arXiv:cs.DS/0607025 v1 7 Jul 2006

The Evolution of Navigable Small-World Networks

Oskar Sandberg [*], Ian Clarke [†]

July 7, 2006

Abstract

Small-world networks, which combine randomized and structured elements, are seen as prevalent in nature. Several random graph models have been given for small-world networks, with one of the most fruitful, introduced by Jon Kleinberg [10], showing in which type of graphs it is possible to route, or navigate, between vertices with very little knowledge of the graph itself.

Kleinberg's model is static, with random edges added to a fixed grid. In this paper we introduce, analyze and test a randomized algorithm which successively rewires a graph with every application. The resulting process gives a model for the evolution of small-world networks with properties similar to those studied by Kleinberg.

[*]The Department of Mathematical Sciences, Chalmers Technical University and Gothenburg University. ossa@math.chalmers.se

[†]FreenetProject Inc. ian@freenetproject.org

1 Introduction

The "Small World Phenonomenon" is the name given to the observation that seemingly random people can often find a short chain of aquaintances connecting them to one another. Mathematically, this has been related [18] [16] to the observation that structured graphs, such as grids, can have their diameter drastically reduced by the introduction on some random edges between the vertices (as proved for the circle in [3]).

Connected with this is the question, raised by Jon Kleinberg in 2000 [10], whether short paths can be found between any two vertices by actors in the network lacking global information about the graph to use when routing. He showed that this is not possible in all families of random graphs with small diameter, but instead depends on very specific properties of certain classes of such graphs. Graphs where short paths can be found are often called "navigable".

The question of whether graphs are navigable is of particular practical interest due to a multitude of applications. Specifically, the type of routing suggested by Kleinberg has been employed in distributed computing, hashtables [14] and peer-to-peer software [6].

1.1 Motivation

While previous results go a long way towards characterizing when graphs are navigable, they leave open the question of how such graphs are formed. At the same time, experiments with social networks (e.g. [15] [8]), seem to indicate that those do, to at least some extent, have navigable properties. A model for evolution and growth of navigable graphs, similar in some respects to the preferential-attachment models of power-law degree distributions [4] [1], would help explain when and how they arise through natural processes. Such a model could also be used to generate graphs for use in networks where efficient routing is important, such as the types of overlay networks on the Internet mentioned above.

In a recent summary paper [12] Kleinberg identifies this problem as one of central open issues in the area.

1.2 Contribution

We summarize our contributions as follows:

1. We present an evolving random graph model where the edges of a graph are re-wrired by performing repeated greedy walks between random points, and changing the edges based on these.

2. We analyze rigourously, under a few simplifications, the stationary case of the model, showing that it is a navigable random graph.

3. We simulate the algorithm in a number of different circumstances, showing that it leads to graphs that perform as well or better then those produced with Kleinberg's model.

1.3 Previous Work

In a followup to his original work [11], Jon Kleinberg motivated why the necessary distribution for navigability might arise in nature by means of "group memberships". He showed that in a more generalized setting, structures are navigable if two vertices are connected with a probability that is inversely proportional the size of the smallest group they both populate. That this should be the case is in some sense natural, since the probability of knowing somebody may decrease with the size of the group in which you know them. Similar arguments can be found in [13] and [17].

A preprint paper by Clauset and Moore [7] presents a different re-wiring algorithm for the creation of navigable graphs. They show positive results for this algorithm using simulation, but do not present any analytic results. In [9] a re-wiring algorithm for the creation of so called scale-free (or power-law) graphs is presented. This does not deal with clustering nor navigability, and no analytic results regarding the stationary distribution are derived.

Early versions of the Freenet peer-to-peer data network, presented in [5] and [6], used a method similar to the algorithm we propose to update the links between peers. The current work is in part inspired by trying to apply the ideas from the design of Freenet to an environment more conductive to analysis. [19] previously related Freenet to the discussion of navigable small-world graphs, but they worked mostly on proposing modifications to the algorithm that resulted in a more robust network, instead of looking more closely at the properties of Freenet's neighbor sampling.

2 Navigable Small Worlds

In his initial study of navigable graphs [10], Kleinberg studied graphs constructed by starting with a two dimensional grid, and adding random long-range contacts according to a certain class of distributions. For the purpose of vertex to vertex routing in such graphs, he defined a decentralized algorithm as one where each vertex has to make a routing decision based only on the grid positions of the query's destination and the vertex's immediate neighborhood[1].

Kleinberg showed that if one starts with a two dimensional grid and adds long-range edges between vertices x and y with probability $\propto |x-y|^{\alpha}$ where $|x-y|$ denotes Manhattan-distance in the grid, then only the case $\alpha = -2$ allows for decentralized

[1] For the upper bounds, he also allowed vertices to know the grid position of all previous vertices in the query and their neighbors. This was not used in the lower bounds, and so strengthens both results.

routing in a polylogarithmic number of steps. For all other values of α a lower bound which is a fractional power of the graph size can be derived.

In the critical case $\alpha = -2$, however, it is sufficient to use the most direct routing possible, so called *greedy routing*. As the name implies, greedy routing means that at each step, one attempts to minimize the distance to the destination. That is, if x wishes to route a query to vertex z, then he picks as the next step the one of his neighbors (long-range or otherwise) which is closest to z. If n is the size of the graph, then a bound of $O(\log^2 n)$ on the expected number of steps needed can be found. Kleinberg's model can easily be extended to graphs formed by adding long-range edges to grids of dimension other than two. If the basic grid has dimension d, it can be seen that $\alpha = -d$ corresponds to the critical case in which routing is possible.

3 The Algorithm

We let V be the set of vertices, each with a position in a grid or some other regular lattice. We will let E be set of directed shortcut (long-range) edges between vertices in V, and $G = (V, E)$ the resulting digraph. Let G' be G augmented by additional edges going both ways between each pair of vertices that are adjacent in the lattice. The proposed algorithm, which we call *destination sampling*, is as follows:

Algorithm 3.1. *Let $G_s = (V, E_s)$ be the directed graph of shortcuts at time s. Let $0 < p < 1$. Then G_{s+1} is defined as follows.*

1. *Choose y_{s+1} and z_{s+1} uniformly from V.*
2. *If $y_{s+1} \neq z_{s+1}$, do a greedy walk in G'_s from y_s to z_s along the lattice and the shortcuts of E_s. Let $x_0 = y_{s+1}, x_1, x_2, ..., x_t = z_{s+1}$ denote the points of this walk.*
3. *For each $x_0, x_1, ..., x_{t-1}$ with at least one shortcut, independently with probability p replace a randomly chosen shortcut with one to z_{s+1}.*

After a walk is made, G_{s+1} is the same as G_s, except that a shortcut from each vertex in walk $s+1$ is with probability p replaced by an edge to the destination. In this way, the destination of each edge is a sample of the destinations of previous walks passing through it. The claim is that updating the shortcuts using this algorithm eventually results in a shortcut graph with greedy path-lengths of $O(\log^2 n)$.

The value of p is a parameter in the algorithm. It serves to disassociate the shortcut from a vertex with that of its neighbors. For this purpose, the lower the value of $p > 0$ the better, but very small values of p will also lead to slower sampling.

4 Analysis

The algorithm above is stated in full generality, but for the sake of analysis, we will make a couple of simplifications. Firstly, it is advantageous to replace the two

dimensional lattice used by Kleinberg with a one dimensional ring of vertices, and move to the directed case where edges follow a single orientation. This means that the lattice distance is the number of steps following the orientation of the ring from one vertex to another. Bariere et al. [2] have performed a thorough investigation of this setting. The case $\alpha = -1$ here corresponds to the single critical, navigable case of Kleinberg's model where greedy routing performs in $O(\log^2 n)$ steps, other values of α do not allow for decentralized routing in a polylogarithmic number of steps.

Secondly we will study only the case where each vertex has exactly one shortcut. Graphs with multiple shortcuts can be derived from this by coalescing multiple vertices, or by slight variations in the analysis. A final simplification of the model we analyze, shortcut independence, will be introduced below.

4.1 Notation

We will index the set of vertices V such that the edges of the base graph are negatively oriented, in the sense that there is an edge from x to $x - 1 \mod n$ for all $x = 0 \dots n-1$. The function $d(x, z)$ gives the distance in this digraph from x to z. It is not symmetric, for example $d(x, x-1) = 1$ while $d(x-1, x) = n-1$. The probability space used will be $V \times V \times \{E : V \mapsto V\}$ with elements (y, z, E) denoting a starting point, destination, and shortcut configuration respectively. On this we define a probability measure $\mathbf{P}$, which chooses the three elements independently, the first two uniformly, and the third with probability defined below.

We will denote by $\ell(x, z)$ the marginal probability that x has a link to z. We let D_z be the event that z is chosen as the destination of a query, and H_x be the event that a query passes through x. The conditional hitting probability of x is denoted by $h(x, z) = \mathbf{P}(H_x \mid D_z)$: that is $h(x, z)$ is the probability that a query from a uniformly selected starting point with destination z passes through the point x. By translation invariance, both $\ell(x, z)$ and $h(x, z)$ are functions of $d(x, z)$, and we will sometimes see them as such (i.e. we let $\ell(x) = \ell(x, 0)$ so that $\ell(d(x, z)) = \ell(x, z)$ and define $h(x)$ equivalently.)

For sets $A \subset V$ we let $\ell(A) = \sum_{x \in A} \ell(x)$ and $h(A) = \sum_{x \in A} \ell(x)$. We let $\tau = h(V) = \sum_{\xi=1}^{n-1} h(\xi)$ and note that τ is exactly the expected greedy routing time of a query from a uniformly chosen point to 0.

4.2 Markov Chain

Each application of Algorithm 3.1 defines the transition of a Markov chain on the set of shortcut configurations. Thus for any n, the Markov chain in question is defined on a finite (if large) state space. Since it can easily be seen that this chain is irreducible and aperiodic, the chain converges to a unique stationary distribution. The goal is to motivate that this distribution leads to short greedy walks. The shortcut from a vertex x at any time is simply a sample of the destination of the previous walks that x has seen. Under the stationary distribution this should not

change with time, so marginally it holds that

$$\ell(x,z) = \mathbf{P}(D_z \mid H_x).$$

By using Bayes' theorem, and the definition above, we can thus write the shortcut distribution in terms of the hitting probability:

$$\ell(x,z) \quad = \quad \frac{h(x,z)\mathbf{P}(D_z)}{\sum_{\xi \neq x} h(x,\xi)\mathbf{P}(D_\xi)}.$$

Since the destination is chosen uniformly at random, $\mathbf{P}(D_\bullet)$ cancels out in numerator and denominator, and we are left with:

$$\ell(x,z) = \frac{h(x,z)}{\sum_{\xi \neq x} h(x,\xi)} = \frac{h(x,z)}{\sum_{\xi=1}^{N-1} h(\xi)} \tag{1}$$

where the last equality follows by using translation independence and re-indexing. In other words, $\ell(x) \propto h(x)$ for all x: we will call shortcut distributions which have this property *balanced*.

4.3 The Independent Case

In order to get a bound on the expected greedy routing time, we will need to make one further simplification. Instead of studying the true stationary distribution of the rewiring process, we will look at graphs where links are chosen independently in such a way that (1) holds. There is no reason to believe that there is independence under the true distribution (in fact, it is quite clear that there isn't), but below we will argue heuristically that these results are still valid.

Theorem 4.1. *For all $n \geq 1$, there exists a distribution $\ell(x)$ on $x \in [n-1]$ which is balanced when shortcuts are chosen independently at each node.*

Proof. If we consider each shortcut as chosen independently, we may view the query, which approaches the destination in each step, as being a Markov chain, and using the backwards equations for the hitting probability of Markov chains, we may deduce that (fixing the destination as 0):

$$h(x) = \sum_{\xi=x+1}^{N-1} h(\xi)\ell(\xi - x) + h(x+1) \sum_{\xi=x+2}^{N-1} \ell(\xi) + \frac{1}{n-1}. \tag{2}$$

The first term above gives the probability that we enter x using a shortcut from a vertex that is ξ steps from the destination, while the second term gives the probability that we enter x from the vertex which is $x+1$ steps from the destination using the base graph.

Fix a distribution $\ell'(x)$. The hitting probability under this distribution $h'(x)$ can be derived from (2), and from this we may derive a new distribution

$$\ell''(x) = \frac{h'(x)}{\sum_{i=1}^{n-1} h'(x)}.$$

The mapping of $\ell' \mapsto \ell''$ is continuous, since $\sum_{i=1}^{n-1} h'(x) > 1$, and maps a convex set (the simplex of $n-1$ valued distributions) into itself. By Brouwer's fix-point theorem, there exists at least one fixpoint ℓ^*, which by construction is a balanced distribution. □

We also note that:

Lemma 4.2. *If the shortcut configuration is chosen according to a translation invariant distribution, then $h(x)$ is non-increasing in x.*

Proof. This can been seen easily by considering any realization of the graph, together with a starting point, which causes a query for 0 to pass through $x+1$. For each such case, there is a corresponding configuration and starting point attained by translating each down one step (modulo n), for which a query for 0 will pass through x. □

Using this we may state and prove the main result:

Theorem 4.3. *For every $N = 2^k$ with $k \geq 4$, let τ be the expected greedy routing time. If shortcuts are selected independently according to a balanced distribution at each node, then*

$$\tau \leq 3k^2.$$

Proof. We fix the destination as 0, and consider routing from a randomly chosen point. Start by dividing $1, \ldots, n-1$ into k contiguous parts F_1,$F_2, \ldots, F_k$ such that

$$h(F_1) \approx h(F_2) \approx \ldots \approx h(F_k)$$

in the sense that $|h(F_i) - h(F_j)| < 2$ for all i, j (such a partition is possible since $h(x) \leq 1$ for all x). It follows by proportionality that

$$\ell(F_i) \geq \frac{1}{k} - \frac{1}{\tau} = \frac{\tau - k}{\tau k}$$

Let $r_0 = 0$, and $F_i = \{r_{i-1}+1, r_i\}$. We now consider a query starting at $r_k = n-1$, the furthest point from 0 in F_k, and want to find the probability that r_k has a shortcut to a vertex in $\{0, \ldots, r_{k-1}\}$.

Assume that $r_{k-1} > r_k/2$. Then $F_k \cap \{r_k - F_k\} = \emptyset$, so the desired probability is at least $\ell(r_k, r_k - F_k) = \ell(F_k)$. It follows from Lemma 4.2 that the probability of finding such a link cannot be less for any other point in F_k. The expected number

of steps spent in F_k is thus bounded from above by the expectation of a geometric random variable with success probability $\frac{\tau-k}{\tau k}$, whence

$$h(F_k) \leq \frac{\tau k}{\tau - k}. \tag{3}$$

However, the expected time spent in each F_i differs at most by a constant, so we can conclude that:

$$\tau = \sum_{i=1}^{k} h(F_i) \leq 2\frac{\tau k^2}{\tau - k}.$$

which implies $\tau \leq 2k^2 + k \leq 3k^2$.

This leaves the case when $r_{k-1} \leq n/2$. If this holds, then by the same reasoning, starting instead at r_{k-1}, we may exclude any case but $r_{k-2} \leq r_{k-1}/2 \leq n/4$. Continuing in this fashion, we can exclude every case but

$$r_1 \leq \frac{n}{2^{k-1}} = 2.$$

The result then follows again since $h(F_1) \leq r_1$ and $2 < k$. □

4.4 Dependencies

In order to fully prove that the rewiring algorithm presented above leads to a navigable graph, one needs to prove that the dependencies present in the resulting distribution of shortcuts are not destructive to the argument. In fact, our reasoning uses independence only at one point. In the proof of Theorem 4.3, after having calculated a marginal bound of $\approx 1/k$ of the probability that each point in F_i has a shortcut to a point outside the phase but closer to 0, we conclude (3): that this means that the expected number of steps in F_i is at most $\approx k$. This is true only if we draw a shortcut independently at each vertex in F_i that we reach, or if conditioned on not having found a useful shortcut in one step, the probability of doing so in the next increases.

Proving the full result formally is still an open problem, but one can see heuristically why it should hold. There are two forms of dependence present between edges created using the destination sampling algorithm. The first comes from the fact that two edges may have been created from the same query, and thus have the same destination. The parameter p is introduced into the algorithm to alleviate this (if p is large, one can very clearly see that nearby vertices tend to have the same shortcut destination, with considerable cost to routing performance), and by choosing p sufficiently small, we can make it negligible. The second type of dependence comes from the fact that what other edges are present around a vertex x will, of course, greatly affect the probability of whether a query for some vertex z passes through x.

When trying to bound the expected time spent in each F_i in the proof of Theorem 4.3, and thus calculating the probability that x has a shortcut that takes the query out of F_i, we have to condition on the previously encountered vertices having shortcuts that failed to do this. These could either be shortcuts from one earlier vertex in F_i to another, or shortcuts that overshoot the target (0) and thus are not used by the query. The presence of neither type of shortcut would seem to make it less likely that a query for a point in $A = \{0, \ldots, r_{i-1}\}$ passes through x, and hence one would not expect that the conditioning should make $h(x, A)$ (and thus $\ell(x, A)$) smaller. Formalizing this argument has, however, proved difficult.

5 Simulation

Simulations indicate that the algorithm gives results which scale as desired in the number of greedy steps, and that the resulting shortcut distribution approximates $1/\log(n)d(x, y)$ as expected.

The results in the directed one-dimensional case can be seen in Figure 1. To get these results, the graph is started with no shortcuts, and then the algorithm is run $10N$ times to initialize the references. The value of $p = 0.1$ is used. The greedy distance is then measured as the average of 100,000 walks, each updating the graph according to the algorithm (this decreases the variance of the estimate).

The square root of the mean greedy distance increases linearly as the graph size increases exponentially, just as we would expect. In fact our algorithm leads to better simulation results than choosing from Kleinberg's distribution. Doubling the graph size is found to increase the square route of the greedy distance by ≈ 0.41 when links are selected using our algorithm, compared to an increase of ≈ 0.51 when Kleinberg's model is used. (For Kleinberg's model we can use (2) to calculate numerically exact values for τ, allowing us to confirm this figure.)

In Figure 3 the marginal distribution of shortcut lengths is plotted. It is roughly harmonic in shape, except that it creates less links of length close to the size of the graph. This may be part of the reason why it is able to outperform Kleinberg's model: while Kleinberg's model is asymptotically correct, this algorithm takes into account finite size effects. (This reasoning is similar to that of the authors of [7]. Like them, we have no strong analytic arguments for why this should be the case, which makes it a tenuous argument at best.)

The algorithm has also been simulated to good effect using base graphs of higher dimensions. Figure 2 shows the mean greedy distance for two dimensional grids of increasing size. Here also, the algorithm creates configurations that seem to display square logarithmic growth, and which perform considerably better than explicit selection according to Kleinberg's model.

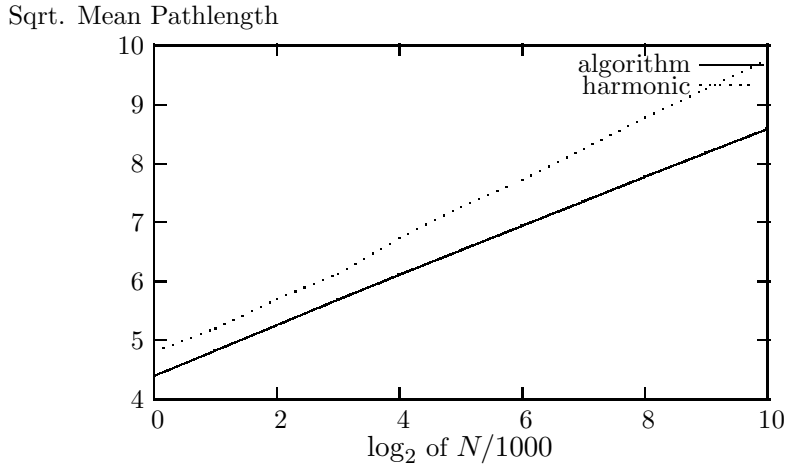

Figure 1: The expected greedy walk length using our selection algorithm, compared to selection according the harmonic distribution, in a directed ring.

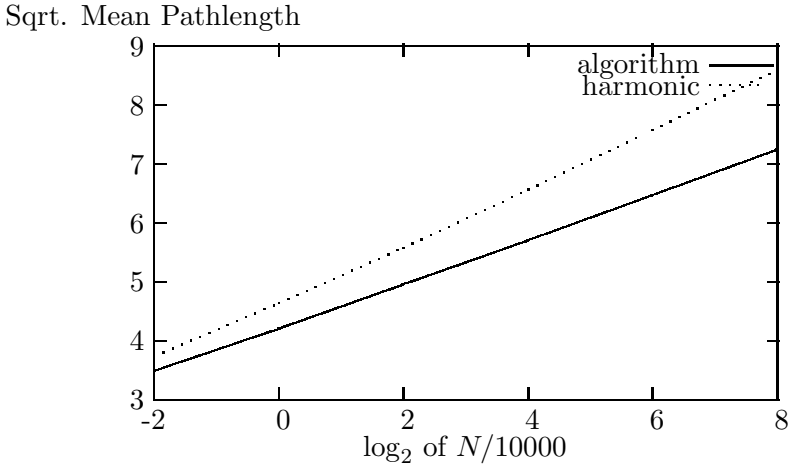

Figure 2: The expected greedy walk length of the selection algorithm, compared to selection according to harmonic distances, in a two dimensional base grid.

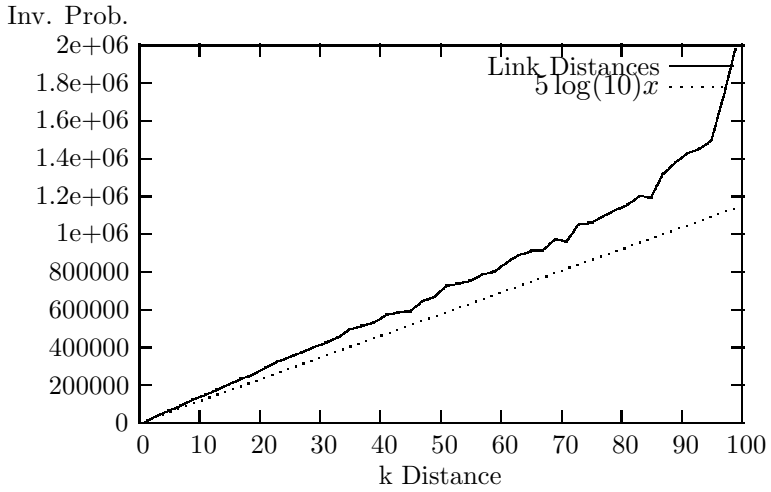

Figure 3: The inverse of distribution of shortcut distances, with $N = 100000$, $p = 0.10$. The straight line is the inverse of the harmonic distribution.

6 Conclusion

We have introduced an evolutionary model that by successively updating the the shortcut edges of a small-world graph creates configurations which are navigable. We have explored this model both analytically and with the help of simulation, and found support for the claim that navigability should arise.

The major open question is to complete the rigorous analysis of the stationary distribution, in particularly with regard to the dependencies between shortcuts. Random graphs with dependencies between the edges are notoriously difficult to analyze mathematically, and possibility of doing so usually relies on finding a formulation where the edges can be seen to be independent conditioned on a certain event (already Kleinberg's model is an example of this: the edges do not exist independently, but are independent conditioned on the position of the nodes). No such formulation has been found here, but the existence cannot be ruled out.

Further, there are interesting questions regarding the scope of the results. We have a upper bound for the navigable case which matches Kleinberg, but it would be interesting to see if lower bounds can be found for the case when $\ell(x, z)$ deviates from greatly from proportionality to $h(x, z)$, in the notation above. While it seems clear that this must be the case, a direct proof would be illustrative. Finally, it is noted that the destination sampling algorithm suggested can be stated and implemented independently of the structure of the underlying graph (and thus distance function), and there is no reason to believe it wouldn't work with just about any graph. Exploring the limits of the algorithms applicability is an interesting, open problem.

References

[1] A.L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286:509512, 1999.

[2] L. Barriere, P. Fraigniaud, E. Kranakis, and D. Krizanc. Efficient routing in networks with long range contacts. In *Proceedings of the 15th International Symposium on Distributed Computing, DISC'01*, 2001.

[3] B. Bollobas and F. Chung. The diameter of a cycle plus a random matching. *SIAM Journal on Discrete Mathematics*, 1:328–333, 1988.

[4] B. Bollobás and O. Riordan. The diameter of a scale-free random graph. *Combinatorica*, 24:5, 2004.

[5] I. Clarke, T. Hong, S. Miller, O. Sandberg, and B. Wiley. Protecting free expression online with Freenet. *IEEE Internet Computing*, 6:40–49, 2002.

[6] I. Clarke, T. Hong, O. Sandberg, and B. Wiley. Freenet: A distributed anonymous information storage and retrieval system. In *Proc. of the ICSI Workshop on Design Issues in Anonymity and Unobservability*, pages 311–320, 2000.

[7] A. Clauset and C. Moore. How do networks become navigable? *Preprint*, 2003.

[8] P. S. Dodds, M. Roby, and D. J. Watts. An experimental study of search in global social networks. *Science*, 301:827, 2003.

[9] D. Eppstein and J. Y. Wang. A steady state model for graph power laws. In *2nd International Workshop on Web Dynamics*, May 2002.

[10] J. Kleinberg. The small-world phenomenon: an algorithmic perspective. In *Proceedings of the 32nd ACM Symposium on Theory of Computing*, 2000.

[11] J. Kleinberg. Small-world phenomena and the dynamics of information. In *Advances in Neural Information Processing Systems (NIPS) 14*, 2001.

[12] J. Kleinberg. Complex networks and decentralized search algorithms. In *Proceedings of the International Congress of Mathematicians (ICM)*, 2006.

[13] D. Liben-Nowell, J. Novak, R. Kumar, P. Raghavan, and A. Tomkins. Geograph routing in social networks. In *Proceedings of the National Academy of Science*, volume 102, pages 11623–11628, 2005.

[14] G. Singh Manku, M. Bawa, and P. Raghavan. Symphony: Distributed hashing in small world. In *Proceedings of the 4th USENIX Symposium on Internet Technologies and Systems*, 2003.

[15] S. Milgram. The small world problem. *Psychology Today*, 1:61, 1961.

[16] M. Newman and D. Watts. Renormalization group analysis of the small-world network model. *Phys. Lett. A*, 263:341–346, 1999.

[17] D.J. Watts, P. Dodds, and M. Newman. Identity and search in social networks. *Science*, 296:1302–1305, 2002.

[18] D.J. Watts and S. Strogatz. Collective dynamics of small world networks. *Nature*, 393:440, 1998.

[19] H. Zhang, A. Goel, and R. Govindan. Using the small-world model to improve Freenet performance. In *Proc. IEEE Infocom*, 2002.