Almost optimal decentralized routing in long-range contact networks

Emmanuelle Lebhar[*] and Nicolas Schabanel[**]

LIP (UMR CNRS, ÉNS Lyon, INRIA, Univ. Claude Bernard Lyon I) [***]
École Normale Supérieure de Lyon, 46 allée d'Italie, 69364 Lyon Cedex 07, France.

Abstract. In order to explain the ability of individuals to find short paths to route messages to an unknown destination, based only on their own local view of a social network (the small world phenomenon), Kleinberg (2000) proposed a network model based on a d-dimensional lattice of size n augmented with k long range directed links per node. Individuals behavior is modeled by a greedy algorithm that forwards the message to the neighbor of the current holder, which is the closest to the destination. This algorithm computes paths of expected length $\Theta(\log^2 n/k)$ between any pair of nodes. Other topologies have been proposed later on to improve greedy algorithm performance. But, Aspnes et al. (2002) shows that for a wide class of long range link distributions, the expected length of the path computed by this algorithm is always $\Omega\big(\log^2 n/(k^2 \log\log n)\big)$.

We design and analyze a new decentralized routing algorithm, in which nodes consult their neighbors near by, before deciding to whom forward the message. Our algorithm uses similar amount of computational resources as Kleinberg's greedy algorithm: it is easy to implement, visits $O\big(\log^2 n/\log^2(1+k)\big)$ nodes on expectation and requires only $\Theta(\log^2 n/\log(1+k))$ bits of memory – note that [1] shows that any decentralized algorithm visits at least $\Omega(\log^2 n/k)$ on expectation. Our algorithm computes however an almost optimal path of expected length $O\big(\log n(\log\log n)^2/\log^2(1+k)\big)$, between any pair of nodes. Our algorithm might fit better some human social behaviors (such as web browsing) and may also have successful applications to peer-to-peer networks where the length of the path along which the files are downloaded, is a critical parameter of the network performance.

1 Introduction

The small world phenomenon. Since the experiment of Milgram in 1967 [2], showing that people are able to route very efficiently messages to an unknown destination through their own local acquaintances (even if only 25% of the messages actually arrived), several models [3,4] have been designed to capture this

[*] http://perso.ens-lyon.fr/emmanuelle.lebhar/.

[**] http://perso.ens-lyon.fr/nicolas.schabanel/.

[***] This works was supported by the CNRS AS Dynamo and AS Grands Graphes grants.

phenomenon. Numerous real graphs (such as the co-author graph, the web graph, peer-to-peer networks...) share similar properties: a very small diameter (typically poly-logarithmic in the size of the network) and the existence of short paths between random nodes, that can be found very efficiently, based only on the local view of the network.

Models for the small world phenomenon. Models for the small world phenomenon have recently received a renew of interest for their potential application to peer-to-peer networks [5,6]. Watts and Strogatz observed in [7] that most of the small world graphs are locally strongly interconnected and proposed a random rewiring model that yields a small diameter and strong local interconnections (see also [8,9]). But these models fail to capture the specific nature of a small world. In [10], Kleinberg demonstrated that, for these models, there does not exist any *decentralized algorithm* (i.e., using only local information) that can find poly-logarithmic length paths, even when the diameter is poly-logarithmic. He then introduced a new model, that in its most general form is a d-dimensional toric lattice augmented with k random directed links per node. The d-dimensional lattice represents the underlying geographic (or *local*) relationships between the individuals. Each node $\mathbf{u}$ is also the origin of $k \leq \log n$ directed links pointing to its k *long range contacts* $\mathbf{v}_1, \ldots, \mathbf{v}_k$, chosen randomly and independently according to the s-harmonic distribution, i.e., with probability proportional to $1/\delta(\mathbf{u}, \mathbf{v})^s$, where $\delta(\mathbf{u}, \mathbf{v})$ is the lattice (Manhattan) distance between $\mathbf{u}$ and $\mathbf{v}$. [10,11] demonstrate that when $s \neq d$, no decentralized algorithm can find a poly-logarithmic length path in the d-dimensional network. For $s = d$, a simple greedy algorithm is proposed, that forwards the message to the closest[1] neighbor of the current holder to the target until it reaches its destination. When $s = d$, this algorithm computes a path of expected length $\Theta(\log^2 n/k)$, between any random pair of nodes. This result demonstrates that there is more to the small world effect than simply the existence of short paths, and that the algorithmic nature of the experiment has to be considered. Variants of this graph, with undirected long range links, based on edge percolation, have been studied in [12,13,1].

Several topologies (e.g., [14,15]) have been proposed to improve the greedy algorithm performances, in the perspective of applications to peer-to-peer networks. [6] demonstrates that for a wide class of long range links distributions on the ring (including the one mentioned above), Kleinberg's greedy algorithm computes path of expected length $\Omega\big(\log^2 n/(k \log\log n)\big)$ (if it is not allowed to "jump over" the target, and $\Omega\big(\log^2 n/(k^2 \log\log n)\big)$ otherwise). In [1,16], the greedy router is aware of the long range contacts of the local neighbors closeby (at lattice distance ≤ 1 in [1] and $\leq \log^{1/d} n$ in [16]) before forwarding the message: the expected length of the computed path is improved to $O\big(\log^2 n/(k \log k)\big)$ in [1] (the network in [1] is also slightly different), and $\Theta(\log^{1+1/d} n)$ in [16].

Our contribution. In this paper, we design and analyze a new decentralized routing algorithm on the d-dimensional Kleinberg's small world model, that

[1] According to the lattice distance.

computes a path of expected length $O\big(\log n \cdot (\log\log n)^2/\log^2(1+k)\big)$ between any pair of nodes. Our algorithm visits $O\big((\log n/\log(1+k))^2\big)$ nodes on expectation to compute this path. The network load induced by the computation of the path and the *latency*[2] of our protocol is then very close to Kleinberg's greedy algorithm. Note that [1] proves that any decentralized routing algorithm visits at least $\Omega(\log^2 n/k)$ nodes. Our algorithm requires small resources as well: it only requires $O(\log^2 n/\log(1+k))$ bits of memory to store the addresses of $O(\log n/\log(1+k))$ nodes (for instance, in the message header); and it is fairly easy to implement. Note also that it is not based on searching for the highest degree nodes, and thus avoids overloading them. Applied to peer-to-peer networks, where the path length is a critical factor of performance (since downloaded files are often large), our algorithm could possibly reduce the load of the network.

2 Model and main results

The network. We consider the d-dimensional variant of the small world network model with $k \leq \log n$ long-range links per node, introduced by Kleinberg in [10]. The network is an augmented d-dimensional toric lattice $\{-n, \ldots, 0, \ldots, n\}^d$ of $(2n+1)^d$ nodes. In addition to its $2d$ neighbors in the lattice (its *local contacts*), each node $\mathbf{u}$ is the origin of k directed links, each of them pointing towards a node $\mathbf{v}_j$, $1 \leq j \leq k$, (*$\mathbf{u}$'s j-th long-range contact*), chosen independently according to the d-harmonic distribution, i.e., with a probability proportional to $1/\delta(\mathbf{u},\mathbf{v})^d$, where $\delta(\mathbf{u},\mathbf{v})$ is the distance between $\mathbf{u}$ and $\mathbf{v}$ on the toric lattice.

In all the following, log stands for the logarithm base 2; ln denotes the natural logarithm, base e, and $H_n = \sum_{i=1}^n 1/i$. Note that $\ln(n+1) < H_n < \ln n + 1$.

Decentralized routing algorithms. We study algorithms that compute a path to transmit a message or a file from a source to a target, along the local and (directed) long range links of the network. Following Kleinberg's definition, such an algorithm is *decentralized* if it navigates through the network using only local information to compute the path. In particular, it has the knowledge 1) of the underlying lattice structure (the d-dimensional torus), 2) of the coordinates of the target in the lattice, and 3) of the nodes it has previously visited as well as their long-range contacts. But, crucially, 4) it can *only* visit nodes that are local or long-range contacts of previously visited nodes, and 5) does not know the long-range contacts of any node that has not yet been visited. However, 6) the algorithm (but *not the path it computes*) is authorized to travel backwards along any directed links it has *already* followed. As Kleinberg pointed out in [17], this is a crucial component of human ability to find short paths: one can interpret point 6) as a web user pushing the back button, or an individual returning the letter to its previous holder (who wrote his address on the envelope before sending it).

The following theorem is the main result of this paper.

[2] defined as the time to compute the path.

Theorem 1. *For any dimension d and $k \leq \log n$, there is a decentralized routing algorithm $\mathcal{A}$ using $\Theta(\log^2 n/\log(1+k))$ bits of memory such that, for any pair of nodes $(\mathbf{s}, \mathbf{t})$, $\mathcal{A}$ computes a path from $\mathbf{s}$ to $\mathbf{t}$ of expected length $O\big(\log n \cdot (\log\log n/\log(1+k))^2\big)$, and visits $O\big((\log n/\log(1+k))^2\big)$ nodes on expectation to compute this path.*

Our algorithm computes an almost optimal path in the following sense: the expected path length is $O(\log n(\log\log n/\log(1+k))^2)$, while the diameter of Kleinberg's network is lower bounded by $\Omega(\log n/\log(1+k))$ (every node has out-degree $2d+k$). The expected path length is thus optimal up to a $(\log\log n)^2/\log(1+k)$ factor. It shows in particular that Kleinberg's greedy algorithm does not compute an optimal path, nor a constant factor approximation.

We present below the depth-first search implementation of our algorithm which is the most time-efficient. We will however analyze in the following sections an equivalent (but less time-efficient) breadth-first search implementation that improves the readability of the proofs. In order to describe the algorithm, we introduce the following definitions.

Definition 2. *We say that a link (local or long-range) from a node* $\mathbf{u}$ *to a node* $\mathbf{v}$ *is* good *if* $\mathbf{v}$ *is strictly closer to the target than* $\mathbf{u}$*, according to the lattice distance. We say then that* $\mathbf{v}$ *is a* good contact *(local or long-range) of* $\mathbf{u}$.

A node $\mathbf{v}$ *is said to* be h good links away *from* $\mathbf{u}$*, if there is a path of length* $\leq h$ *from* $\mathbf{u}$ *to* $\mathbf{v}$ *only composed of good links;* $\mathbf{v}$ *is* h local good links away *from* $\mathbf{u}$ *if this path is only composed of good local links.*

Every node $\mathbf{u}$ (but the target) is the origin of at least one good (local) link (and in fact, up to d local good links depending on the relative position of $\mathbf{u}$ to the target) and, with some probability, of up to k other good (long-range) links. We now describe the algorithm with the parameters set for the one-dimensional network (the parameters for the d-dimensional network are given Section 4 - the algorithm remains unchanged).

The algorithm (depth-first search implementation - one-dimensional network). Let $\mathbf{s}$ and $\mathbf{t}$ be respectively the source and the target. Let us assume we are given three functions $h_{\max}(x)$, $b_{\max}(x)$ and $g(x)$ whose values will be given in Section 3 and 4, for $d = 1$ and $d \geq 2$ respectively. Let $\mathbf{x}$ the current holder of the message and π the current path from $\mathbf{s}$ to $\mathbf{x}$.
While $\delta(\mathbf{x}, \mathbf{t}) > k\log^2 n$: explore in depth-first order the nodes $h_{\max}(x)$ good links away from $\mathbf{x}$, record in a set F all the good long range contacts visited, but skip in the search all the long range contacts that are at lattice distance $< h_{\max}(x) + g(x)$ from any node of the current set F. The depth-first search also stops as soon as $|F| = b_{\max}(x)$. Each time a node, $\mathbf{z}$, exactly $h_{\max}(x)$ good links away from $\mathbf{x}$ is reached, read the addresses of the long range contacts of the nodes $g(x)$ good local links away from $\mathbf{z}$ and record in a variable $\mathbf{y}$ the closest node to the target (according to the lattice distance) among the visited nodes and their contacts. At the end of the depth-first exploration, route the message

from $\mathbf{x}$ to $\mathbf{y}$ along the links followed from $\mathbf{x}$ to $\mathbf{y}$ during the exploration, and extend the path π to $\mathbf{y}$ accordingly.
Once $\delta(\mathbf{x}, \mathbf{t}) \leq k \log^2 n$: apply Kleinberg's greedy algorithm, i.e., forward the message to the closest contact of $\mathbf{x}$ to the target $\mathbf{t}$, and extend the path π accordingly, until the target is reached.

Figure 1 illustrates the structure visited during each exploration step: straight lines represent good local links and arrows represent good long range links; the nodes in F are represented by white circles, each of them starts a new chain of $\leq h_{\max}(x) + g(x)$ local links towards the target. The structure is composed of a (partial) $(1 + k)$-ary tree of height $h_{\max}(x)$ extended by chains of local links of length $g(x)$ attached to its leaves. The chains of local links, rooted on the nodes in F, are guaranteed not to overlap, since only good long range contacts far enough from any already present node in F are considered. The tree is drawn on the plane to highlight the tree structure but is in fact mapped on the ring. At the end of the exploration step, the path is extended from $\mathbf{x}$ to the closest[3] node $\mathbf{y}$ to the target, among the explored nodes and their contacts. A new exploration step then begins from $\mathbf{y}$.

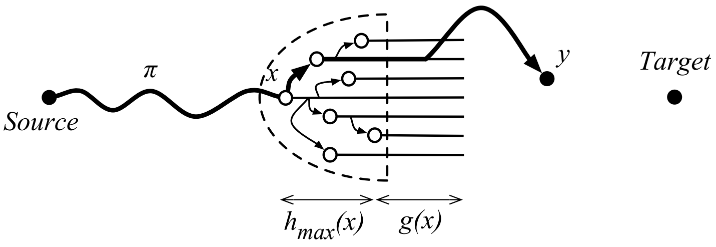

Fig. 1. Extension of the path π (in bold) at the end of an exploration step.

The following sections analyze this algorithm in detail and demonstrate the theorem: we start with the one-dimensional network (Section 3) and show in Section 4 how the results on the one-dimensional network extend to arbitrary d-dimensional networks.

3 One-dimensional network

In dimension 1, the network is an augmented ring of $2n + 1$ nodes, numbered from $-n$ to n. In addition to its two neighbors in the ring (its *local contacts*), each node $\mathbf{u}$ is the origin of k ($\leq \log n$) extra directed links, each of them pointing towards a node $\mathbf{v}_j$ ($\mathbf{u}$'s j-th long-range contact), chosen independently according to the 1-harmonic distribution, i.e., with probability $1/(2H_n\delta(\mathbf{u}, \mathbf{v}))$, where $H_n = \sum_{i=1}^{n} 1/i$. We define a *chain* as a set of locally neighboring nodes, i.e., a path of local links.

[3] According to the lattice distance.

In order to simplify the analysis of the algorithm, we use a breadth-first search implementation of the exploration step in our algorithm (below). The analysis consists in the study of the explored tree structure: basically, that this tree is large enough to guarantee the existence of a contact whose lattice distance to the target $\mathbf{t}$ is $\frac{\log(1+k)}{2k}$ times $\mathbf{x}$'s distance to $\mathbf{t}$. Since this analysis is independent of the way the tree is searched, it will apply to the depth-first search implementation as well.

Routing algorithm (breadth-first search implementation)

```
    Let $h_{\max}(x) = (\log\log x - \log\log\log n)/\log(1 + kH_x/(6H_n))$,
                        (*Note that* $h_{\max}(x) = O(\log n \log\log x/(\log(1+k)\log x))$)
        $b_{\max}(x) = \log x/\log\log n$,
        and $g(x) = \log n \log\log n/(\log(1+k)\log x)$.

    **Input:** the source $\mathbf{s}$ and the target $\mathbf{t}$.

1.  **Initialization:** $\mathbf{x} \leftarrow \mathbf{s}$.
2.  While $\delta(\mathbf{x},\mathbf{t}) > k\log^2 n$, do:
    **Exploration step:**
        $x \leftarrow \delta(\mathbf{x},\mathbf{t})$, $A_0 \leftarrow \{\mathbf{x}\}$, $B_0 \leftarrow \{\mathbf{x}\}$, $F \leftarrow \{\mathbf{x}\}$, $h \leftarrow 0$.
        While $h < h_{\max}(x)$ and $|B_h| < b_{\max}(x)$:
            $B_{h+1} \leftarrow \varnothing$.
            for each $\mathbf{u} \in B_h$ do
                $B_{h+1} \leftarrow$ the good local neighbors of $\mathbf{u}$.
                for each good long range contact $\mathbf{v}$ of $\mathbf{u}$ do
                    if $\forall \mathbf{w} \in F$, $\delta(\mathbf{v},\mathbf{w}) \geq h_{\max}(x) + g(x)$ then
                        $F \leftarrow F \cup \{\mathbf{v}\}$, $B_{h+1} \leftarrow B_{h+1} \cup \{\mathbf{v}\}$.
            $A_{h+1} \leftarrow A_h \cup B_{h+1}$.
            $h{+}{+}$.
        if $|B_h| > b_{\max}(x)$ then
            remove the $(|B_h| - b_{\max}(x))$ last inserted nodes from $B_h$ and $F$.
        $h_{\text{stop}} \leftarrow h$, $A \leftarrow A_{h_{\text{stop}}-1} \cup B_{h_{\text{stop}}}$.      (*Note that* $|B_{h_{\text{stop}}}| \leq b_{\max}(x)$)
        $C \leftarrow \bigcup_{b \in B_{h_{\text{stop}}}} C_b$, where $C_b$ is the set of the nodes that are $\leq g(x)$
            local good links away from $b$.
    **Message forward step:**
        $\mathbf{x} \leftarrow$ the closest node to the target $\mathbf{t}$, according to the lattice distance,
            among the local or long-range contacts of a node in $A \cup C$.
        Route the message to $\mathbf{x}$, along the shortest path in $A \cup C$ to $\mathbf{x}$.
3.  **Final step (*Kleinberg's greedy algorithm*):** Forward the message to
    the closest node towards the target among the (local or long-range) contacts
    of its current holder, until it reaches the target $\mathbf{t}$.
```

Figure 2 illustrates the notations used in the algorithm described in the frame above. A is the set of the nodes explored. The links followed during any exploration step map a non-overlapping $(1+k)$-ary tree structure of height $h_{\mathrm{stop}} \leq h_{\max}(x)$ on A, whose set of leaves is $B_{h_{\mathrm{stop}}}$, and extended by $|B_{h_{\mathrm{stop}}}| \leq b_{\max}(x)$ chains of length $g(x)$, rooted on the nodes in $B_{h_{\mathrm{stop}}}$, pointing towards the target. The set B_h is the set of nodes at level h in the tree structure mapped on A. A may as well be seen as a set of $|F| = |B_{h_{\mathrm{stop}}}|$ non-overlapping chains of length $\leq h_{\max}(x) + g(x)$ rooted on the nodes in F, pointing towards the target, connected one to the other by a tree structure of long range links.

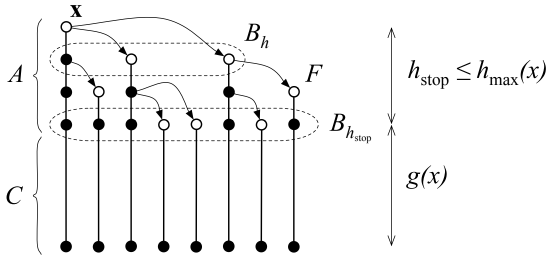

Fig. 2. The sets A, B_h, C and F (the nodes in white) during an exploration step.

Map of the proof. We will show that at the end of any exploration step, with constant probability, the message is routed to a node at lattice distance $\leq \frac{\log(1+k)}{2k}x$ from the target, where x is the lattice distance of the message at the beginning of the exploration step (Proposition 3, from which we deduce Theorem 1). In order to prove Proposition 3, we show that with constant probability, there are at least $\Omega(\frac{\log n}{\log(1+k)})$ nodes in C, whose long range contacts have not yet been explored (Lemma 7). Combined with Lemma 4, this yields Proposition 3. The proof of Lemma 7 consists in showing that the number of branches in the tree structure of A is large enough. This is ensured by lower bounding the probability that a new non-overlapping branch is created (Corollary 6), and then carefully tuning $h_{\max}(x)$ and $g(x)$ to realize a trade-off between limiting overlapping and maximizing the tree growth to minimize its height. The size of the tree is then lower bounded by the growth of a branching process, which concludes the result.

Proposition 3. *There exist two constants $p_1 > 0$ and n_0, independent of n and $\mathbf{x}$, such that, for $n \geq n_0$, at the end of any exploration step, with probability $\geq p_1$, there is a node $\mathbf{u}$ in $A \cup C$ such that $\mathbf{u}$ or one of its long-range contact is at distance $\leq \frac{\log(1+k)}{2k}x$ from the target.*

The following lemma is directly inspired from [10]; its proof is omitted.

Lemma 4. *Given $\gamma > 0$, there is a constant $p_2 > 0$, such that, for any subset Γ of $\gamma \cdot \frac{\log n}{\log(1+k)}$ vertices at lattice distance in $(\frac{\log(1+k)}{2k}x, x]$ from the target, one vertex in Γ (at least) has a long-range contact at lattice distance $\leq \frac{\log(1+k)}{2k}x$ to the target, with probability at least p_2.*

The next lemma will be used to lower bound the probability of creating a new non-overlapping branch in A.

Lemma 5. *Let* $\mathbf{u}$ *be a node at lattice distance* u *from the target* $\mathbf{t}$*,* $\mathbf{v}$ *its* j*-th long-range contact,* Q *a set of* q *forbidden nodes, and* r *an integer. The probability that* $\mathbf{v}$ *is good and at lattice distance* $\geq r$ *from any node of* Q*, is* $\geq (H_{2u-1} - H_{2rq-1})/(2H_n)$*.*

Proof. Let $\mathcal{E}$ be the event that $\mathbf{v}$ is good and is at distance $\geq r$ from any node of Q. $\mathcal{E}$ is the event that $\mathbf{v}$ is good and does not belong to the any of the q chains of nodes of length $2r$ centered on the nodes of Q. We bound the probability of $\mathcal{E}$ by noticing that the probability that $\mathbf{v}$ is at distance δ from $\mathbf{u}$ is decreasing with δ. Therefore, the probability of $\mathcal{E}$ is minimized when the nodes in the q chains are all distinct, in the interval of nodes of radius $u-1$ around the target, and as close as possible to $\mathbf{u}$, according to the lattice distance. A simple case analysis (depending on whether $u \leq n/2$, or $n/2 \leq u \leq n - rq$, or $u \geq n - rq$) shows that the probability of $\mathcal{E}$ is then greater than the probability that $\mathbf{v}$ is at distance $\geq 2rq$ from $\mathbf{u}$, and is at distance $< u$ from the target. We conclude that: $\Pr \mathcal{E} \geq \frac{1}{2H_n} \sum_{i=2rq}^{2u-1} \frac{1}{i} = \frac{H_{2u-1} - H_{2rq-1}}{2H_n}$.

Corollary 6. *There exists a constant* n_0 *independent of* x*,* n*, and* k*, such that if* $n \geq n_0$*, during any exploration step, for any unvisited node* $\mathbf{u}$ *at lattice distance* $u > \frac{\log(1+k)}{2k} x$ *from the target, the probability* α_u *that the* j*-th long range contact* $\mathbf{v}$ *of* $\mathbf{u}$*, is good and is at lattice distance* $\geq h_{\max}(x) + g(x)$ *from any node in* F*, is greater than* $H_x/(6H_n) =_{\text{def}} \alpha^-$*.*

Proof. F contains less than $b_{\max}(x)$ nodes. By Lemma 5, $\alpha_u \geq (H_{2u-1} - H_{2b_{\max}(x)(h_{\max}(x)+g(x))-1})/2H_n$. But $h_{\max}(x) \leq \frac{6H_n}{H_x} \frac{\log \log x}{\log(1+k)} \leq 6g(x)$ and $b_{\max}(x) g(x) = \log n$, thus: $\alpha_u \geq \frac{H_{2u-1} - H_{16 \log n}}{2H_n} \geq \ln(\frac{2u-1}{16 \log n})/(2H_n)$. Since $x > k \log^2 n$ and $k \leq \log n$, we have $\frac{x}{k} < x^{1/3} \log n$ and $\ln\left(\frac{2u-1}{16 \log n}\right) > \ln\left(\frac{\log(1+k) \cdot x/k - 1}{16 \log n}\right) > H_{x^{1/3}} \geq \frac{1}{3} H_x$, for $n \geq n_0$, for some constant n_0 independent of x, n, and k. We conclude that $\alpha_u \geq H_x/(6H_n)$.

The following lemma shows that at the end of any exploration step, with constant probability, either we have already reached a node in $A \cup C$ at distance $\leq \frac{\log(1+k)}{2k} x$ from the target, or the tree is wide enough to contain the required number of nodes to apply Lemma 4.

Lemma 7. *There exists a constant* $p_3 > 0$*, independent of* n *and* x*, such that, at the end of any exploration step, with probability at least* p_3*, either there exists a node in* A *at lattice distance* $\leq \frac{\log(1+k)}{2k} x$ *from the target, or* $|B_{h_{\text{stop}}}| \geq \log x/(2 \log \log n)$*.*

Proof. Let $\mathcal{E}$ the event that at the end of the exploration step, there exists a node in A at lattice distance $\leq \frac{\log(1+k)}{2k} x$ from the target, or $|B_{h_{\text{stop}}}| \geq \log x/(2 \log \log n)$.

Let $Z = \{\mathbf{z} : \delta(\mathbf{z},\mathbf{t}) > \frac{\log(1+k)}{2k} x\}$ and $\bar{Z}$ its complementary set. By Corollary 6, during any exploration step, for every unvisited node $\mathbf{u} \in Z$, for all $1 \leq j \leq k$, the probability that the j-th long range contact of $\mathbf{u}$ is good and at lattice distance $\geq h_{\max}(x) + g(x)$ of any node in the current F, is at least α^-. Thus, as long as nodes in Z are considered, each of their long range contact will be added to B_{h+1} with probability $\geq \alpha^-$. As soon as a node $\mathbf{u}$ from $\bar{Z}$ is inserted in A_h, for some h, the probability that, for a given j, its j-th long range contact is good and at lattice distance $\geq h_{\max}(x) + g(x)$ of any node in the current F, is no longer lower bounded by α^-; but the event $\mathcal{E}$ is verified. We use a probabilistic coupling argument to lower bound the probability of $\mathcal{E}$, by virtually running the exploration step on a gadget network, constructed from the original network as follows: this gadget network has the same underlying lattice; the nodes in Z have the exact same links as in the original network; but we consider a virtual link distribution for the nodes of $\bar{Z}$ such that for every unvisited node $\mathbf{u}$, the probability that its j-th long range contact is good and at lattice distance $\geq h_{\max}(x) + g(x)$ from any set of nodes G of size $\leq b_{\max}(x)$, is α^- (note that this distribution does not need to exist effectively). We run the exploration step on this gadget network from the same $\mathbf{x}$ as in the real network, except that we don't interrupt it until $h = h_{\max}(x)$. It yields three sets families (A'_h), (B'_h) and F', such that: $A_h \cap Z = A'_h \cap Z$, $B_h \cap Z = B'_h \cap Z$, and $F \cap Z = F' \cap Z$, for all $1 \leq h \leq h_{\text{stop}}$. The links followed during the exploration of the gadget network define a non-overlapping tree structure of height exactly $h_{\max}(x)$ on $A' = \cup_h A'_h$ where B'_h is the set of the nodes at level h. Let $\mathcal{E}'$ be the event that $|B'_{h_{\max}(x)}| \geq \log x/(2\log\log n)$. We now show that $\Pr\{\mathcal{E}\} \geq \Pr\{\mathcal{E}'\}$:

- If, in the original network, $A \cap \bar{Z} = \varnothing$, then $B'_{h_{\text{stop}}} = B_{h_{\text{stop}}}$. If $h_{\text{stop}} < h_{\max}(x)$, then $|B'_{h_{\max}(x)}| \geq |B'_{h_{\text{stop}}}| = |B_{h_{\text{stop}}}| = b_{\max}(x) = \log x/\log\log n$, and then $\mathcal{E}$ and $\mathcal{E}'$ are both verified. If $h_{\text{stop}} = h_{\max}$, $B_{h_{\text{stop}}} = B'_{h_{\max}(x)}$ and then $\mathcal{E}$ and $\mathcal{E}'$ are equivalent. Then, whatever the gadget network is inside $\bar{Z}$, $\Pr\{\mathcal{E}|\ A \cap \bar{Z} = \varnothing\} = \Pr\{\mathcal{E}'|A \cap \bar{Z} = \varnothing\}$.
- If, in the original network, $A \cap \bar{Z} \neq \varnothing$, then $\mathcal{E}$ is verified, so, whatever the gadget network is inside $\bar{Z}$, $\Pr\{\mathcal{E}|A \cap \bar{Z} \neq \varnothing\} = 1 \geq \Pr\{\mathcal{E}'|A \cap \bar{Z} \neq \varnothing\}$.

We now lower bound $\Pr\{\mathcal{E}'\}$. The set $A' = \cup_h A'_h$ is structured as a random tree of root $\mathbf{x}$, in which every node $\mathbf{u}$ at level h has, independently, a random number $1 + l$ of children (one local contact and l long range contacts), where l is given by a binomial law of parameters $(k, \alpha_{\mathbf{u}})$, with $\alpha_{\mathbf{u}} \geq \alpha^-$. Thus the number of nodes at level h, $|B'_h|$, stochastically dominates the random variable[4] b_h for the number of nodes at level h in the following branching process: start with one node; at step h, each node at level $h-1$ is given, independently, exactly $1 + l$ children, with probability $\rho_l = \binom{k}{l}(\alpha^-)^l(1-\alpha^-)^{k-l}$, where $0 \leq l \leq k$. Bounding the variance of b_h (omitted), gives: $\mathbb{E}[b_h] = (1 + k\alpha^-)^h$ and a constant $p_3 > 0$, independent of α^- and h, such that, with probability at least p_3, $b_h \geq \mathbb{E}[b_h]/2$. Then, since $(1 + k\alpha^-)^{h_{\max}(x)} = \log x/\log\log n$, $\Pr\{\mathcal{E}\} \geq \Pr\{\mathcal{E}'\} = \Pr\{|B'_{h_{\max}(x)}| \geq \log x/(2\log\log n)\} \geq p_3$.

[4] i.e., for all z, $\Pr\{|B'_h| \geq z\} \geq \Pr\{b_h \geq z\}$.

Corollary 8. *For $n \geq n_0$, at the end of any exploration step, with probability at least p_3, there is a node in A at lattice distance $\leq \frac{\log(1+k)}{2k}x$ from the target or there are more than $\frac{\log n}{2\log(1+k)}$ distinct nodes in C (where p_3 is given by Lemma 7).*

Combined with Lemma 4, Corollary 8 yields Proposition 3.

Proof. (of Theorem 1) W.l.o.g., the target is $\mathbf{0}$ and the source $\mathbf{s}$ is at lattice distance s from $\mathbf{0}$. Let $\mathbf{x}$ denote the current message holder of the message and x its lattice distance from the target. First recall that at the end of each exploration step, the algorithm selects the closest node to the target among the local and long-range contacts of $A \cup C$, and that the set $A \cup C$ grows towards the target; therefore, every exploration step visits unexplored nodes, and each exploration step is independent of the previous ones.

Let T and U be the solutions to $(\frac{2k}{\log(1+k)})^T = s$ and $(\frac{2k}{\log(1+k)})^U = k\log^2 n$. Note that $T \sim \log s/\log(1+k)$ and $U \sim (2\log\log n + \log k)/\log(1+k)$. We decompose the execution of $\mathcal{A}$ in T phases. The execution is in phase i, $0 \leq i \leq T$, as long as $(\frac{2k}{\log(1+k)})^{i-1} < x \leq (\frac{2k}{\log(1+k)})^i$. We say that an exploration step in phase i *succeeds* if it leads to a phase $\leq i-1$. Let Y_i and Z_i be respectively the random variables for the number of visited nodes in phase i, and for the length of the path along which the message is routed in phase i.

Suppose that we are in phase i, with $T \geq i > U$, then $x > k\log^2 n$. According to Proposition 3, each exploration step succeeds with probability $\geq p_1$. Each exploration step visits $\leq (h_{\max}(x) + g(x))\ b_{\max}(x) \leq 7g(x)b_{\max}(x)$ nodes, and routes the message, along a path of length $\leq h_{\max}(x) + g(x) \leq 7g(x)$ towards the target. Then, $\mathbb{E}[Y_i] \leq 7g(x)b_{\max}(x)/p_1 \leq \frac{7}{p_1}\frac{\log n}{\log(1+k)}$ and $\mathbb{E}[Z_i] \leq 7g(x)/p_1 \leq \frac{7}{p_1}\frac{\log n \log\log n}{i\log^2(1+k)}$, since $\log x \geq i\log(1+k)$.

Once we reach a phase $i \leq U$, we have $x \leq k\log^2 n$ and the algorithm runs Kleinberg's greedy algorithm. From [10], we know that this greedy computes a path of expected length $\leq A(\log n \log x)/k \leq 3A(\log n\log\log n)/k$ while visiting $\leq 3A(\log n \log\log n)/k$ nodes on expectation, for some constant A.

The expected length of the path from $\mathbf{s}$ to $\mathbf{0}$ computed by our algorithm is bounded by:

$$\sum_{i=0}^{T}\mathbb{E}[Z_i] \leq 3A\ \frac{\log n \log\log n}{k} + \frac{7}{p_1}\frac{\log n\log\log n}{\log^2(1+k)}\sum_{U<i\leq T}\frac{1}{i}$$
$$= O\Bigg(\log n\left(\frac{\log\log n}{\log(1+k)}\right)^2\Bigg).$$

And the expected number of nodes visited by our algorithm is bounded by:

$$\sum_{i=0}^{T}\mathbb{E}[Y_i] = O\Bigg(\left(\frac{\log n}{\log(1+k)}\right)^2\Bigg).$$

For the last of each exploration step, our algorithm just needs $\Theta\big(\log n \cdot (b_{\max}(x) + h_{\max}(x) + g(x))\big) = O(\log^2 n/\log(1+k))$ bits of memory. Indeed, each node address requires $\log n$ bits, and each exploration step needs only to store: the address of the target, the address of the nodes in F (whose size is $\leq b_{\max}(x) = O(\log n / \log\log n) = O(\log n / \log(1+k))$), the state of the stack during the depth-first search of $A \cup C$ (whose height is bounded by $h_{\max}(x) + g(x) = O(\log n / \log(1+k))$), and both the address and the state of the stack for the current best node $\mathbf{y}$ among $A \cup C$ and $A \cup C$'s contacts.

4 *d*-dimensional network

In a d-dimensional network, $d > 1$, the underlying lattice is a d-dimensional torus $\{-n, \ldots, n\}^d$. Each node $\mathbf{u}$ has k extra directed links (its long range links) each one pointing towards a node $\mathbf{v}$ chosen independently according to the d-harmonic distribution, i.e., with probability proportional to $1/\delta(\mathbf{u}, \mathbf{v})^d$.

We denote by $\mathcal{S}(\mathbf{u}, r)$ and $\mathcal{B}(\mathbf{u}, r)$, respectively the ℓ_1-sphere and ℓ_1-ball centered on $\mathbf{u}$ and of radius r. We denote by $S(r)$ and $V(r)$ their respective cardinality. Clearly, for $r \leq n$, $S(r) = \Theta(r^{d-1})$ and $V(r) = \Theta(r^d)$. More precisely, for $r \leq n$, $S(r) = \frac{2^d}{(d-1)!} r^{d-1} + \xi(r)$, and $V(r) = \frac{2^d}{d!} r^d + \eta(r)$, where $\xi(r)$ and $\eta(r)$ are positive polynomials of respective degree $d-2$ and $d-1$. These expressions are upper bounds on $S(r)$ and $V(r)$ when $r > n$.

The algorithm on a d-dimensional network. We only need to adapt the parameters of the one-dimensional routing algorithm, as follows, and everything else in the algorithm is unchanged: $b_{\max}(x)$ and $g(x)$ are unchanged; $h_{\max}(x)$ is now set to $h_{\max}(x) = (\log\log x - \log\log\log n) / \log(1 + \frac{kH_x}{2^d(2^d+2)H_n})$; and, the exploration phases now stop as soon as $x \leq k \log^{2^d+1} n$ (the while condition Item 2), and then the algorithm runs Kleinberg's greedy algorithm.

Sketch of the analysis of the algorithm on a d-network. The analysis of the algorithm is exactly identical to the one-dimensional case. Only the lower bound on the probability of creating a new branch, α^-, in the tree structure $A \cup C$ has to be evaluated in order to get the result. Lemma 10 shows that for our choice of the parameters $b_{\max}(x)$, $g(x)$, $h_{\max}(x)$ and the while condition in Item 2, this probability is again $\Theta(H_x/H_n)$, as in dimension 1, from which we get the result.

The next lemmas correspond to Lemma 5 and Corollary 6 in dimension 1; their proofs rely on the geometry of the balls in dimension d, and are omitted.

Lemma 9. *Let $1 \leq j \leq k$, $\mathbf{u}$ a node at distance $u \in (\frac{\log(1+k)}{2k} x, x]$ from the target, $\mathbf{v}$ its j-th long-range contact, Q a set of q forbidden nodes, and r an integer. The probability that $\mathbf{v}$ is good and at distance $\geq r$ from any node in Q, is $\geq \frac{H_u - 2^d H_{rq^{1/d}} - c_1}{2^d H_n + c_2}$, for two constants $c_1, c_2 \geq 0$, that only depend on d.*

Lemma 10. *There exists a constant n_1, independent of x, n, and k, such that if $n \geq n_1$, during any exploration step, for any unvisited node $\mathbf{u}$ at lattice distance $u > \frac{\log(1+k)}{2k} x$ from the target, the probability $\alpha_{u,d}$ that the j-th long range contact*

$\mathbf{v}$ *of* $\mathbf{u}$*, is good and is at lattice distance* $\geq h_{\max}(x) + g(x)$ *from any node in* F*, is greater than* $\frac{H_x}{2^d(2^d+2)H_n} =_{\mathrm{def}} \alpha_d^-$.

The properties of the tree structure on $A \cup C$ are then similar to dimension 1. Lemma 7, Property 3 and Theorem 1 follow then for any dimension $d \geq 1$.

Conclusion. Our algorithm could possibly have interesting applications in peer-to-peer networks, since its latency is comparable to Kleinberg's greedy algorithm and since it computes almost optimal paths based only on local information. Note that, if we get a bound on the expected path length between random pair of nodes, the question of the exact Kleinberg's network diameter remains open.

References

1. Manku, G.S., Naor, M., Wieder, U.: Know thy neighbor's neighbor: The power of lookahead in randomized P2P networks. To appear in Proc. of 36th ACM STOC 2004 (2004)
2. Milgram, S.: The small world problem. Psychology Today **61** (1967)
3. Newman, M.E.J.: Models of the small world. J. Stat. Phys. **101** (2000)
4. Newman, M.E.J.: The structure and function of complex networks. SIAM Review **45** (2003) 167–256
5. Zhang, H., Goel, A., Govindan, R.: Using the small-world model to improve Freenet performance. Proceedings of IEEE INFOCOM (2002)
6. Aspnes, J., Diamadia, Z., Shah, G.: Fault-tolerant routing in peer-to-peer systems. In: Proc. of ACM 3st Symp. on Princ. of Distr. Comp. (PODC 2002). Volume 31. (2002) 223–232
7. Watts, D., Strogatz, S.: Collective dynamics of small-world networks. Nature **393** (1998)
8. Newman, M.E.J., Moore, C., Watts, D.J.: Mean-field solution of the small-world network model. Phys. Rev. Lett. **84** (2000) 3201–3204
9. Bollobás, B., Chung, F.: The diameter of a cycle plus random matching. SIAM J. Discrete Math. **1** (1988) 328–333
10. Kleinberg, J.: The small-world phenomenon: an algorithmic perspective. Proc. 32nd ACM Symposium on Theory of Computing (2000) 163–170
11. Barrière, L., Fraigniaud, P., Kranakis, E., Krizanc, D.: Efficient routing in networks with long range contacts. LNCS Proc. of 15th International Symposium on Distributed Computing (DISC '01) **2180** (2001) 270–284
12. Benjamini, I., Berger, N.: The diameter of long-range percolation clusters on finite cycles. Random Structures and Algorithms **19** (2001) 102–111
13. Coppersmith, D., Gamarnik, D., Sviridenko, M.: The diameter of a long range percolation graph. Random Structures and Algorithms **21** (2002) 1–13
14. Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishnan, H.: Chord: a scalable peer-to-peer lookup service for internet applications. In: Proc. of ACM SIGCOMM 2001. (2001)
15. Manku, G.S., Bawa, M., Raghavan, P.: Symphony: Distributed hashing in a small world. In: Proc. of 4th Usenix Symp. on Internet tech. and syst. (2003)
16. Fraigniaud, P., Gavoille, C., Paul, C.: Eclecticism shrinks the world. Technical Report LRI-1376, University Paris-Sud (2003)
17. Kleinberg, J.: Small-world phenomena and the dynamics of information. Advances in Neural Information Processing Systems, MIT Press. **14** (2002)