathbf{x}+1) \bmod n$. Node $\mathbf{x}$ also establishes $k \ge 1$ *long-distance* edges as follows: For each edge, node $\mathbf{x}$ first draws a random number r from the probability distribution $p(x) = 1/(x \ln n)$ where $x \in [1, n]$ and then establishes a link with node $\lceil \mathbf{x} + r \rceil \bmod n$. Edges are directed. The resulting graph is thus a multi-graph node $\mathbf{x}$ could be connected to $\mathbf{y}$ by more than one edge. The out-degree of each node is $k+1$.

Symphony with $k = 1$ is identical to Kleinberg's construction [23] in one dimension. Randomized-Hypercube and randomized-Chord are structurally similar to small-world percolation networks with $d = 1$ (see Definition 2.1). An important distinction is that the out-degree for each of the P2P routing networks is fixed.

Some easy adaptations of Lemma 2.3 and 2.4 could be used to prove the following theorem:

Theorem 3.1. *Given two nodes s, t in a randomized-Chord or a randomized-Hypercube network over n nodes, with probability at least $1 - \frac{1}{n^3}$ the NoN-GREEDY algorithm routes a message from s to t in $O(\frac{\log n}{\log \log n})$ hops, (the probability is taken over the configuration of the graph).*

The Theorem implies that these P2P networks are degree optimal using NoN. In Section 5 we show that GREEDY routing takes $\Omega(\log n)$, thus GREEDY makes suboptimal routing decisions.

In Symphony with out-degree $k+1$ the expected path length found by GREEDY is $\Theta(\frac{\log^2 n}{k})$. The following Theorem shows that NoN-GREEDY improves upon GREEDY.

Theorem 3.2. *The expected number of hops taken by NoN-GREEDY to route between any two nodes in Symphony is $O\left(\frac{\log^2 n}{k \log k}\right)$, when $1 \le k \le \log n$ and the expectation is over the formation of the graph.*

Theorem 3.2 means that NoN improves upon GREEDY for any degree. When the degree is $\log n$ then NoN improves such that it is degree optimal w.h.p. Martel and Nguyen show that the *diameter* of symphony for any constant k is $O(\log n)$, which means that for these values of k NoN does not route along approximately shortest paths.

Proof. Consider node $\mathbf{x}$ that holds a message destined for node $\mathbf{y}$ lying clockwise distance d away. It is proven in [28] that GREEDY routing takes $O(\frac{\log n \log d}{k})$ hops. Therefore, if $\log d \le \log n / \log k$, then the remaining distance can be covered by NoN (which is faster than plain GREEDY) in $O(\log^2 n/(k \log k))$ hops.

We now consider large d satisfying $\frac{\log n}{\log k} < \log d \le \log n$. Let $r(d) = \frac{ck \log d}{\log n}$ where d is the clockwise distance currently remaining and c is a constant that we will shortly fix. Since $\frac{\log n}{\log k} < d \le \log n$, we deduce that $\frac{ck}{\log k} < r(d) \le ck$.

Lemma 3.3. *Let $\mathcal{E}$ denote the event that the current node is able to diminish the remaining distance from d to at most $\frac{d}{r(d)}$ in (at most) two hops, then $\Pr[\mathcal{E}]$ is $\Omega(\frac{k}{\log n})$, independent of d.*

Thus the expected number of nodes encountered before event $\mathcal{E}$ occurs is $O(\frac{\log n}{k})$. Since $\frac{ck}{\log k} < r(d)$, there can be at most $O(\frac{\log n}{\log k})$ such events for a total of $O(\frac{\log^2 n}{(k \log k)})$ hops. When d becomes small enough to satisfy $\log d < \frac{\log n}{\log k}$, plain GREEDY routing will take at most $O(\frac{\log^2 n}{k \log k})$ hops. Summing the two, the total number of hops is $O(\frac{\log^2 n}{k \log k})$. Thus it only remains to prove Lemma 3.3.

Proof that $\Pr[\mathcal{E}]$ is $\Omega(\frac{k}{\log n})$: Denote by $B(\mathbf{x})$ the number of nodes connected by an edge to $\mathbf{x}$ which are at a clockwise distance of at most d away. By the definition of Symphony it holds that $\mathbb{E}[B(\mathbf{x})] \ge \frac{k \log d}{\log n}$. Let $d' = \lceil d(1 - \frac{1}{r(d)}) \rceil$. Let ψ denote the event that such a node has a link in clockwise distance $[d', d]$ from $\mathbf{x}$. Since ψ is independent from $B(\mathbf{x})$ and $\Pr[\mathcal{E}]$ is monotone in $B(\mathbf{x})$ we have that overall, the probability that one or more of these nodes has a link in distance $[d', d]$ from $\mathbf{x}$ is:

$$\Pr[\mathcal{E}] \ge 1 - (1 - \Pr[\psi])^{\frac{k \log d}{2 \log n}} \cdot \Pr\left[B(\mathbf{x}) \ge \frac{k \log d}{2 \log n}\right] \tag{1}$$

We handle each element in (1) separately. $B(\mathbf{x})$ is the sum of k Bernoulli variables, each with success probability $\frac{\log d}{\log n} \ge \frac{1}{k}$. By Chernoff's bound (e.g [18]) we have:

$$\Pr\left[B(\mathbf{x}) \le \frac{k \log d}{2 \log n}\right] \le \exp(-\frac{1}{8}\mathbb{E}^2[B(\mathbf{x})]) \le e^{-\frac{1}{8}}$$

Next we show that $\Pr[\psi]$ is $\Omega(\frac{k}{r(d) \log n})$. Recall that ψ denotes the probability that node $\mathbf{x}$ or one of its neighbors has a link in distance $[d', d]$ where $d' = \lceil d(1 - \frac{1}{r(d)}) \rceil$. According the definition of Symphony, the probability node $\mathbf{x}$ or one of its neighbors does not have a link in distance $[d', d]$ is

$$\Pr[\overline{\psi}] \le \left(1 - \frac{\log(\frac{r(d)}{r(d)-1})}{\log n}\right)^k \le \left(1 - \frac{\log(1+\frac{1}{r(d)})}{\log n}\right)^k \le \left(1 - \frac{1}{2r(d)\log n}\right)^k \le e^{-\frac{k}{2r(d)\log n}}$$

In the third inequality we used the fact that $\log(1+\frac{1}{r(d)}) \le \frac{1}{2r(d)}$. We have:

$$\Pr[\psi] \ge 1 - e^{-\frac{k}{2r(d)\log n}} \ge \frac{c'k}{r(d)\log n}$$

for some constant c'. Now all that remains is to substitute in (1). We had defined $r(d) = \frac{ck\log d}{\log n}$. We set $c \ge c'$ which ensures that

$$\frac{c'k}{r(d)\log n} \cdot \frac{k\log d}{\log n} \le \frac{k}{\log n} \le 1.$$

Using the fact that $1-(1-x)^t \ge xt/2$ if $x \in (0,1)$ and $xt \le 1$, we deduce that

$$\Pr[\mathcal{E}] \ge \frac{c'k^2\log d}{2r(d)\log^2 n}.$$

Substituting $r(d) = \frac{c'k\log d}{\log n}$, we get $\Pr[\mathcal{E}]$ is $\Omega(\frac{k}{\log n})$ as needed. This completes the proof of Lemma 3.3 and thus of Theorem 3.2. □

4 Skip Graphs

In this section, we analyze NoN-greedy routing in skip-graphs [5] and SkipNets [20], which adapt skip-lists [36] for creating a randomized P2P routing network[2]. We follow the description in [5].

4.1 A Brief Review of Skip Graphs

In a skip graph, each node represents a resource to be searched. Node x holds two fields: the first is a key, which is arbitrary and may be the resource name. Nodes are ordered according to their keys. We assume for notational convenience that the keys are the integers $1, 2, \ldots, n$; as the keys have no function in the construction other than to provide an ordering and a target for searches there is no loss of generality. The second field is a membership vector $m(x)$ which is for convenience treated as an infinite string of random bits chosen independently by each node; in practice, it is enough to generate an $O(\log n)$-bit prefix of this string with overwhelming probability.

The nodes are ordered lexicographically by their keys in a circular doubly-linked list S_ϵ so that node i is connected to $i-1 \mod n$ and $i+1 \mod n$. For each finite bit-vector σ, an additional circular doubly-linked list S_σ is constructed by taking all nodes whose membership vectors have σ as a prefix, and linking adjacent nodes in the lexicographic key order. More formally, let $m(x)_k$ be

[2]There are some differences between the two suggestions, but essentially they are the same, and our results apply for both.

the restriction of $m(x)$ to its first k bits; then nodes $x < y$ are connected by an edge if there exists some k such that $m(x)_k = m(y)_k$, and either (a) $m(z)_k \neq m(x)_k$ for each z between x and y, or (b) $m(z)_k \neq m(x)_k$ for all $z > x$ and all $z < y$. In such case we say the edge (x, y) *corresponds* to a prefix of length k. Note that the cycle edges could be seen as corresponding to the empty prefix.

In analyzing a skip graph as a graph, we treat each pair of links as a single undirected edge, and take the union of the resulting edge sets for all lists S_σ.

The main merit of skip-graphs is the following: edges do not depend on the keys themselves but rather on their ordering and the random vectors. Thus the keys may be arbitrary and can carry *semantic meaning*. Furthermore, since the nodes are ordered by their keys, the skip-graph data structure supports *prefix search*. This is in stark contrast with the other networks we discuss, which require that keys be random.

Claim 4.1. *Let x, y be two nodes such that $x < y$. The probability there exists a (clockwise) edge (y, x) is $\Theta(\frac{1}{y-x})$.*

Proof. Denote by $y \sim x$ the event that (y, x) is an edge. The probability (x, y) is an edge which corresponds to a prefix of length k is $2^{-k} \cdot (1 - 2^{-k})^{|x-y|-1}$. Setting $k = \lfloor \log(y - x) \rfloor$ we have $\Pr[y \sim x] \in \Omega(\frac{1}{y-x})$. On the other hand:

$$\Pr[y \sim x] \leq \sum_k 2^{-k} \cdot (1 - 2^{-k})^{(y-x)-1} \in O\left(\frac{1}{y - x}\right)$$

□

The claim above implies that the expected degree of each node is about $\log n$. It is easy to see that w.h.p. the maximum degree in the graph is logarithmic: With high probability all prefixes of length $3 \log n$ are different, therefore all the edges in the graph correspond to prefixes of length at most $3 \log n$.

4.2 Routing in Skip Graphs

The routing algorithm suggested in [5], [20] routes the message using the longest prefix of $m(x)$ possible, without overshooting the target. In other words it is a greedy algorithm which moves the message as close to the target as possible without overshooting, and routes in $O(\log n)$ hops on expectation[3]. We improve this routing by showing in Theorem 4.2 that the NoN-Greedy algorithm routes in $O(\log n / \log \log n)$ hops w.h.p, and by showing in Theorem 5.1 that the Greedy algorithm needs $\Omega(\log n)$ time to route. Each NoN hop considers paths of two edges and routes the message as close as possible to the target without overshooting it. That is, assuming the target node is 0, at each hop the algorithm routes the message to the neighbor of neighbor with the smallest positive id[4]. Since the target node is set to 0 we abuse notation slightly and let a node's key indicate its distance from the target.

Theorem 4.2. *Let s be some node in a skip graph of n nodes, with probability $1 - \frac{1}{n^2}$ the NoN algorithm finds a path between node s and node 0 of length $O(\frac{\log n}{\log \log n})$, where the probability is taken over the choices of membership vectors.*

[3] A high probability bound can be easily derived using the machinery introduced in this section.

[4] For notational convenience we assume that the routing is always done in the clockwise direction.

Given the low probability of failure, Theorem 4.2 implies that with high probability *all* nodes are connected to node 0 via a short path:

Corollary 4.3. *With probability at least* $1 - \frac{1}{n}$ *the diameter of a skip-graph with* n *nodes is* $O(\frac{\log n}{\log\log n})$*, where the probability is taken over the choices of membership vectors.*

The proof of Theorem 4.2 is rather technical, though the outline is similar to that of percolation small world graphs. The proof is divided into two parts. In the first we show that with high probability the message reaches quickly a distance of $\exp(\sqrt{\log n})$ from the target and in the second part we show that with high probability all nodes of distance at most $\exp(\sqrt{\log n})$ are connected to the target through short chains. We need to re-prove Lemmas 2.3 and 2.4 and deal with the dependencies created by the properties of skip graphs.

4.2.1 The First Phase of the Routing

Let X_m be the point the NoN algorithm reached after performing m NoN hops. Since the target is 0, X_m also represents the distance from the target. $X_0 = s$ is the starting point. The following is a restatement of Lemma 2.3 for the case of skip graphs.

Lemma 4.4. *There exists a constant* c *such that for* $m \geq c(\frac{\log n}{\log\log n})$ *with probability at least* $1 - \frac{1}{n^2}$ *it holds that* $X_m \leq \exp(\sqrt{\log n})$*.*

The general outline of the proof follows that of Lemma 2.3 in [11]. Our goal is to show that each NoN hop the distance to the target reduces by some poly-logarithmic factor. The main source of technical difficulty in skip graphs is that the probability of a hop of a certain length depends upon all the hops taken so far. In other words the value $\Pr[X_r \leq \frac{X_{r-1}}{(\log n)^{1/4}}]$ is now a *random variable* whose distribution depends upon r and the membership vectors sampled in the segment $[s, X_{r-1}]$. Fix some $1 \leq r \leq m$. It is sufficient to prove the following lemma:

Lemma 4.5. *Denote by* $\mathcal{E}_r$ *the event that* $\Pr[X_r \leq \frac{X_{r-1}}{(\log n)^{1/4}}] \geq 1 - \frac{c}{\sqrt{\log n}}$*. There exists a constant* c *such that* $Pr[\mathcal{E}_r] \geq 1 - \frac{1}{n^3}$

Before proving Lemma 4.5 lets see why it derives Lemma 4.4. With probability $1 - \frac{1}{n^3}$ the event $\mathcal{E}_r$ holds, so with probability greater than $1 - \frac{1}{n^2}$ the event $\mathcal{E}_r$ holds for every r, i.e. all through the path. Call a NoN hop *successful* if it reduces the distance to 0 by a factor of $(\log n)^{-1/4}$. It holds that $\frac{4\log n}{\log\log n}$ successful hops suffice to bring the message to a distance of $\exp(\sqrt{\log n})$. The previous discussion implies that with high probability each NoN hop along the path is not successful with probability at most $c\log^{-1/2} n$. Therefore we can simulate the process in the following way: for each hop we toss $\frac{1}{2}\log\log n + \log c$ *fair* coins, and the hop fails if *all* of them turn out successful. As seen in the proof of Lemma 2.4, for large enough m, with probability $1 - \frac{1}{n^2}$ there would be $\frac{4\log n}{\log\log n}$ successful NoN hops within the first $\frac{m\log n}{\log\log n}$ attempts.

Proof of Lemma 4.5. Our goal is to use the same outline as in [11]: Assume the current node is x (i.e. $X_{r-1} = x$). The proof has two components. First we show that with sufficiently high probability x has $\Omega(\log x)$ neighbors in $[x-1, 0]$, then we show that with sufficiently high probability *one* of these neighbors is part of a successful NoN step.

Before we do that however we need to build the machinery that will help us deal with the dependencies. The main problem is that conditioning on the path taken so far, the membership vectors in the segment $[x-1,0]$ are not drawn from the uniform distribution. To see this consider for instance the case that for some $y \in [x-1,0]$ it holds that $m(y)_k = m(s)_k$ for large k, then the edge (s,y) would exist and the NoN path starting at s would bypass x altogether. In other words the conditioning that $X_r = x$ affects the distribution of membership vectors in the segment $[x-1,0]$. We conclude that for every choice of membership vectors in the segment $[n,x]$ there is a set of vectors $\mathcal{F}_x \subseteq \{0,1\}^*$ such that the following two conditions hold:

i If a node $y \in [x-1,0]$ has $m(y) \in \mathcal{F}_x$ then there would be an edge from some node in $[n,x+1]$ to y that would cause the NoN path from s to 0 to bypass node x.

ii If all nodes in $[x-1,0]$ have their membership vectors drawn from $\{0,1\}^* \setminus \mathcal{F}_x$ then node x would belong to the NoN path starting from s.

It holds therefore that when conditioning on the path taken so far, the membership vectors in $[x-1,0]$ are drawn *uniformly* and *independently* from $\{0,1\}^* \setminus \mathcal{F}_x$.

Denote by $\mu(\mathcal{F}_x)$ the measure of $\mathcal{F}_x$, i.e. the probability a random membership vector falls within $\mathcal{F}_x$. Note that every choice of membership vectors for the nodes $[n,x]$ defines a path to x and a set $\mathcal{F}_x$, so $\mu(\mathcal{F}_x)$ is a random variable determined by the membership vectors in $[n,x]$. Our goal now is to show that with high probability $\mu(\mathcal{F}_x)$ is small. For every $0 \le \alpha \le 1$ we have:

$$\Pr[\mu(\mathcal{F}_x) \ge \alpha \mid X_{r-1} = x] \le \frac{\Pr[X_{r-1} = x \mid \mu(\mathcal{F}_x) \ge \alpha]}{\Pr[X_{r-1} = x]}.$$

For $\{X_{r-1} = x\}$ to occur all vectors in $[0, x-1]$ must be outside $\mathcal{F}_x$, therefore for every $0 \le \alpha \le 1$ it holds:

$$\Pr[X_{r-1} = x \,|\, \mu(\mathcal{F}_x) \ge \alpha] \le (1-\alpha)^x$$

Let $S_{r-1} \subseteq [0,n]$ denote the set of nodes such that $\Pr[X_{r-1} = x] \ge \frac{1}{n^3}$. The event that $x \notin S_{r-1}$ is negligible. For every $x \in S_{r-1}$ we have:

$$\Pr[\mu(\mathcal{F}_x) > \alpha \mid X_{r-1} = x] \le n^3(1-\alpha)^x \tag{2}$$

Setting $\alpha = \frac{4\log n}{x}$ we have that with probability $\ge 1 - \frac{1}{n^3}$, it holds that $\mu(\mathcal{F}_{X_{r-1}}) \le \frac{4\log n}{x}$. Denote by A the high probability event that $x \in S_{r-1}$ and $\mu(\mathcal{F}_{X_{r-1}}) \le \frac{4\log n}{x}$.

Now all that remains is to show that the occurrence of A implies the occurrence of $\mathcal{E}_r$. First we show that the occurrence of A implies that the distribution of membership vectors in $[x-1,0]$ is almost uniform.

For a prefix $\psi \in \{0,1\}^k$ denote by b_ψ the probability a random and uniform vector in $\{0,1\}^*$ falls in $\mathcal{F}_x$ conditioned on its prefix being ψ (i.e. if u is sampled uniformly from $\{0,1\}^*$ then $b_\psi = \Pr[u \in \mathcal{F}_x | m(u)_k = \psi]$). Denote by w_ψ the probability a random vector in $\{0,1\}^* \setminus \mathcal{F}_x$ has ψ as a prefix, i.e. w_ψ is the probability ψ is a prefix in the conditional distribution of vectors in $[x-1,0]$.

Claim 4.6. *For every integer $k > 0$*

$$\sum_{\psi \in \{0,1\}^k} b_\psi = \mu(\mathcal{F}_x) \cdot 2^k \tag{3}$$

$$\frac{1-b_\psi}{2^k} \le w_\psi \le \frac{1}{2^k(1-\mu(\mathcal{F}_x))} \tag{4}$$

Proof. Let u be a vector sampled uniformly in $\{0,1\}^*$. Equation (3) follows since:

$$\mu(\mathcal{F}_x) = \Pr[u \in \mathcal{F}_x] = \sum_{\psi \in \{0,1\}^k} \Pr[u \in \mathcal{F}_x \mid m(u)_k = \psi] \Pr[m(u)_k = \psi] = 2^{-k} \sum_{\psi \in \{0,1\}^k} b_\psi$$

The first inequality in Equation (4) follows since $\frac{1-b_\psi}{2^k}$ is the probability a single sample from $\{0,1\}^*$ has ψ as a prefix and falls outside $\mathcal{F}_x$. The second inequality holds since $\frac{1}{2^k(1-\mu(\mathcal{F}))}$ is the normalized probability after removing a measure $\mu(\mathcal{F}_x)$. □

We are now set to prove the claims needed for the proof of Lemma 4.5.

Lemma 4.7. *Let $B(x)$ be the number of nodes in $[0, x-1]$ which are connected to x by an edge corresponding to a prefix of length at most $\frac{1}{10}\log x$.*

$$\Pr\left[B(x) \geq \frac{1}{100}\log x \,\middle|\, A\right] \geq 1 - \frac{1}{\sqrt{\log n}}$$

Proof. First note that for every $k \leq \log(\frac{1}{\mu(\mathcal{F}_x)}) - 2$ Claim 4.6 implies that $\sum b_\psi \leq \frac{1}{4}$ and therefore for every $\psi \in \{0,1\}^k$ it holds that

$$\frac{3}{4} \cdot \frac{1}{2^k} \leq w_\psi \leq \frac{5}{4} \cdot \frac{1}{2^k}$$

Next we claim that with high probability there is a series of nodes $y_1 > y_2 > \ldots > y_{1/10 \log x}$ where y_k is the largest node in $[x-1, 0]$ such that $m(y_k)_k = m(x)_k$ for $1 \leq k \leq \frac{1}{10}\log x$. We toss the membership vectors in $[x-1, 0]$ one by one from $m(x-1)$ to $m(0)$ finding the y_i's one by one. In other words - consider a series of independent geometric random variables $g_1, g_2, \ldots$ where the parameter of g_i is the probability a vector shares a prefix of length i with $m(x)$. Since we conditioned on $\mu(\mathcal{F})$ being small, by the previous discussion the success probability of g_i is at least $\frac{3}{4}2^{-i}$. Each g_i is repeatedly tossed until there is a success, in which case g_{i+1} is tossed and so on. On expectation, the number of attempts needed until $g_{1/10 \log x}$ succeeds is at most $\sum_{i=1}^{1/10 \log x} 2^{i+1} \leq 4x^{1/10}$. We have x tosses at our disposal so by Markov's inequality, with probability greater than $1 - 4x^{-\frac{9}{10}}$ there is a series of nodes $y_1, y_2, \ldots, y_{\frac{1}{10}\log x}$ such that $m(x)_k = m(y_k)_k$. Typically it is not the case that all these nodes are neighbors of x. If for some $j < i$ it holds that $m(x)_i = m(y_j)_i$ then y_i is not a neighbor of x. It holds however that

$$\Pr[m(y)_{i+1} = m(x)_{i+1} | m(y)_i = m(x)_i] \leq \frac{\frac{3}{4}2^{-(i+1)}}{\frac{5}{4}2^{-i}} = \frac{5}{6}$$

It follows then that each y_i is a neighbor of x with probability at least $\frac{1}{6}$ and independently from all other y_i's. So on expectation at most $\frac{1}{6}$ of the y_i's are neighbors of x. By Chernoff's bound, the probability less than $\frac{1}{10}$ of the y_i's are connected to x is at most $e^{-\epsilon \log x}$, where ϵ is some constant. Now, since $\log x \geq \sqrt{\log n}$ we conclude that with probability greater than $1 - e^{\epsilon\sqrt{\log n}} > 1 - \frac{1}{\sqrt{\log n}}$ it holds that $B(x) \geq \frac{1}{100}\log x$. □

We continue with the proof of Lemma 4.5. recall that $y_1, y_2, \ldots, y_m$ are neighbors of x where $m \in \Omega(\log x)$ and each y_i corresponds to a prefix of length at most $\frac{1}{10}\log x$. It remains to show that with probability $1 - \frac{1}{\sqrt{\log n}}$ one of them has an edge towards $[\frac{x}{\log^{1/4} n}, 0]$. Consider the first $\log x$ bits of each $m(y_i)$.

Claim 4.8. *Let z be some node in $[x-1,0]$. $\Pr[y_i \sim z \mid A] \geq \frac{1}{40x}$ where the edge corresponds to a prefix of length $\lfloor \log x \rfloor$.*

Proof. Let $L \subset \{0,1\}^{\log x}$ be vectors with $m(y_i)_i$ as a prefix, so $|L| \geq x^{9/10}$. Now, according to Claim 4.6 there are at least $x^{9/10} - 4\log^2 n$ prefixes $\psi \in L$ such that $\frac{1}{2x} \leq w_\psi \leq \frac{2}{x}$. Denote by y_ψ the probability that $m(y_i)_{\log x} = \psi$. The same arguments show that for the vast majority of vectors in L it holds that $\frac{1}{2x^{9/10}} \leq y_\psi \leq \frac{2}{x^{9/10}}$. So to conclude, in at least half of the prefixes in L the conditioning on A changes the probability by a factor of at most 2. Now:

$$\Pr[y_i \sim z \mid A] \geq \sum_{\psi \in L} y_\psi w_\psi (1-w_\psi)^x \geq \frac{1}{40x}$$

□

Claim 4.8 implies that for every node $z \in [\frac{x}{\log^{1/4} n}, 0]$, the probability (y_i, z) is an edge corresponding to a prefix of length $\lfloor \log x \rfloor$ is $\Theta(1/x)$. Let Y_i be the random variable indicating that y_i is connected to some node in$[\frac{x}{\log^{1/4} n}, 0]$ with an edge which corresponds to a prefix of length *exactly* $\lfloor \log x \rfloor$. There could be at most one such edge so

$$\Pr[Y_i] \geq \frac{1}{40 \log^{1/4} n}$$

For $i \neq j$ it holds that $\Pr[Y_i | Y_j] \leq \Pr[Y_i]$. The reason is that if Y_j holds then $m(y_j)_{\log x}$ appears in $[\frac{x}{\log^{1/4} n}, 0]$ and does not appear in $[x, \frac{x}{\log^{1/4} n}]$, both events reduce the probability of Y_i, and this holds also when conditioning on A. In other words, the $Y_i|A$ are negatively correlated, so:

$$\Pr[\overline{Y_1} \wedge \ldots \wedge \overline{Y_m} \mid A] \leq \prod_{i=1}^{\Theta(\log x)} \left(1 - \Theta(\log^{-1/4} n)\right) \leq \exp(-\Theta(\log^{1/4} n)) \leq \frac{1}{\sqrt{\log n}}$$

where in the second inequality we used the assumption that $\log x \geq \sqrt{\log n}$. Adding all the error probabilities implies that given the high probability event A, the event $\mathcal{E}_r$ holds, and as seen, Lemma 4.5 implies Lemma 4.4.

□

4.2.2 Routing the Remaining Distance

For the second phase of the routing we basically re-prove Lemma 4.4 with different number crunching. Assume the Skip Graph contains $e^{\sqrt{n}}$ nodes and as before denote by X_i the location of the NoN algorithm after i steps.

Lemma 4.9. *There exists a constant c such that for $m \geq c(\frac{\log n}{\log \log n})$ with probability at least $1 - \frac{1}{n^2}$ it holds that $X_m \leq c(\frac{\log n}{\log \log n})$.*

Proof. We use the same notation and mechanism of Lemma 4.4. Assume $X_{r-1} = x$ is the current node. Note that $x-1$ is always a neighbor of x, so it is enough to show that a greedy choice from $x-1$ would reduce the distance by a factor of $1 - \frac{1}{\log^{1/4} n}$ with probability $1 - \frac{1}{\log^{\epsilon} n}$ for some $\epsilon \geq 0$.

Recall that Equation (2) states that $\Pr[\mu(\mathcal{F}_x) > \alpha \mid X_{r-1} = x] \leq n^3(1-\alpha)^x$. The Equation implies that for every δ there exist $c(\delta)$ such If we assume that $x \geq \frac{c \log n}{\log\log n}$ then $\Pr[\mu(\mathcal{F}_x) > 1 - \log^{-\delta} n \mid X_{r-1} = x] \leq \frac{1}{n^3}$. The exact value of δ would be set later on. As before we name this high probability event A and condition on it to hold.

Denote by $k = 0.8 \log x$. If k is not integer then take a coefficient slightly smaller than 0.8. As before, for every prefix ψ of length k denote by w_ψ the probability that ψ is sampled condition on the path taken so far. Now:

$$\begin{aligned}
\Pr\left[X_r \leq (1 - \frac{1}{\log^{1/4} n})x \mid A\right] &\geq \Pr\left[x - 1 \sim y \text{ for some } y \leq (1 - \frac{1}{\log^{1/4} n})x\right] \\
&\geq \sum_{\psi \in \{0,1\}^k} w_\psi (1 - w_\psi)^{\frac{x}{\log^{1/4} n}} \left(1 - (1 - w_\psi)^{(1 - \frac{1}{\log^{1/4} n})}\right) \\
&= \sum_{\psi \in \{0,1\}^k} w_\psi (1 - w_\psi)^{\frac{x}{\log^{1/4} n}} - \sum_{\psi \in \{0,1\}^k} w_\psi (1 - w_\psi)^x \qquad (5)
\end{aligned}$$

Our goal is to show that the sum in Equation (5) is at least $1 - \frac{1}{\log^\epsilon n}$. Claim 4.6 implies that if A occurs then for every $\psi \in \{0,1\}^k$ it holds that $w_\psi \leq \frac{\log^\delta n}{x^{0.8}}$. We deal with the two sums of Equation (5) separately:

$$\begin{aligned}
\sum_{\psi \in \{0,1\}^k} w_\psi (1 - w_\psi)^{\frac{x}{\log^{1/4} n}} &\geq \min\{(1 - w_\psi)^{\frac{x}{\log^{1/4} n}}\} \\
&\geq \left(1 - \frac{\log^\delta n}{x^{0.8}}\right)^{\frac{x}{\log^{1/4} n}} \\
&\geq 1 - \frac{1}{\log^{1/25} n} \quad \text{for sufficiently small } \delta
\end{aligned}$$

The first inequality holds since $\sum w_\psi = 1$. In the last inequality we used the assumption that $x > \frac{\log n}{\log\log n}$.

For the second part of Equation (5) we divide the sum into elements of small and big weight. Denote by S all the elements $\psi \in \{0,1\}^k$ such that $w_\psi \leq \frac{1}{x^{0.9}}$. There are at most $x^{0.8}$ elements in S therefore

$$\sum_{\psi \in S} w_\psi (1 - w_\psi)^x \leq \frac{x^{0.8}}{x^{0.9}} \leq \frac{1}{\log^{1/25} n}.$$

On the other hand

$$\sum_{\psi \notin S} w_\psi (1 - w_\psi)^x \leq \max_{\psi \notin S}\{e^{-w_\psi x}\} \leq e^{-\frac{x}{x^{0.9}}} \leq \frac{1}{\log^{1/25} n}.$$

Now as before we have that $O(\frac{\log}{\log\log n})$ distance reductions of factor $1 - \frac{1}{\log^{1/4} n}$ are enough to bring the path to a distance of $O(\frac{\log}{\log\log n})$ from the target, and with probability greater than $1 - \frac{1}{n^2}$ this indeed happens within $O(\log n / \log\log n)$ attempts.

□

The remaining distance to the target could be covered by the cycle edges. Now the proof of Theorem 4.2 is concluded by union bounding the error probabilities of Lemmata 4.4 and 4.9.

5 Lower Bounds

In this section we prove that in order to find a path between nodes at distance n, a routing algorithm must either run in $\Omega(\log n)$ time w.h.p (i.e. $\Omega(\log n)$ hops), or must use additional knowledge about the neighbor's neighbors of a node. The lower bound holds for a model which generalizes the GREEDY algorithm, thus it applies for a larger family of algorithms which includes GREEDY. It holds both for small-world percolation networks and skip graphs.

A logarithmic lower bound of $\Omega(\log^2 n)$ for GREEDY routing in Kleinberg's construction [23] in one dimension was proved by Barrière *et al* [7]. Aspnes *et al* [4] extended the result to a larger family of random graphs. They show that if the average degree is $O(\log n)$ then GREEDY routing would take $\Omega(\log n)$ hops on average. The proof however is limited to the case where the nodes are set on a one dimensional line and the probability upon the edges has some symmetry assumptions that do not apply to skip graphs. We show lower bounds for small-world percolation networks and skip-graphs. Randomized-Chord, randomized-hypercube and Symphony are quite similar to small-world percolation networks, and the proofs could be adapted for each of them.

5.1 A Probing Model

Assume that our goal is to find a path between two specific vertices distance n apart, say node 0 and node n. In order to do so, an algorithm must *probe* the vertices of the graph, where the probing of a vertex reveals all the edges connected to it. Our lower bounds apply in a *probing* model, where we bound the number of probes needed to find a path. Clearly, a lower bound on the number of probes needed by the algorithm is a lower bound on the (sequential) time complexity of a routing algorithm.

We define a $1-$local algorithm to be a probing algorithm with the following properties:

1. The algorithm begins by probing the node 0.
2. The algorithm only probes nodes to which it has already established a path from 0.

The term *local* derives from the assumption that the algorithm starts at 0 and is only allowed to probe nodes it has already reached. The term $1-$local is used, since the probing of a node reveals its neighborhood of radius 1, i.e. its neighbors. If it is assumed that a probe reveals a neighborhood of radius k then the algorithm is termed *k-local*. Every routing algorithm which relies on local information only, corresponds to a $1-$local probing algorithm. The GREEDY algorithm therefore is $1-$local. The NoN-GREEDY algorithm could be viewed, following Theorems 2.2 and 4.2 as either a $2-$local algorithm with $O(\log n/\log\log n)$ probes w.h.p, or as a $1-$local algorithm having probing complexity of $O(\log^2 n/\log\log n)$. Other $1-$local algorithms could be though of, see for instance [24].

5.2 Lower Bounds in the Probing Model

Theorem 5.1. *(i) In a skip graph - any* 1-local *algorithm that outputs a path between two nodes at distance n, must probe $\Omega(\log n)$ probes, with probability at least $1-\frac{1}{n^\epsilon}$. In particular, the expected number of probes is $\Omega(\log n)$.*

(ii) In a $d-$dimensional small-world percolation network - *any* 1-local *algorithm that outputs a path between two nodes at distance n, must probe $\Omega(\log n)$ probes, with probability at least $1 - \frac{1}{n^\epsilon}$. In particular, the expected number of probes is $\Omega(\log n)$.*

The theorem implies that if a node holds only its neighbors then *any* routing algorithm would need $\Omega(\log n)$ probes w.h.p. Thus the assumption that nodes have some knowledge of their neighbor's neighbors is essential.

We first argue that GREEDY dominates any other $1-$local algorithm. The following lemma holds both for skip graphs and small-world percolation networks.

Lemma 5.2. *Let $\mathcal{A}$ be a $1-$local algorithm . Denote by A_d, G_d the random variables representing the number of probes it takes the algorithm $\mathcal{A}$ and the* GREEDY *algorithm respectively, to find a path between two nodes at distance d. For all $d > k > 0$ it holds that $\Pr[G_d \le k] \ge \Pr[A_d \le k]$.*

Proof. We distinguish between the two cases.

Small-World Percolation Networks For convenience, we label the target node as $\vec{0}$, and assume that the mesh is infinite. The trick is to give $\mathcal{A}$ some extra power. Assume that at some step, the closest node to $\vec{0}$ which $\mathcal{A}$ had found is at distance d from $\vec{0}$, where the distance is measured by the L_1 norm. At this point, we grant $\mathcal{A}$ access to all nodes outside a ball of radius d from $\vec{0}$. Now if $d_1 > d_2$ then for every configuration of edges, every move A can do in case the distance is d_1, is also available when the distance is d_2, so without loss of generality, for every k, $\Pr[A_{d_1} \le k] \le \Pr[A_{d_2} \le k]$. In other words, for every k, $\Pr[A_d \le k]$ is monotonically decreasing in d. The algorithm $\mathcal{A}$ samples some point v. The greedy choice is to sample a point closest 0, call that point u. Let $f(v)$ denote the the distance from $\vec{0}$ of the neighbor of v which is closest to $\vec{0}$. Now

$$\Pr[A_d \le k] = \sum_{i<d} \Pr[f(v) = i] \cdot \Pr[A_i \le k-1]$$

Since $||u|| \le ||v||$ it holds that for every i, $\sum_{j=0}^{i} \Pr[f(u) = j] \ge \sum_{j=0}^{i} \Pr[f(v) = j]$. Since $\Pr[A_d \le k]$ is monotonically decreasing we have:

$$\sum_i \Pr[f(u) = i] \Pr[A_i \le k-1] \ge \sum_i \Pr[f(v) = i] \Pr[A_i \le k-1]$$

In other words, the best thing $\mathcal{A}$ can do is sample the greedy point, which implies that the GREEDY algorithm dominates any other $1-$local algorithm.

Skip Graphs The technique used in the previous section applies here as well. Now if at some step the closest node to 0 which $\mathcal{A}$ had found is at distance d from 0 we supply to $\mathcal{A}$ both the access and the membership vectors of all the nodes in the segment $[n, d]$. We need to handle some dependencies. Denote by M_d the membership vectors of this segment. Assume that $\mathcal{A}$ probes point v and GREEDY probes point u. Using the notation of the previous section we have

$$\Pr[(A_d \le k)|M_d] = \sum_{i=0}^{d-1} \Pr[(f(v) = i)|M_d] \cdot \Pr[(A_i \le k-1)|M_d]$$

It is easy to see that for every instance of M_d, it holds that $\Pr[f(v) = i|M_d] \le \Pr[f(u) = i|M_d]$. To see this consider prefixes of length k. If v does not have a neighbor corresponding to a prefix of length k within the segment $[n, d]$ then the probability $f(v) = i$ due to a prefix of length k is equal to the probability $f(u) = i$ due to a prefix of length k. If v does have a neighbor corresponding to a prefix of length k in $[n, d]$ then $\Pr[f(v) = i] < \Pr[f(u) = i]$. Conclude that

$$\begin{aligned}\Pr[(A_d \leq k)|M_d] &= \sum_{i=0}^{d-1} \Pr[(f(v) = i) \mid M_d] \cdot \Pr[(A_i \leq k-1) \mid M_i] \\ &\leq \sum_{i=0}^{d-1} \Pr[(f(u) = i) \mid M_d] \cdot \Pr[(A_i \leq k-1) \mid M_i]\end{aligned}$$

which concludes the proof of the lemma. □

It remains to lower bound the number of hops taken by the GREEDY algorithm. Assume as before that the initial node is $||z|| = n$ and the destination is $\vec{0}$. Divide the nodes of the graph into sets $B_0, B_1, \ldots, B_{\log n}$ according to their distance from $\vec{0}$ (or L_1 norm), such that $B_i = \{u | 2^{i-1} \leq ||u|| < 2^i\}$. So $\vec{0} \in B_0$ and $z \in B_{\lceil \log n \rceil}$. We slightly change the GREEDY algorithm thus: if the algorithm reaches a node within a ball B_i it is granted access to all nodes with distance at least 2^{i-1} from 0, i.e. to all nodes in $B_i, B_{i+1}, ..., B_n$. When routing in a skip graph the algorithm is also given the membership vectors of these nodes. The reason for this change is to cancel the dependencies on previous hops, it may only reduce the number of hops GREEDY takes, since it allows the algorithm a 'free' hop to the edge of the ball B_i. For each $0 \leq i \leq \log n$ let X_i be the indicator of the event: "The path taken by GREEDY includes at least one vertex in B_i". Clearly the number of nodes in the path is at least $\sum_{i=o}^{\log n} X_i$.

Lemma 5.3. *Both for skip graphs and for small world graphs and for each i, it holds that*

$$\Pr[X_i = 1 | X_{i+1}, X_{i+2}, \ldots, X_{\log n}] \geq c$$

for some constant c independent of n.

Before proving the lemma we show why it suffices to prove Theorem 5.1. Let Y_i be a Bernoulli variable with $\Pr[Y_i = 1] = c$. Now $\mathbb{E}[\sum Y_i] = c \log n \leq \mathbb{E}[\sum X_i]$. Furthermore the random variable $\sum X_i$ dominates the random variable $\sum y_i$. We have

$$\Pr[\sum X_i \leq \frac{1}{2} c \log n] \leq \Pr[\sum Y_i \leq \frac{1}{2} c \log n] \leq \frac{1}{n^{\epsilon}}$$

according to Chernoff's bounds.

Proof of Lemma 5.3. As before we distinguish between the two cases:

Small World Percolation Networks Assume that the values of $X_{i+1}, \ldots, X_{\log n}$ are already set and that j is the smallest index such that $X_j = 1$. Since we changed the algorithm such that when a ball B_i is reached all nodes in it are revealed, it is clear that X_i is independent from $X_{j+1}, X_{j+2}, \ldots, X_{logn}$, it remains to analyze $\Pr[X_i = 1 | X_{i+1} = 0, X_{i+2} = 0, \ldots, X_j = 1]$. Let y be the node in B_j which is closest to 0, i.e. the node probed by GREEDY. The notation $y \sim B_i$ stands for the event - 'y is connected by an edge to B_i'. For convenience let $B = \cup_{j=0}^{i-1} B_j$. All edges are independent of each other. Therefore $\Pr[X_i = 1 | X_{i+1} = 0, X_{i+2} = 0, \ldots, X_j = 1]$ is the probability

y is connected to B_i and is not connected to $B_0, B_1, ..., B_{i-1}$, conditioned on it being connected to one of them. We need to compute:

$$\Pr[y \sim B_i \wedge y \not\sim B | y \sim B \cup B_i] = \frac{\Pr[y \sim B_i \wedge y \not\sim B]}{\Pr[y \sim B \cup B_i]}$$
$$= \frac{\Pr[y \sim B_i] \cdot \Pr[y \not\sim B]}{1 - Pr[y \not\sim B] \Pr[y \not\sim B_i]}$$

It is easy to verify that $\Pr[y \not\sim B] \geq \Pr[y \not\sim B_i]$ and that $\Pr[y \not\sim B] \geq \epsilon$ for some constant ϵ. We have:

$$\frac{\Pr[y \sim B_i] \cdot \Pr[y \not\sim B]}{1 - Pr[y \not\sim B] \Pr[y \not\sim B_i]} \geq \frac{(1 - \Pr[y \not\sim B_i]) \Pr[y \not\sim B]}{(1 - \Pr[y \not\sim B_i])(1 + \Pr[y \not\sim B])} \geq \frac{\epsilon}{1+\epsilon}$$

Skip Graphs Here we have to deal with some dependencies. Let D denote the event that the algorithm reached the node y (i.e. the segment B_j which contains y). As before we need to compute:

$$\frac{\Pr[(y \sim B_i \wedge y \not\sim B)|D]}{\Pr[y \sim B \cup B_i|D]}$$

The events $\{y \sim B_i\}$ and $\{y \not\sim B\}$ are positively correlated, even when conditioned on D. So the calculation of the previous section applies here as well. □

6 Implementations of NoN

We ran simulations in which we compared the performance of the GREEDY algorithm and the performance of the NoN-GREEDY algorithm. We constructed a skip graph of up to 2^{17} nodes and a small world percolation graph of up to 2^{24} nodes. In a small world graph it is not necessary to create the full graph in advance. Each time the message reached a node, we randomly created the neighborhood of radius 2 around the node. This is a negligible compromise over the definition of the model, since the edge in which the node was entered might not be sampled. This technique allowed us to run simulations on much larger graphs. For each graph size we ran 150 executions. A substantial improvement could be seen. Figures 3 and 4 demonstrate an improvement of about 48% for skip graphs of size 2^{17} and an improvement of 34% for small world percolation graphs of size 2^{24}. Figure 3 also depicts the average shortest path in the graph. We see that the shortest paths may be 30% shorter than the paths found by NoN, yet even for moderate network sizes, the NoN algorithm performs substantially better than then GREEDY.

An even more impressive improvement could be seen when the size of the graph is fixed and the average degree changes. We fixed a small world percolation graph of size 2^{20}. After that we deleted each edge with a fixed probability which varied from 0 to 0.9 (a graph with roughly one tenth of the edges). Figure 5 depicts the results of these simulations. It shows that the reduction in the number of hops is more or less independent from the number of edges. The path length achieved by the GREEDY algorithm when the degree is 26 is achieved by the NoN algorithm when the degree is merely 12. In the case of skip graphs we ran the simulation for a graph of size 2^{17} and varied the size of the alphabet of the membership vectors. When the alphabet size is s the average degree is $O(\log_s n)$. We can see in Figure 5 that NoN with alphabet size 20 is better than Greedy with alphabet size 2, i.e. when the average degree is $\log_2 20 \simeq 4.3$ bigger.

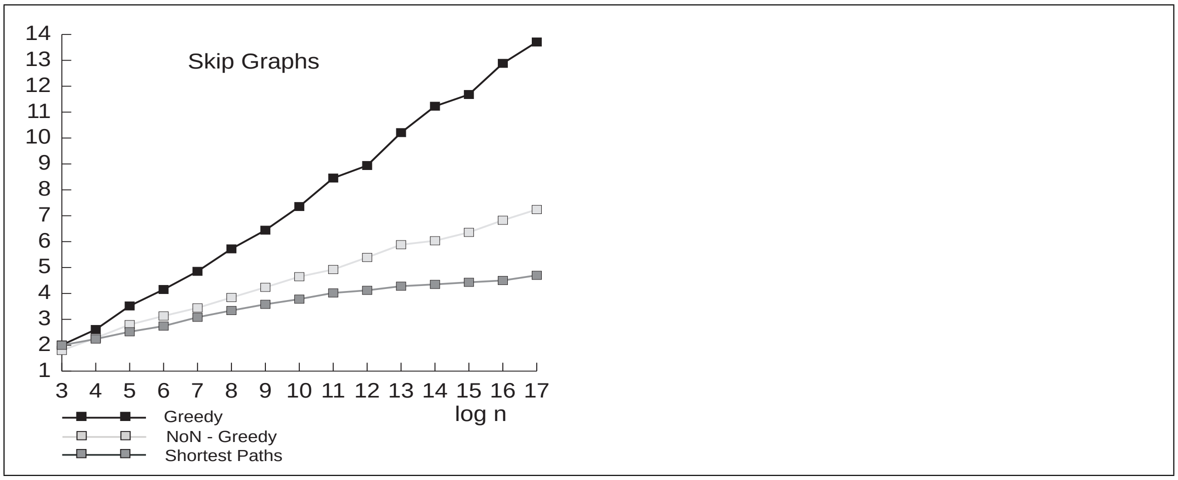

Figure 3: The number of hops in skip graphs.

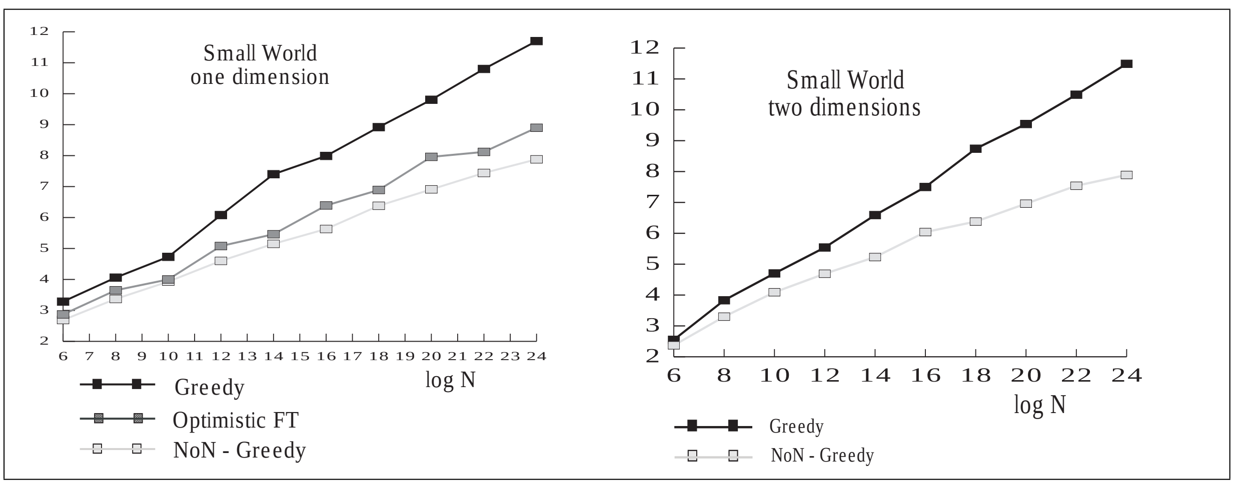

Figure 4: The number of hops in a small world percolation graphs of dimensions 1, 2.

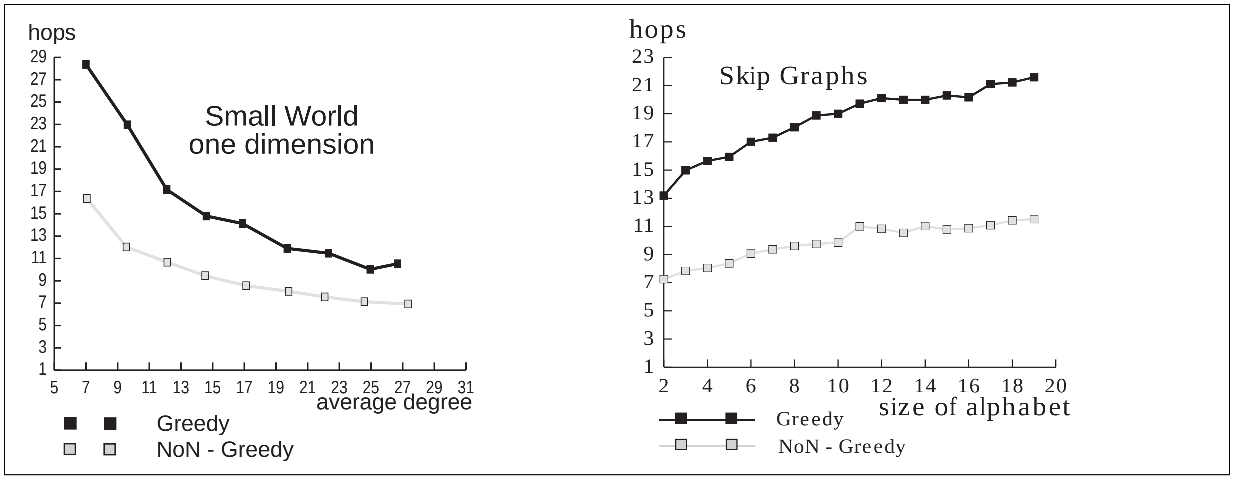

Figure 5: The tradeoff between average degree and latency in a small world with 2^{20} nodes (left) and skip graphs of 2^{17} nodes (right).

6.1 A different Implementation

The algorithm presented in Figure 1 is somewhat unnatural. Each NoN step has two phases. In the first phase the message is sent to a neighbor whose neighbor is close to the target. In the second phase a greedy step is taken (i.e. the message moves to the neighbor of neighbor). A 1−phase implementation would let each node initiate a NoN step again, i.e. each node upon receiving a message, finds the closest neighbor of neighbor, and passes the message on. This variant is harder to analyze, indeed Theorem 4.2 holds for the 2−phase version only. Yet, as Figure 6 shows, in practice the two variants have basically the same performance.

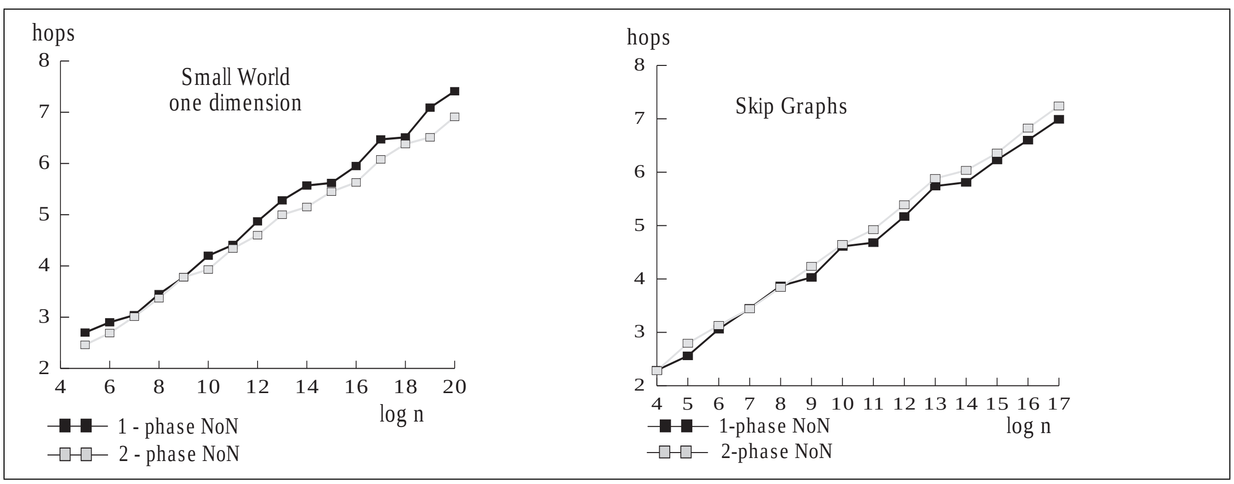

Figure 6: Comparison between the two variants of NoN

6.2 System Issues with NoN-GREEDY

An implementation of the NoN-GREEDY algorithm in a P2P network necessitates that each node acquire knowledge of its neighbor's neighbors. At first glance, it might appear that maintenance of such knowledge is problematic since it is tantamount to squaring the degree of the graph and therefore, squaring the size of the routing table at each node. However, it is important to note that the bottleneck in the system is actually the run-time cost of maintaining the TCP links between nodes. This cost remains unchanged, irrespective of which routing protocol we use: GREEDY or NoN-GREEDY. The primary concern in implementing NoN-GREEDY is the amount of communication-overhead needed to keep the neighbor-of-neighbor lists (reasonably) up-to-date. Updates could be piggy-backed on top of maintenance messages (the "keep-alive" messages). Moreover, the neighbor-of-neighbor information at a node does not have to be perfectly up-to-date at all times to derive the benefits of NoN-GREEDY routing, as could be seen in the next Section.

In the following we analyze more carefully the amount of communication needed in order to keep the routing lists up-to-date. The execution NoN requires that nodes should update each other regarding their own lists of neighbors. Such an update occurs in two scenarios:

1. Each node upon entrance sends its list of neighbors to its neighbors.
2. Whenever a node encounters a change in its neighbor list (due to the entrance or exit of a node), it should update its neighbors.

The extra communication imposed by these updates is not heavy due to the two following reasons. First, assume nodes u, v are neighbors. Node u periodically checks that v is alive (for instance by pinging it). Checking whether v's neighbor list has changed could be piggy-backed on the maintenance protocol by letting v send a hash of its neighbor list. A possible hash function may be `MD5` (though the cryptographic properties of this hash function are not needed). Another possibility is simply to treat the id of neighbors as coefficients of a polynomial, and evaluate this polynomial at a random point. Either way the actual cost in communication is very small. When an actual update occurs there is no reason for v to send its entire neighbor list. It may only send the part of it which u misses. If it does not know which part it is then u, v may participate in a very fast and communication efficient protocol that reconciles the two sets, see e.g. [32] for details. The second reason the communication overhead is small is that the the actual updates are not urgent (as the next section will show) and may be done when the system is not busy.

It is important to notice that the implementation of the NoN algorithm does not affect the Join/Leave operations - the needed updates are passe only after the node enters the system. We conclude that implementing NoN has little cost both in communication complexity and in internal running time. It is almost a free tweak that improves performance considerably and may be implemented on top of existing constructions.

6.3 Fault Tolerance

The previous simulations assumed that the list of neighbor's neighbors each node holds is always correct. In reality this might not be the case. We examine two scenarios which capture the two extremes of this problem.

Optimistic Scenario: In this case we assume that a node knows whether its neighbors of neighbors lists are up-to-date or not. Whenever a node has a stale list it avoids a NoN hop and performs instead a greedy step based on its own neighbors list. We ran simulations in which each node performs with probability $\frac{1}{2}$ a NoN step, and with probability $\frac{1}{2}$ a greedy step. Whenever a NoN step is performed, both phases of it are performed correctly. Figures 4 and 7 show that the total performance is hardly compromised. A small world of size 2^{22} suffered a relative delay of less than one hop, A skip graph of size 2^{17} suffers a relative delay of 1.2 hops. But why is the optimistic scenario justified? Our suggestion is that each node would calculate a hash of its neighbors list. This hash would be sent to all its neighbors on top of the maintenance messages. Thus with a minuscule overhead in communication each node would know whether its lists are up-to-date.

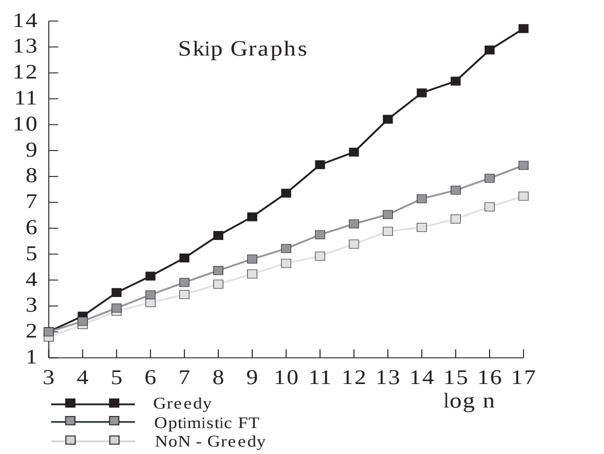

Figure 7: Optimistic fault tolerance in Skip Graphs

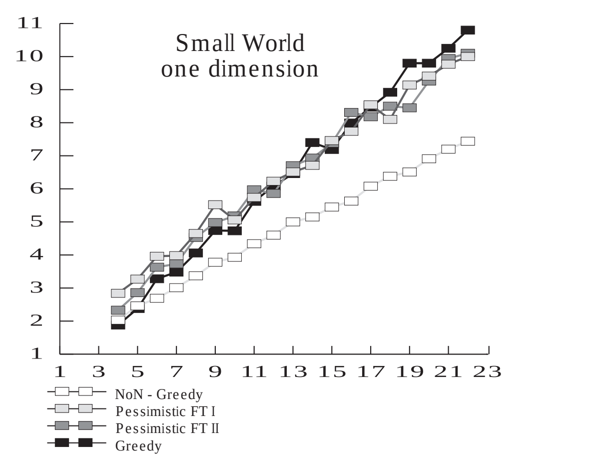

Figure 8: Pessimistic Fault Tolerance in Percolation Small World

Pessimistic Scenario: In this scenario we assume that a node is unaware that its neighbor's neighbors lists are not up-to-date. So when node u passes a message to node w expecting it to move on to node z, with probability $\frac{1}{2}$ the edge (w, z) no longer exists. We tested two variants: in the first one, whenever this occurs the intermediate node w performs a greedy step. In the second variant the intermediate node w initiates another NoN step. The results of the simulations appear in Figure 8. It could be seen that in the pessimistic scenario, the performance of NoN is approximately the same as the Greedy algorithm.

We conclude that the NoN-GREEDY algorithm is beneficia even if the neighbor of neighbor lists are error prone.

7 Discussion

Randomization Reduces Latency: A common strategy in the design of P2P routing networks is to first identify a static graph which is known to possess good properties, and then to adapt the static graph topology to handle the dynamism (arrival/departure of nodes) and scale (changes in the average number of nodes). The resulting dynamic routing network resembles the underlying static graph closely. In the case of skip graphs, a 'perfect' skip graph has the i^{th} edge of each node cover a distance of 2^i, i.e., the lengths of edges of a node form a geometric series. The randomization involved in the dynamic construction is usually considered as a negative by-product and much effort is put in reducing it. For instance, a deterministic P2P routing network that guarantees that the skip graph is almost 'perfect' is presented in [19]. As was noticed by Harvey *et al* [20], a perfect skip graph is similar to Chord [39]. The average length of shortest paths in both Chord (studied in [16]) and hypercubes is $\Omega(\log n)$. Xu *et al* [40] proves that if edges are added to the cycle deterministically such that the existence of an edge (x, y) is a function of $|x - y|$ (and not say x, y themselves), then the diameter of a network of degree $\log n$ is bounded by $\Omega(\log n)$. This leads to the following counter-intuitive and surprising fact:

> *Randomization of edges* reduces *the average length of shortest paths in the hypercubes, Chord and Skip Graphs.*

The reason is that the randomization enables a routing algorithm to use an 'exceptionally' long edge once in a while. The density of these long edges is just large enough so that the NoN-Greedy algorithm finds them. In a 'perfect' skip graph, Chord, and in the hypercube - these long edges do not exist. Our results show that safety has a price: while these network topologies have guaranteed worst-case route-lengths, they enlarge the expected length of routes.

Acknowledgments

We thank James Aspnes and Gauri Shah for supplying us with their implementation of skip graphs, and for reading and commenting on an earlier version of the paper.

References

[1] Ittai Abraham, Baruch Awerbuch, Yossi Azar, Yair Bartal, Dahlia Malkhi, and Elan Pavlov. A generic scheme for building overlay networks in adversarial scenarios. In *Proc. Intl. Parallel and Distributed Processing Symposium (IPDPS 2003)*, April 2003.

[2] Ittai Abraham, Dahlia Malkhi, and Gurmeet S. Manku. Brief announcement: Papillon: Greedy routing in rings. In *DISC*, 2005.

[3] Michael Aizenman and Charles M. Newman. Discontinuity of the percolation density in one dimensional $1/|x-y|^2$ percolation models. *Communications in Mathematical Physics*, 107:611–647, 1986.

[4] James Aspnes, Zoë Diamadi, and Gauri Shah. Fault-tolerant routing in peer-to-peer systems. In *Proc. 21st ACM Symp. on Principles of Distributed Computing (PODC 2002)*, pages 223–232, July 2002.

[5] James Aspnes and Gauri Shah. Skip graphs. In *Proc. 14th ACM-SIAM Symp. on Discrete Algorithms (SODA 2003)*, pages 384–393, January 2003.

[6] Alberto László Barabási. *Linked: The New Science of Networks*. Perseus Publishing, 2002.

[7] Lali Barrière, Pierre Fraigniaud, E Kranakis, and D Krizanc. Efficient routing in networks with long range contacts. In *Proc. 15th Intl. Symp. on Distributed Computing (DISC 2001)*, pages 270–284, October 2001.

[8] Itai Benjamini and Noam Berger. The diameter of a longe-range percolation clusters on finite cycles. *Random Structures and Algorithms*, 19(2):102–111, 2001.

[9] Miguel Castro, Peter Druschel, Y Charlie Hu, and Antony I T Rowstron. Topology-aware routing in structured peer-to-peer overlay networks. In *Proc. Intl. Workshop on Future Directions in Distrib. Computing (FuDiCo 2003)*, pages 103–107, 2003.

[10] H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Annals of Mathmematical Statistics*, 23:493–509, 1952.

[11] Don Coppersmith, David Gamarnik, and Maxim Sviridenko. The diameter of a longe-range percolation graph. *Random Structures and Algorithms*, 21(1):1–13, 2002.

[12] Ithiel de Sola Pool and Manfred Kochen. Contacts and influence. *Social Networks*, 1:1–48, 1978.

[13] Peter Sheridan Dodds, Muhamad Roby, and Duncan J Watts. An experimental study of search in global social networks. *Science*, 301:827–829, 2003.

[14] Pierre Fraigniaud and Philippe Gauron. (brief announcement) An overview of the content-addressable network D2B. In *Proc 22nd ACM Symposium on Principles of Distributed Computing (PODC 2003)*, pages 151–151, July 2003.

[15] Pierre Fraigniaud, Cyril Gavoille, and Christophe Paul. Eclecticism shrinks even small worlds. In *Proc. 23rd ACM Symposium on Principles of Distributed Computing, (PODC).*

[16] Prasanna Ganesan and Gurmeet Singh Manku. Optimal routing in Chord. In *Proc. 15th ACM-SIAM Symp. on Discrete Algorithms (SODA 2004)*, pages 169–178, January 2004.

[17] Krishna P Gummadi, Ramakrishna Gummadi, Steven D Gribble, Sylvia Ratnasamy, Scott Shenker, and Ion Stoica. The impact of DHT routing geometry on resilience and proximity. In *Proc. ACM SIGCOMM 2003*, pages 381–394, 2003.

[18] Torben Hagerup and Christine Rüb. A guided tour of Chernoff bounds. *Information Processing Letters*, 33(6):305–308, 1990.

[19] Nicholas Harvey and J Ian Munro. (brief announcement) deterministic Skipnet. In *Proc 22nd ACM Symposium on Principles of Distributed Computing (PODC 2003)*, pages 152–153, 2003.

[20] Nicholas J A Harvey, Michael Jones, Stefan Saroiu, Marvin Theimer, and Alec Wolman. Skipnet: A scalable overlay network with practical locality properties. In *Proc. 4th USENIX Symposium on Internet Technologies and Systems (USITS 2003)*, 2003.

[21] M Frans Kaashoek and David R Karger. Koorde: A simple degree-optimal hash table. In *Proc. 2nd Intl. Workshop on Peer-to-Peer Systems (IPTPS 2003)*, pages 98–107, 2003.

[22] Jon Kleinberg. Navigation in a small world. *Nature*, 406:845, 2000.

[23] Jon Kleinberg. The small-world phenomenon: An algorithmic perspective. In *Proc. 32nd ACM Symposium on Theory of Computing (STOC 2000)*, pages 163–170, 2000.

[24] Emmanuelle Lebhar and Nicolas Schabanel. Searching for optimal paths in long-range contact networks. Technical Report RR LIP 2003-55, Computer Science Department, École Normale Supérieure de Lyon (ÉNS Lyon), 2003.

[25] Emmanuelle Lebhar and Nicolas Schabanel. Almost optimal decentralized routing in long-range contact networks. In *Proceedings of th 31st International Colloquium on Automata, Languages and Programming (ICALP)*, pages 894–905, 2004.

[26] Dahlia Malkhi, Moni Naor, and David Ratajczak. Viceroy: A scalable and dynamic emulation of the butterfly. In *Proc 21st ACM Symposium on Principles of Distributed Computing (PODC 2002)*, pages 183–192, 2002.

[27] Gurmeet Singh Manku. Routing networks for distributed hash tables. In *Proc. 22nd ACM Symp. on Principles of Distributed Computing (PODC 2003)*, pages 133–142, July 2003.

[28] Gurmeet Singh Manku, Mayank Bawa, and Prabhakar Raghavan. Symphony: Distributed hashing in a small world. In *Proc. 4th USENIX Symposium on Internet Technologies and Systems (USITS 2003)*, pages 127–140, 2003.

[29] Gurmeet Singh Manku, Moni Naor, and Udi Wieder. Know thy neighbour's neighbour: The role of lookahead in randomized P2P networks. In *Proc. 36th ACM Symposium on Theory of Computing (STOC 2004)*, 2004.

[30] Charles Martel and Van Nguyen. Analyzing kleinberg's (and other) small-world models. In *Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing (PODC)*, pages 179–188, 2004.

[31] Stanley Milgram. The small world problem. *Psychology Today*, 67(1):60–67, May 1967.

[32] Yaron Minsky and Ari Trachtenberg. Practical set reconciliation. Technical Report 2002-03., Boston University, 2002.

[33] Moni Naor and Udi Wieder. Novel architectures for P2P applications: The continuous-discrete approach. In *Proc. 15th ACM Symp. on Parallelism in Algorithms and Architectures (SPAA 2003)*, pages 50–59, June 2003.

[34] Moni Naor and Udi Wieder. Know thy neighbor's neighbor: Better routing for skip-graphs and small worlds. In *The Third International Workshop on Peer-to-Peer Systems (IPTPS)*, 2004.

[35] Charles M Newman and Lawrence S. Schulman. One dimensional $1/|j-i|^s$ percolation models: The existence of a transition for $s \leq 2$. *Communications in Mathematical Physics*, 180:483–504, 1986.

[36] William Pugh. Skip lists: A probabilistic alternative to balanced trees. *Communications of the ACM*, 33(6):668–676, June 1990.

[37] Antony I T Rowstron and Peter Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001)*, pages 329–350, 2001.

[38] Lawrence S. Schulman. Long range percolation in one dimension. *Journal of Physics A*, 16(17):L639–L641, 1983.

[39] Ion Stoica, Robert Morris, David Karger, M Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proc. ACM SIGCOMM 2001*, pages 149–160, 2001.

[40] Jun Xu, Abhishek Kumar, and Xingxing Yu. On the fundamental tradeoffs between routing table size and network diameter in peer-to-peer networks. In *INFOCOM*, 2003.

[41] Hui Zhang, Ashish Goel, and Ramesh Govindan. Incrementally improving lookup latency in distributed hash table systems. In *ACM SIGMETRICS 2003*, pages 114–125, June 2003.

[42] Ben Y Zhao, Ling Huang, Jeremy Stribling, Sean C Rhea, Anthony D Joseph, and John D Kubiatowicz. Tapestry: A resilient global-scale overlay for service deployment. *IEEE Journal on Selected Areas in Communications*, 22(1), January 2004.