Available online at www.sciencedirect.com
SCIENCE DIRECT®

Theoretical Computer Science 348 (2005) 294 – 310

Theoretical Computer Science

www.elsevier.com/locate/tcs

Close to optimal decentralized routing in long-range contact networks[☆]

Emmanuelle Lebhar*, Nicolas Schabanel

LIP (UMR CNRS, ÉNS Lyon, INRIA, University 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 [The small-world phenomenon: an algorithmic perspective, Proc. 32nd ACM Symp. on Theory of Computing, 2000, pp. 163–170] 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, given a source and destination, forwards a 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. [Fault-tolerant routing in peer-to-peer systems, in: Proc. of ACM 3st Symp. on Princ. of Distr. Comp. (PODC 2002), Vol. 31, 2002, pp. 223–232] shows that for a wide class of long-range link distributions, the expected length of the path computed by this algorithm is always $\Omega(\log^2 n/(k^2 \log\log n))$.

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 $O(\log^2 n/\log^2(1+k))$ nodes on expectation and requires only $\Theta(\log^2 n/\log(1+k))$ bits of memory—note that [G.S. Manku, M. Naor, U. Wieder, Know thy neighbor's neighbor: the power of lookahead in randomized P2P networks, in: Proc. of 36th ACM STOC 2004, 2004, to appear], shows that any decentralized algorithm visits at least $\Omega(\log^2 n/k)$ on expectation. Our algorithm computes however a path of expected length $O(\log n(\log\log n)^2/\log^2(1+k))$ 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.

Keywords: Small world; Random graph model; Routing algorithm

1. Introduction

The small world phenomenon: Since the experiment of Milgram in 1967 [13], 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 have been designed to capture this phenomenon (see [14,15] for a state of

[☆] This work was supported by the CNRS AS Dynamo and AS Grands Graphes grants.

* Corresponding author. Tel.: +33 4 72 72 85 04.

E-mail address: emmanuelle.lebhar@ens-lyon.fr (E. Lebhar)

URLs: http://perso.ens-lyon.fr/emmanuelle.lebhar/, http://perso.ens-lyon.fr/nicolas.schabanel/.

0304-3975/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2005.09.019

the art). Numerous real graphs (such as the co-author graph, the web graph, peer-to-peer networks, etc.) 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 [19,1]. Watts and Strogatz observed in [18] 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 [16,4]). But these models fail to capture the specific nature of a small world. Indeed, in [8], 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, (i.e., of poly-logarithmic length in the size of the network), 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 directed links ($1 \leqslant k \leqslant \log n$) 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 $\mathbf{v}$. In [8,2], it is proved 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 neighbor of the current holder that is the closest [1] 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 [3,5,11].

Several topologies (e.g., [17,10]) have been proposed to improve the greedy algorithm performances, in the perspective of applications to peer-to-peer networks. [1] 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 $\Omega(\log^2 n/(k \log\log n))$ (if it is not allowed to "jump over" the target, and $\Omega\big(\log^2 n/(k^2 \log\log n)\big)$ otherwise). In [11,6,12], the greedy router is aware of the long-range contacts of the local neighbors closeby (at Manhattan distance $\leqslant 1$ in [11], $\leqslant \log^{1/d} n$ in [6,12]) before forwarding the message: the expected length of the computed path is improved to $\mathrm{O}\big(\log^2 n/(k \log k)\big)$ in [11] (the network in [11] is also slightly different), $\Theta(\log^{1+1/d} n)$ in [6,12]. In all the following, we consider the ℓ^1-norm, also known as the Manhattan distance.

Our contribution: In this paper, we design and analyze a new decentralized routing algorithm on the d-dimensional Kleinberg's small world model, that computes a path of expected length $\mathrm{O}\big(\log n \cdot (\log\log n)^2/\log^2(1+k)\big)$ between any pair of nodes. Our algorithm visits $\mathrm{O}\big((\log n/\log(1+k))^2\big)$ 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 [11] 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 $\mathrm{O}(\log^2 n/\log(1+k))$ bits of memory to store the addresses of $\mathrm{O}(\log n/\log(1+k))$ 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 $k \leqslant \log n$ long-range links per node, introduced by Kleinberg in [8]. 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 $\mathbf{v}_j$, $1 \leqslant j \leqslant k$, (*$\mathbf{u}$'s jth long-range contact*), chosen independently according to the d-harmonic distribution, i.e., with a probability proportional to $1/\delta(\mathbf{u},\mathbf{v}_j)^d$, where $\delta(\mathbf{u},\mathbf{v}_j)$ is the distance between $\mathbf{u}$ and $\mathbf{v}_j$ on the toric lattice.

[1] According to the Manhattan distance.

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

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. Precisely, the algorithm:

1. has the knowledge of the underlying lattice structure, of the coordinates of the target in the lattice, and of the nodes it has previously visited as well as their long-range contacts;
2. can *only* visit nodes that are local or long-range contacts of previously visited nodes and does not know the long-range contacts of any node that has not yet been visited;
3. is authorized to travel backwards along any directed links it has *already* followed (but the path computed cannot).

As Kleinberg pointed out in [9], this is a crucial component of human ability to find short paths: one can interpret point 3 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.

The following theorem is the main result of this paper.

Theorem 1. *For any dimension d and $1 \leqslant k \leqslant \log n$, 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 $\mathrm{O}\big(\log n \cdot (\log\log n/\log(1+k))^2\big)$, and on expectation it visits $\mathrm{O}\big((\log n/\log(1+k))^2\big)$ nodes to compute this path.*

Our algorithm computes an almost optimal path in the following sense: the expected path length is $\mathrm{O}\big(\log n(\log\log n/\log(1+k))^2\big)$, while the diameter of Kleinberg's network is lower bounded by $\Omega\big(\log n/\log(2d+k)\big)$ (every node has constant out-degree $2d+k$). Later on, [12] has shown that the diameter is indeed $\Theta(\log n)$ w.h.p. when $k = \Theta(1)$. The expected path length is thus optimal up to a $(\log\log n)^2/\log(1+k)$ factor. Our result 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 breadth-first search implementation that improves the readability of the proofs, while being less time-efficient. In order to describe the algorithm, we introduce the following definitions.

Definition 1. We say that a link (local or long-range) from a node $\mathbf{u}$ to a node $\mathbf{v}$ is it good if $\mathbf{v}$ is strictly closer to the target than $\mathbf{u}$, according to the Manhattan 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 $\leqslant h$ 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.

Principle of the algorithm: In order to improve readability, assume here that $d = 1$. Starting from the source, our algorithm explores the local and long-range links that point towards the target, and constructs a $(1+k)$-ary tree, where one child of a given node is its good local neighbor and the (up to) k others are its (up to) k good long-range contacts. Basically, we will show that when $\Theta(\log n/\log(1+k))$ are reached in this structure, we are guaranteed that with constant probability, one visited node has a contact at least $\Theta(2k/\log(1+k))$ closer to the target. Unfortunately, due to overlapping, this structure collapses prematurely, before reaching the $\Theta(\log n/\log(1+k))$ required nodes. We get around this problem by interrupting the growth of the tree towards the target, at a given height, $h_{\max}^{d}(x)$ (a function of the current distance x to the target) and by extending the resulting leaves by suitably long chains of local contacts pointing towards the target. We are then able to visit the required number of nodes, that will allow us to get much closer to the target, while avoiding overlapping. The path from the source to the target, is then extended by the shortest path in the tree from the root to the closest contact to the target of a visited node. The process restarts then from there, until we get so close to the target that long-range links are not useful anymore. We then apply Kleinberg's greedy algorithm to finally reach the target. The tight balance between the $(1+k)$-ary part of the tree (at the top) and the linear part of tree (the chains at the bottom), allows us to reduce the expected total length path to $\mathrm{O}(\log n(\log\log n/\log(1+k))^2)$. We now describe more precisely the parameters of the algorithm.

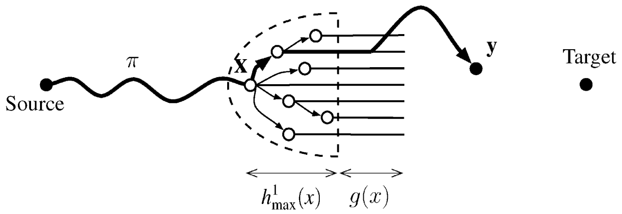

Fig. 1. Extension of the path π (in bold) at the end of an exploration step rooted in **x** at distance x to the target.

Description of the algorithm (*depth-first search implementation*): Let **s** and **t** be, respectively, the source and the target. Assume we are given three functions $h^d_{\max}(x)$, $b_{\max}(x)$ and $g(x)$, and a parameter stop_d, whose values will be given in Sections 3 and 4, for $d = 1$ and $d \geqslant 2$, respectively (only $h^d_{\max}(x)$ and stop_d depend on the dimension d). Let **x** denote the current holder of the message, x the Manhattan distance from **x** to **t**, and π the current path from **s** to **x**. The description of the algorithm is done in dimension 1 for simplicity, but it remains identical in dimension d up to the adaptation of the parameters.

(1) While $\delta(\mathbf{x}, \mathbf{t}) > \text{stop}_1$: explore in depth-first order the nodes $h^1_{\max}(x)$ good links away from **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 Manhattan distance $< h^1_{\max}(x) + g(x)$ 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, **z**, exactly $h^1_{\max}(x)$ good links away from **x** is reached, read the addresses of the long-range contacts of the nodes $g(x)$ good local links away from **z** and record in a variable **y** the closest node to the target (according to the Manhattan distance) among the visited nodes and their contacts. At the end of the depth-first exploration, route the message from **x** to **y** along the links followed from **x** to **y** during the exploration, and extend the path π to **y** accordingly.

(2) Once $\delta(\mathbf{x}, \mathbf{t}) \leqslant \text{stop}_1$, apply Kleinberg's greedy algorithm, i.e., forward the message to the closest contact of **x** to the target **t**, and extend the path π accordingly, until the target is reached.

Fig. 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 $\leqslant h^1_{\max}(x) + g(x)$ local links towards the target. The structure is composed of a partial $(1+k)$-ary tree of height $h^1_{\max}(x)$ 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 plan to highlight the tree structure but is in fact mapped on the torus. At the end of the exploration step, the path is extended from **x** to the closest[3] node **y** to the target, among the explored nodes and their contacts. A new exploration step then begins from **y**.

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 **u** is the origin of k extra directed links ($1 \leqslant k \leqslant \log n$), each of them pointing towards a node $\mathbf{v}_j$ (**u**'s jth long-range contact), chosen independently according to the 1-harmonic distribution, i.e., with probability $1/(2H_n\delta(\mathbf{u}, \mathbf{v}_j))$, where $H_n = \sum_{i=1}^{n} 1/i$.

Definition 2. We call *chain*, a set of locally neighboring nodes, i.e., a path of local links.

[3] According to the Manhattan distance.

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

Algorithm 1. Routing algorithm (breadth-first search implementation)

```
Let $h^1_{\max}(x) = \dfrac{\log\log x - \log\log\log n}{\log\left(1 + k\frac{\log x}{6\log n}\right)}$,    (Note that $h^1_{\max}(x) = \mathrm{O}(\log n \log\log x/(\log(1+k)\log x))$)
    $b_{\max}(x) = \dfrac{\log x}{\log\log n}$,   $g(x) = \dfrac{\log n \log\log n}{\log(1+k)\log x}$,   and   $\mathrm{stop}_1 = k\log^2 n$.
Input: the source $\mathbf{s}$ and the target $\mathbf{t}$.
Initialization: $\mathbf{x} \leftarrow \mathbf{s}$.
while $\delta(\mathbf{x},\mathbf{t}) > \mathrm{stop}_1$ 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 $h < h^1_{\max}(x)$ and $|B_h| < b_{\max}(x)$ do
       $B_{h+1} \leftarrow \varnothing$.
       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 $\forall \mathbf{w} \in F,\ \delta(\mathbf{v},\mathbf{w}) \geqslant h^1_{\max}(x) + g(x)$ then
                $F \leftarrow F \cup \{\mathbf{v}\}$, $B_{h+1} \leftarrow B_{h+1} \cup \{\mathbf{v}\}$.
             end if
          end for
       end for
    end while
    $A_{h+1} \leftarrow A_h \cup B_{h+1}$.
    $h$++.
    if $|B_h| > b_{\max}(x)$ then
       remove the $(|B_h| - b_{\max}(x))$ last inserted nodes from $B_h$ and $F$.
    end if
    $h_{\mathrm{stop}} \leftarrow h$, $A \leftarrow A_{h_{\mathrm{stop}}-1} \cup B_{h_{\mathrm{stop}}}$.    (Note that $|B_{h_{\mathrm{stop}}}| \leqslant b_{\max}(x)$)
    $C \leftarrow \bigcup_{b \in B_{h_{\mathrm{stop}}}} C_b$, where $C_b$ is the set of the nodes that are $\leqslant g(x)$ local good links away from $b$. otherwise.
 Message forward step:
    $\mathbf{x} \leftarrow$ the closest node to the target $\mathbf{t}$, according to the Manhattan 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 $\mathbf{x}$.
end while
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 $\mathbf{t}$.
```

Fig. 2 illustrates the notations used in Algorithm 1. 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 $h_{\mathrm{stop}} \leqslant h^1_{\max}(x)$ on A, whose set of leaves is $B_{h_{\mathrm{stop}}}$, and extended by $|B_{h_{\mathrm{stop}}}| \leqslant b_{\max}(x)$ 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 $\leqslant h^1_{\max}(x) + g(x)$ rooted on the nodes in F, pointing towards the target, connected one to the other by a long-range link.

Proof outline: In order to prove Theorem 1, we will show that at the end of any exploration step, with constant probability, the message is routed to a node at Manhattan distance $\leqslant \frac{\log(1+k)}{2k}x$ to the target, where x is the Manhattan distance of the message at the beginning of the exploration step (Proposition 10). In order to prove Proposition 10,

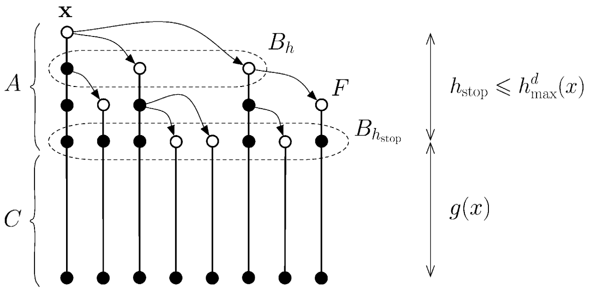

Fig. 2. The sets A, B_h, C and F (the nodes in white) during an exploration step.

we show that with constant probability, there are at least $\Omega\left(\frac{\log n}{\log(1+k)}\right)$ nodes in C, whose long-range contacts have not yet been explored (Lemma 8). Combined with Lemma 5, this yields Proposition 10. The proof of Lemma 8 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 7), and then carefully tuning $h^1_{\max}(x)$ 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.

Section 3.1 presents pure probalistic lemmas that will be used in Section 3.2 to analyze the length of the path computed by our routing algorithm.

3.1. Probabilistic lemmas

In this section, we analyze the growth of a branching process that will be used in the next subsection to lower bound the number of nodes that are visited during the exploration step in our algorithm.

Let $0 < \alpha \leqslant 1$ and $\rho_l = \binom{k}{l}\alpha^l(1-\alpha)^{k-l}$, for $0 \leqslant l \leqslant k$. Consider the following branching process: start with one node; during the hth step, every node at level $h-1$, is given independently, with probability ρ_l, $1+l$ children ($0 \leqslant l \leqslant k$). Note that, ρ_l is exactly the probability to get exactly l children among k potential children that each exists independently with probability α. The two following lemmas show that the number of branches at level h in the resulting partial $(1+k)$-ary tree, is above half of its expectation with constant probability. These probabilistic lemmas are the key to Lemma 8.

Let b_h be the random variable for the number of branches at level h (that is to say for the number of nodes at level h in the tree). Lemma 2 is a standard Galton–Watson process analysis.

Lemma 2. $\mathbb{E}[b_h] = (1+k\alpha)^h$ *and* $\mathbb{E}[(b_h)^2] \leqslant \left(1+\frac{1-\alpha}{1+k\alpha}\right)(1+k\alpha)^{2h}$.

Proof. We proceed by induction. For $h = 0$, $\mathbb{E}[b_0] = 1$ and $\mathbb{E}[b_0] = 1 \leqslant 1 + \frac{1-\alpha}{1+k\alpha}$. Suppose now $h > 0$. The expected number of children of every node at level $h-1$, is $(1+k\alpha)$. The expected number of nodes at level h, is thus $(1+k\alpha)\mathbb{E}[b_{h-1}]$. Therefore, $\mathbb{E}[b_h] = (1+k\alpha)^h$.

We now evaluate the expectation of $(b_h)^2$. With probability ρ_l, the root of branching process has exactly $1+l$ children; if we denote by $\beta_0, \beta_1, \ldots, \beta_l$ the random variables for the number of branches at level $h-1$ in each of the $1+l$ subtrees of the root, we have: $(b_h)^2 = (\sum_{i=0}^{l}\beta_i)^2 = \sum_{i=0}^{l}(\beta_i)^2 + 2\sum_{i=0}^{l}\sum_{j=i+1}^{l}\beta_i\beta_j$. The β_is are independent variables, distributed identically as b_{h-1}. Thus

$$\mathbb{E}[b_h^2| \text{ the root has exactly } l \text{ children}] = (l+1)\mathbb{E}[(b_{h-1})^2] + l(l+1)\mathbb{E}[b_{h-1}]^2.$$

Summing over l yields

$$\begin{aligned}\mathbb{E}[(b_h)^2] &= \sum_{l=0}^{k}\binom{k}{l}\alpha^l(1-\alpha)^{k-l}\left((l+1)\mathbb{E}[(b_{h-1})^2] + l(l+1)\mathbb{E}[b_{h-1}]^2\right)\\ &= (1+k\alpha)\mathbb{E}[(b_{h-1})^2] + k\alpha(2+\alpha(k-1))\mathbb{E}[b_{h-1}]^2.\end{aligned}$$

Since, by induction, $\mathbb{E}[(b_{h-1})^2] \leqslant \left(1 + \frac{1-\alpha}{1+k\alpha}\right)(1+k\alpha)^{2h-2}$, and $\mathbb{E}[b_{h-1}] = (1+k\alpha)^{h-1}$, we conclude that

$$\mathbb{E}[(b_h)^2] \leqslant (1+k\alpha)^{2h-2}(1+2k\alpha+k^2\alpha^2+1-\alpha) = \left(1+\tfrac{1-\alpha}{1+k\alpha}\right)(1+k\alpha)^{2h}. \qquad \square$$

Let X be a real valued random variable. Let $X^+ = \max(X,0)$ and $X^- = \max(-X,0)$, so that $X = X^+ - X^-$. Interestingly enough, the second moment method introduced in [7, Remark 3.1], still holds if X takes negative values, under some conditions.

Lemma 3 (*Second moment method*). *If* $\mathbb{E}[X^-] \leqslant 2\mathbb{E}[X^+]$, *then* $\Pr\{X > 0\} \geqslant \mathbb{E}[X]^2/\mathbb{E}[X^2]$.

Proof. This is an immediate generalization of [7, Remark 3.1]. Let $Y = \mathbb{1}_{X>0}$ the random variable whose value is 1 when $X > 0$, and 0 otherwise. We have

$$\mathbb{E}[X]^2 = (\mathbb{E}[X^+] - \mathbb{E}[X^-])^2 = \mathbb{E}[X^+]^2 - \mathbb{E}[X^-](2\mathbb{E}[X^+] - \mathbb{E}[X^-]) \leqslant \mathbb{E}[X^+]^2 = \mathbb{E}[XY]^2,$$

since $X^+ = XY$. Then, applying Cauchy–Schwarz inequality yields

$$\mathbb{E}[X]^2 \leqslant \mathbb{E}[XY]^2 \leqslant \mathbb{E}[X^2]\mathbb{E}[Y^2] = \mathbb{E}[X^2]\Pr\{X > 0\},$$

since $Y^2 = Y$ and $\mathbb{E}[Y] = \Pr\{X > 0\}$. This concludes the proof. □

We now show that with constant positive probability, the number of branches in the tree is at least a fraction of its expectation.

Lemma 4. *For any h and any $0 < \alpha \leqslant 1$, with probability at least $\frac{1}{5}$, $b_h \geqslant \mathbb{E}[b_h]/2$.*

Proof. We define a random variable $X = b_h - \mathbb{E}[b_h]/2$, and study $\Pr\{X > 0\}$. Since $\mathbb{E}[X] = \mathbb{E}[b_h]/2 > 0$, the second moment method (Lemma 3) applies and $\Pr\{X > 0\} \geqslant \frac{\mathbb{E}[X]^2}{\mathbb{E}[X^2]}$. Now, $\mathbb{E}[X]^2 = \mathbb{E}[b_h]^2/4$, and $\mathbb{E}[X^2] = \mathbb{E}[(b_h)^2] - \frac{3}{4}\mathbb{E}[b_h]^2$. By Lemma 2, $\mathbb{E}[(b_h)^2] < \left(1 + \frac{1\alpha}{1+k\alpha}\right)\mathbb{E}[b_h]^2 < 2\mathbb{E}[b_h]^2$. Then, $\mathbb{E}[X^2] < \frac{5}{4}\mathbb{E}[b_h]^2$. And finally, $\Pr\{X > 0\} \geqslant \frac{\mathbb{E}[X]^2}{\mathbb{E}[X^2]} > \frac{1}{4}\frac{1}{(5/4)} = \frac{1}{5}$. □

3.2. Analysis of the algorithm

We first lower bound the probability of creating a new branch in A during the exploration step. Then, we show that, with constant probability, enough new nodes are visited to guarantee that one of them has a contact at least $2k/\log(1+k)$ times closer to the target. We then conclude the theorem.

The following lemma is directly inspired from [8]. It shows that visiting $\Theta\big(\log n/\log(1+k)\big)$ new nodes is enough to find a long-range link that gets at least $2k/\log(1+k)$ times closer to the target, with constant probability.

Lemma 5. *Given $\gamma > 0$, there is a constant $p'_1 > 0$ (independent of n, x, and k), such that, for any subset Γ of $\gamma \cdot \frac{\log n}{\log(1+k)}$ vertices at Manhattan distance in $\left(\frac{\log(1+k)}{2k}x, x\right]$ to the target, one vertex in Γ (at least) has a long-range contact at Manhattan distance $\leqslant \frac{\log(1+k)}{2k}x$ to the target, with probability at least p'_1.*

Proof. Let $y = \frac{\log(1+k)}{2k}x$ and $\mathbf{u}$ be a node at Manhattan distance $u \in (y, x]$ to the target. The probability that the jth long-range contact of $\mathbf{u}$ is at distance $\leqslant y$ to the target, is larger or equal to[4]

$$\frac{1}{2H_n}\sum_{i=u-y}^{u+y}\frac{1}{i} \geqslant \frac{1}{2H_n}\int_{u-y}^{u+y}\frac{\mathrm{d}t}{t} = \frac{1}{2H_n}\ln\left(1 + \frac{2x\,\log(1+k)}{2ku - x\,\log(1+k)}\right)$$
$$\geqslant \frac{1}{2H_n}\ln\left(1 + \frac{\log(1+k)}{k}\right) \geqslant \frac{\ln 2}{2}\cdot\frac{\log(1+k)}{kH_n},$$

[4] Equality holds for $|u| \leqslant n/2$.

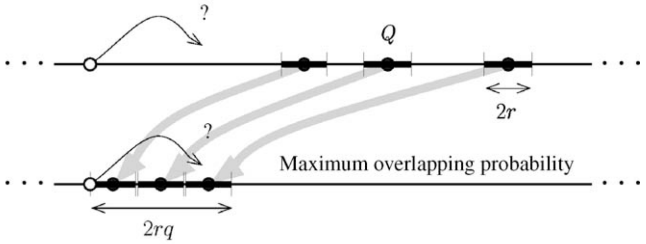

Fig. 3. Minimizing non-overlapping probability (Lemma 6).

since $u \leqslant x$ and $\ln(1+z) \geqslant z \ln 2$, for $z \in [0,1]$. The probability that all the $k|\Gamma|$ long-range contacts of the nodes in Γ are at distance $> \frac{\log(1+k)}{2k}x$ to the target, is then less than $\left(1 - \frac{\ln 2}{2} \cdot \frac{\log(1+k)}{kH_n}\right)^{\gamma(k \log n/\log(1+k))} \leqslant \mathrm{e}^{-\gamma/2} =_{\mathrm{def}} 1 - p'_1 < 1$. □

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

Lemma 6. *Let* **u** *be a node at Manhattan distance u to the target* **t**, **v** *its jth long-range contact, Q a set of q forbidden nodes, and r an integer. The probability that* **v** *is good and at Manhattan distance $\geqslant r$ from any node of Q, is $\geqslant (H_{2u-1} - H_{2rq-1})/(2H_n)$.*

Proof. Let $\mathcal{E}$ be the event that **v** is good and is at distance $\geqslant r$ from any node of Q. $\mathcal{E}$ is the event that **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 **v** is at distance δ from **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 **u**, according to the Manhattan distance (see Fig. 3). A simple case analysis (depending on whether $u \leqslant n/2$, or $n/2 \leqslant u \leqslant n - rq$, or $u \geqslant n - rq$) shows that the probability of $\mathcal{E}$ is then greater than the probability that **v** is at distance $\geqslant 2rq$ from **u**, and is at distance $< u$ to the target. We conclude that $\Pr \mathcal{E} \geqslant \frac{1}{2H_n} \sum_{i=2rq}^{2u-1} \frac{1}{i} = \frac{H_{2u-1} - H_{2rq-1}}{2H_n}$. □

Corollary 7. *There exists a constant n_0 independent of x, n, and k, such that if $n \geqslant n_0$, during any exploration step, for any unvisited node* **u** *at Manhattan distance $u > \frac{\log(1+k)}{2k}x$ to the target, the probability α_u that the jth long-range contact* **v** *of* **u**, *is good and is at Manhattan distance $\geqslant h^1_{\max}(x) + g(x)$ from any node in F, is greater than $\log x/(6 \log n) =_{\mathrm{def}} \alpha^-$.*

Proof. F contains less than $b_{\max}(x)$ nodes. Applying Lemma 6 with $Q = F$ and $r = h^1_{\max}(x) + g(x)$, we have $\alpha_u \geqslant (H_{2u-1} - H_{2b_{\max}(x)(h^1_{\max}(x)+g(x))-1})/2H_n$. Since $h^1_{\max}(x) = \frac{\log\log x - \log\log\log n}{\log(1+k\log x/(6\log n))}$ and $\log(1+kz) \geqslant z \log(1+k)$ for all $0 \leqslant z \leqslant 1$, we have

$$h^1_{\max}(x) \leqslant \frac{6 \log n}{\log x} \frac{\log \log x}{\log(1+k)} \leqslant 6g(x).$$

Since $b_{\max}(x)g(x) = \log n$, we have $2b_{\max}(x)(h^1_{\max}(x) + g(x)) - 1 \leqslant 14 \log n$, and

$$\alpha_u \geqslant \frac{H_{2u-1} - H_{14\log n}}{2H_n} \geqslant \log\left(\frac{\log(1+k) \cdot x/k - 1}{14 \log n}\right) \Big/ (2 \log n),$$

since $u \geqslant x \log(1+k)/(2k)$. Now, rewrite $x/k = (x^{2/3}/k) \cdot x^{1/3}$. Since $x > k \log^2 n$, then: $x^{2/3} > k^{2/3} \log^{4/3} n$, so: $x/k > ((\log^{4/3} n)/k^{1/3}) \cdot x^{1/3}$. As $k \leqslant \log n$, then $x/k > \log n \cdot x^{1/3}$. We conclude that, for $n \geqslant n_0$, for some absolute constant n_0 (independent of x, n and k):

$$\alpha_u > \frac{\log(x^{1/3})}{2\,\log n} = \frac{\log x}{6\,\log n}. \qquad \square$$

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 $\leqslant \frac{\log(1+k)}{2k}x$ from the target, or the tree is wide enough to contain the required number of nodes needed to apply Lemma 5.

Lemma 8. *At the end of any exploration step, with probability at least $\frac{1}{5}$, either there exists a node in A at Manhattan distance $\leqslant \frac{\log(1+k)}{2k}x$ to the target, or $|B_{h_{\mathrm{stop}}}| \geqslant \log x/(2\,\log\log n)$.*

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

Let $Z = \{\mathbf{z} : \delta(\mathbf{z}, \mathbf{t}) > \frac{\log(1+k)}{2k}x\}$ and $\bar{Z}$ its complementary set. During any exploration step, by Corollary 7, the probability that the jth long-range contact of any unvisited node $\mathbf{u} \in Z$ is good and at Manhattan distance $\geqslant h^1_{\max}(x) + g(x)$ of any node in the current F, is at least α^-, for all $1 \leqslant j \leqslant k$. Thus, as long as nodes in Z are considered, each of their long-range contact will be added to B_{h+1} with probability $\geqslant \alpha^-$. 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 jth long-range contact is good and at Manhattan distance $\geqslant h^1_{\max}(x) + g(x)$ 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. We then use *stochastic domination* between the variables: if X_1 and X_2 are random variables taking non-negative integer values, then X_1 is said to *stochastically dominate* X_2 if $\Pr\{X_1 \leqslant i\} \leqslant \Pr\{X_2 \leqslant i\}$ for all $i \geqslant 0$.

The gadget network is built as follows: it 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 jth long-range contact is good and at Manhattan distance $\geqslant h^1_{\max}(x) + g(x)$ from any set of nodes of size $\leqslant b_{\max}(x)$, is α^- (note that this distribution does not need to really exist). We run the exploration step on this gadget network from the same $\mathbf{x}$ as in the real network, except that we do not interrupt it until $h = h^1_{\max}(x)$. 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 $1 \leqslant h \leqslant h_{\mathrm{stop}}$. The links followed during the exploration of the gadget network define a non-overlapping tree structure of height exactly $h^1_{\max}(x)$ on $A' = \bigcup_h A'_h$ where B'_h is the set of the nodes at level h. Let $\mathcal{E}'$ be the event that $|B'_{h^1_{\max}(x)}| \geqslant \log x/(2\,\log\log n)$. We now show that $\Pr\{\mathcal{E}\} \geqslant \Pr\{\mathcal{E}'\}$:

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

We now lower bound $\Pr\{\mathcal{E}'\}$. The set $A' = \bigcup_h A'_h$ 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 $\alpha_{\mathbf{u}} \geqslant \alpha^-$. Thus the number of nodes at level h, $|B'_h|$, stochastically dominates the random variable 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 $0 \leqslant l \leqslant k$. From Lemma 2, we have: $\mathbb{E}[b_h] = (1+k\alpha^-)^h$ and, as a consequence of the second moment method (Lemma 4), with probability at least $\frac{1}{5}$, $b_h \geqslant \mathbb{E}[b_h]/2$. Then, since $(1+k\alpha^-)^{h^1_{\max}(x)} = \log x/\log\,\log n$, we conclude that $\Pr\{\mathcal{E}\} \geqslant \Pr\{\mathcal{E}'\} = \Pr\{|B'_{h^1_{\max}(x)}| \geqslant \log x/(2\,\log\,\log n)\} \geqslant \frac{1}{5}$. $\square$

Corollary 9. *For $n \geqslant n_0$, at the end of any exploration step, with probability at least $\frac{1}{5}$, there is a node in A at Manhattan distance $\leqslant \frac{\log(1+k)}{2k}x$ to the target or there are more than $\frac{\log n}{2\,\log(1+k)}$ distinct unvisited nodes in C.*

Proof. From Lemma 8, with probability at least $\frac{1}{5}$, either there is a node in A at Manhattan distance $\leqslant \frac{\log(1+k)}{2k}x$ to the target or $|B_{h_{\text{stop}}}| \geqslant \log n/(2 \log\log n)$. In the latter case, since $A \cup C$ is composed of $|B_{h_{\text{stop}}}|$ chains of length $\leqslant h^1_{\max}(x) + g(x)$ rooted on the nodes in F, and since the nodes in F are at Manhattan distance $\geqslant h^1_{\max}(x) + g(x)$ from each other, there are exactly $|B_{h_{\text{stop}}}| \cdot g(x) \geqslant \frac{\log n}{2 \log(1+k)}$ unvisited distinct nodes in C. □

We can now conclude that with constant probability, one will find a long-range link that gets close enough to the target.

Proposition 10. *There exist two constants* $p_1 > 0$ *and* n_0, *independent of* n *and* $\mathbf{x}$, *such that, for* $n \geqslant n_0$, *at the end of any exploration step, with probability* $\geqslant p_1$, *there is a node* $\mathbf{u}$ *in* $A \cup C$ *such that* $\mathbf{u}$ *or one of its long-range contact is at distance* $\leqslant \frac{\log(1+k)}{2k}x$ *to the target.*

Proof. From Corollary 9, with probability at least $\frac{1}{5}$, if none of nodes in A is at distance $\leqslant \frac{\log(1+k)}{2k}x$ to the target, there are $\frac{\log n}{2 \log(1+k)}$ distinct nodes in C, and none of their long-range contacts has been visited yet. In that case, either a node in C is at Manhattan distance $\leqslant \frac{\log(1+k)}{2k}x$ or, from Lemma 5, with probability greater than $p'_1 > 0$, at least one node in C has a long-range contact at distance $\leqslant \frac{\log(1+k)}{2k}x$ to the target. Since the long-range links in C have not been visited yet, these events are independent, and setting $p_1 = p'_1/5$ concludes the proof. □

We now can prove Theorem 1 for the case $d = 1$.

Proof (*of Theorem 1*). W.l.o.g., the target is $\mathbf{0}$ and the source $\mathbf{s}$ is at Manhattan distance s from $\mathbf{0}$. Let $\mathbf{x}$ denote the current message holder of the message and x its Manhattan distance to 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 $\left(\frac{2k}{\log(1+k)}\right)^T = s$ and $\left(\frac{2k}{\log(1+k)}\right)^U = \text{stop}_1$. Note that $T \sim \log s/\log(1+k)$ and $U \leqslant 3 \log\log n/\log(1+k)$. We decompose the execution of $\mathcal{A}$ in T phases. The execution is in phase i, $0 \leqslant i \leqslant T$, as long as $\left(\frac{2k}{\log(1+k)}\right)^{i-1} < x \leqslant \left(\frac{2k}{\log(1+k)}\right)^i$. We say that an exploration step in phase i *succeeds* if it leads to a phase $\leqslant i-1$. 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 $T \geqslant i > U$, then $x > \text{stop}_1$. According to Proposition 10, each exploration step succeeds with probability $\geqslant p_1$. Each exploration step visits $\leqslant \left(h^1_{\max}(x) + g(x)\right) b_{\max}(x) \leqslant 7g(x)b_{\max}(x)$ nodes, and routes the message, along a path of length $\leqslant h^1_{\max}(x) + g(x) \leqslant 7g(x)$ towards the target. Then, $\mathbb{E}[Y_i] \leqslant 7g(x)b_{\max}(x)/p_1 \leqslant \frac{7}{p_1}\frac{\log n}{\log(1+k)}$ and $\mathbb{E}[Z_i] \leqslant 7g(x)/p_1 \leqslant \frac{7}{p_1}\frac{\log n \ \log\log n}{i \log^2(1+k)}$, since $\log x \geqslant i \ \log(1+k)$.

Once we reach a phase $i \leqslant U$, we have $x \leqslant \text{stop}_1 = k \ \log^2 n$ and the algorithm runs Kleinberg's greedy algorithm. From [8], we know that this greedy computes a path of expected length $\leqslant A(\log n \ \log(\text{stop}_1)/k \leqslant 3A(\log n \ \log\log n)/k$ while visiting $\leqslant 3A(\log n \log \ \log n)/k$ nodes on expectation, for some constant A.

Let $x_i = \left(\frac{2k}{\log(1+k)}\right)^i$. Note that, $\log x_i \geqslant i \ \log(1+k)/2$. The expected length of the path from $\mathbf{s}$ to $\mathbf{0}$ computed by our algorithm is bounded by

$$\begin{aligned}\sum_{i=0}^{T} \mathbb{E}[Z_i] &\leqslant A \ \log n \ \log(\text{stop}_1)/k + \frac{1}{p_1} \sum_{U<i\leqslant T} (h^1_{\max}(x_i) + g(x_i)) \\ &\leqslant 3A \ \frac{\log n \ \log\log n}{k} + \frac{14}{p_1}\frac{\log n \ \log\log n}{\log^2(1+k)} \sum_{U<i\leqslant T} \frac{1}{i} \\ &= \mathrm{O}\left(\log n \left(\frac{\log \ \log n}{\log(1+k)}\right)^2\right).\end{aligned}$$

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

$$\sum_{i=0}^{T} \mathbb{E}[Y_i] = \frac{1}{p_1} \sum_{i=0}^{T} (h^1_{\max}(x_i) + g(x_i))\, b_{\max}(x_i) = \mathrm{O}\left(\left(\frac{\log n}{\log(1+k)}\right)^2\right).$$

For the last of each exploration step, our algorithm just needs $\Theta\big(\log n \cdot (b_{\max}(x) + h^1_{\max}(x) + g(x))\big) = \mathrm{O}(\log^2 n / \log(1+k))$ 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 $\leqslant b_{\max}(x) = \mathrm{O}(\log n / \log \log n) = \mathrm{O}(\log n / \log(1+k))$), the state of the stack during the depth-first search of $A \cup C$ (whose height is bounded by $h^1_{\max}(x) + g(x) = \mathrm{O}(\log n / \log(1+k))$), 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. This concludes the proof for $d = 1$. $\square$

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 $r \leqslant n$, $S(r) = \Theta(r^{d-1})$ and $V(r) = \Theta(r^d)$. More precisely, for $r \leqslant n$, $S(r) = \frac{2^d}{(d-1)!} r^{d-1} + \xi(r)$, 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 also upper bounds on $S(r)$ and $V(r)$ when $r > n$.

Furthermore, let $H_{n,d} = \sum_{i=1}^{nd} S(i)/i^d$ be the repartition function for the long-range link length probability distribution, we get the following lower and upper bounds:

$$H_{n,d} \geqslant \sum_{i=1}^{n} \frac{S(i)}{i^d} \geqslant \frac{2^d}{(d-1)!} H_n$$

and

$$H_{n,d} \leqslant \sum_{i=1}^{nd} \left(\frac{2^d}{(d-1)!} \frac{i^{d-1}}{i^d} + \mathrm{O}\left(\frac{1}{i^2}\right) \right) \leqslant \frac{2^d}{(d-1)!} H_n + \mathrm{O}(1),$$

The algorithm for d-dimensional networks: We only need to adapt the parameters of the one-dimensional routing algorithm. Everything else in the algorithm is unchanged: $b_{\max}(x)$ and $g(x)$ are unchanged; $h^1_{\max}(x)$ and stop_1 are, respectively, replaced by $h^d_{\max}(x)$ and stop_d, given in frame Algorithm 2.

Algorithm 2. Decentralized routing algorithm for *d*-dimensional networks

Let $h^d_{\max}(x) = \dfrac{\log \log x - \log \log \log n}{\log\left(1 + k \frac{\log x}{2^{d+2} \log n}\right)}$, *(Note that again,* $h^d_{\max}(x) = \mathrm{O}(\log n \log \log x / (\log(1+k) \log x))$*)*

$b_{\max}(x) = \dfrac{\log x}{\log \log n}$, $g(x) = \dfrac{\log n \log \log n}{\log(1+k) \log x}$, and $\mathrm{stop}_d = k^{2/d} \log^{2^{2d+2}} n$.

Run Algorithm 1 on the *d*-dimensional network with these adapted parameters.

Outline the analysis of the algorithm on a d-dimensional network: The analysis of the algorithm is essentially identical to the one-dimensional case. The key is again to prove that, with some constant probability p_d, there exists a vertex in $A \cup C$ whose long-range contact is, this time, at least $\left(\frac{2k}{\log(1+k)}\right)^{1/d}$ times closer to the target (Proposition 19). We first derive several geometrical lemmas that will be used to obtain similar lemmas to the case $d = 1$. The main point is that with our choice of the parameters $b_{\max}(x)$, $g(x)$, $h^d_{\max}(x)$ and stop_d, Lemma 17 yields the same kind of lower bound on the growth of the underlying process and thus yields Theorem 1.

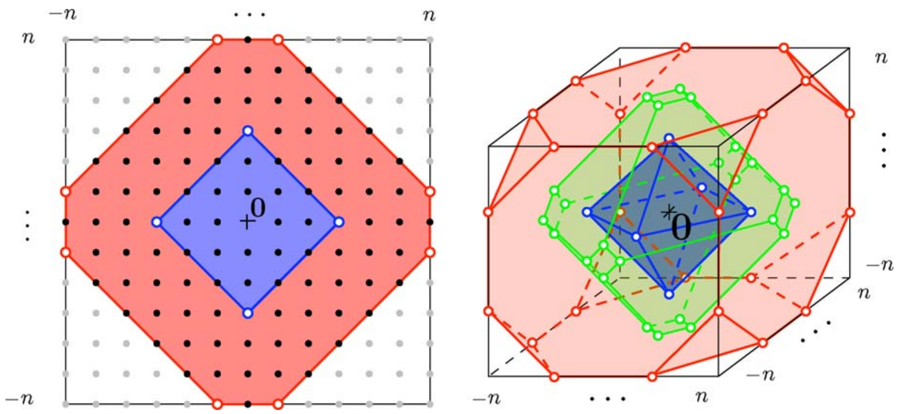

Fig. 4. Two examples of corners (points in white) of balls centered on **0** in the torus two-dimensional and three-dimensional.

4.1. Preliminary geometrical lemmas

This section presents several geometrical properties of the d-dimensional Kleinberg network that will be used in the next section to lower bound the number of nodes visited during each exploration step.

Definition 3. We say that a node $\mathbf{u}$ is a corner of the sphere $\mathcal{S}(\mathbf{0}, \mathbf{r})$ if $\mathbf{u} \in \mathcal{S}(\mathbf{0}, \mathbf{r})$ and at most one coordinate of $\mathbf{u}$ does not belong to $\{-n, 0, n\}$ (see Fig. 4). The corners of a sphere $\mathcal{S}(\mathbf{v}, r)$ centered on an arbitrary node $\mathbf{v}$ are defined by translation.

Note that when $r \leqslant n$, $\mathcal{S}(\mathbf{v}, r)$ is a "d-dimensional octahedron", and our definition of a corner matches the usual geometric definition.

Lemma 11. *Let* $\mathbf{u}$ *a node at distance* u *to the target. For any* $r \geqslant 0$, *the probability that* $\mathbf{u}$*'s jth long-range contact belongs to the ball* $\mathcal{B}(\mathbf{t}, r)$, *is minimum when* $\mathbf{u}$ *is at a corner of the sphere* $\mathcal{S}(\mathbf{t}, u)$ *of radius u around the target* $\mathbf{t}$.

Proof. W.l.o.g., $\mathbf{0}$ is the target. Take a vertex $\mathbf{u} = (u_1, \ldots, u_d)$, which is not a corner of $\mathcal{S}(\mathbf{0}, u)$. W.l.o.g., $n > u_1 \geqslant u_2 > 0$. Let $\mathbf{v} = (u_1 + 1, u_2 - 1, u_3, \ldots, u_d) \in \mathcal{S}(\mathbf{0}, u)$. Let $\rho_{\mathbf{u}}$ (resp., $\rho_{\mathbf{v}}$) be the probability that $\mathbf{u}$'s (resp., $\mathbf{v}$'s) jth long-range contact belongs to $\mathcal{B}(\mathbf{0}, \mathbf{r})$. We will show that $\rho_{\mathbf{v}} \leqslant \rho_{\mathbf{v}}$. Iterating this process a finite number of times pushes $\mathbf{v}$ to a corner, and yields the result.

In order to prove the result, we construct a bijection $\phi : \mathcal{B}(\mathbf{0}, r) \to \mathcal{B}(\mathbf{0}, r)$, such that for all $\mathbf{z} \in \mathcal{B}(\mathbf{0}, r)$, $\delta(\phi(\mathbf{z}), \mathbf{v}) \geqslant \delta(\phi(\mathbf{z}), \mathbf{u})$. In fact, for most of the nodes $\mathbf{z} \in \mathcal{B}(\mathbf{0}, r)$, the Manhattan distance $\delta(\phi(\mathbf{z}), \mathbf{v})$ will be equal to $\delta(\mathbf{z}, \mathbf{u})$; and for the rest of the nodes $\mathbf{z}$, $\delta(\phi(\mathbf{z}), \mathbf{v}) \geqslant \delta(\mathbf{z}, \mathbf{u})$. Since the probability that a node $\mathbf{z}$ is a long-range contact of $\mathbf{u}$ decreases with the Manhattan distance $\delta(\mathbf{z}, \mathbf{u})$, this implies the result:

$$\rho_{\mathbf{v}} = \sum_{\mathbf{z} \in \mathcal{B}(\mathbf{0}, r)} \frac{1}{\delta(\mathbf{z}, \mathbf{v}) H_{n,d}} = \sum_{\mathbf{z} \in \mathcal{B}(\mathbf{0}, r)} \frac{1}{\delta(\phi(\mathbf{z}), \mathbf{v}) H_{n,d}} \leqslant \sum_{\mathbf{z} \in \mathcal{B}(\mathbf{0}, r)} \frac{1}{\delta(\mathbf{z}, \mathbf{u}) H_{n,d}} = \rho_{\mathbf{u}}.$$

ϕ is based on the translation of vector $(1, -1, 0, \ldots, 0)$. The action of ϕ is illustrated Fig. 5. Let

$$\Gamma_1 = \{\mathbf{z} \in \mathcal{S}(\mathbf{0}, r) \cup \mathcal{S}(\mathbf{0}, r-1) : z_1 \geqslant 0 \text{ and } z_2 \leqslant 0\}$$

and

$$\Gamma_2 = \{\mathbf{z} \in \mathcal{S}(\mathbf{0}, r) \cup \mathcal{S}(\mathbf{0}, r-1) : z_1 \leqslant 0 \text{ and } z_2 \geqslant 0\}.$$

Note that $\Gamma_1 \cap \Gamma_2 = \varnothing$. We define $\phi : \mathcal{B}(\mathbf{0}, r) \to \mathcal{B}(\mathbf{0}, r)$ as follows:

$$\phi : \mathbf{z} \mapsto \begin{cases} (z_1 + 1, z_2 - 1, z_3, z_4, \ldots, z_d) & \text{if } \mathbf{z} \notin \Gamma_1, \\ (z_2, z_1, z_3, z_4 \ldots, z_d) & \text{otherwise.} \end{cases}$$

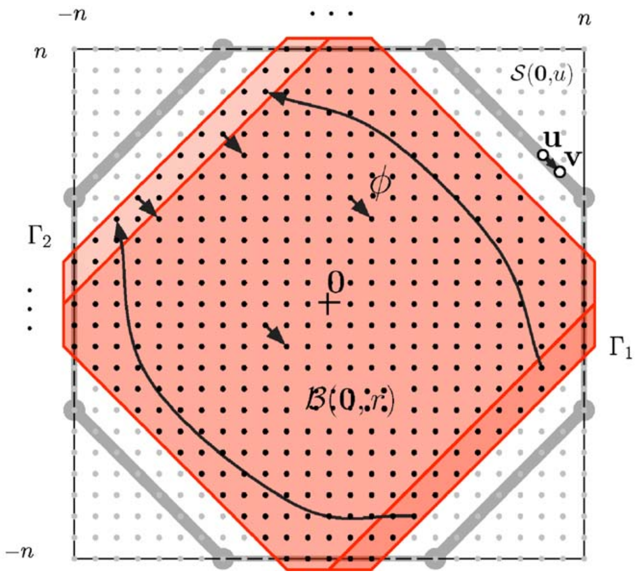

Fig. 5. Γ_1, Γ_2, and the action of ϕ (shown by the black arrows).

Note that ϕ is a bijection since: $\mathbf{z} \in \Gamma_1 \Leftrightarrow \phi(\mathbf{z}) \in \Gamma_2$, and the restriction of ϕ to $\mathcal{B}(\mathbf{0}, r) \setminus \Gamma_1$ is a translation onto $\mathcal{B}(\mathbf{0}, r) \setminus \Gamma_2$. In particular, for $\mathbf{z} \in \mathcal{B}(\mathbf{0}, r) \setminus \Gamma_1$, $\delta(\mathbf{z}, \mathbf{u}) = \delta(\phi(\mathbf{z}), \mathbf{v})$. Consider now $\mathbf{z} \in \Gamma_1$, and let

$$\Delta = \delta(\phi(\mathbf{z}), \mathbf{v})) - \delta(\mathbf{z}, \mathbf{u}) = |z_2 - 1 - u_1| + |z_1 + 1 - u_2| - |z_1 - u_1| - |z_2 - u_2|.$$

Since $z_1 \geqslant 0$, $z_2 \leqslant 0$, and $0 < u_2 \leqslant u_1$, $\Delta = 1 + u_1 - u_2 + |z_1 + 1 - u_2| - |z_1 - u_1|$. Now, If $z_1 \leqslant u_2 - 1$, then $\Delta = 0$; if $u_2 - 1 \leqslant z_1 \leqslant u_1$, $\Delta = 2(1 - u_2 + z_1) \geqslant 0$; and finally, if $z_1 \geqslant u_1$, $\Delta = 2(1 + u_1 - u_2) \geqslant 2$. Δ is thus always non-negative, which concludes the proof. □

Lemma 12. *For all* $1 \leqslant x \leqslant \min(r, n)$, *the number of nodes in* $\mathcal{B}(\mathbf{0}, r)$ *at Manhattan distance* x *from a corner of* $\mathcal{S}(\mathbf{0}, r)$ *is greater than* $\frac{x^{d-1}}{(d-1)!}$.

Proof. First, assume $r \leqslant n$. W.l.o.g., consider the corner $c = (r, 0, \ldots, 0)$. Let $\mathbf{c}' = (r - x/2, 0, \ldots, 0)$ (w.l.o.g. assume $r - x/2$ is an integer). For $x \leqslant r$, each node $\mathbf{y}$ in $\mathcal{S}(\mathbf{c}', x/2)$ such that $y_1 \leqslant r - x/2$ belongs to $\mathcal{B}(\mathbf{0}, r)$ and is at distance x from $\mathbf{c}$ (see Fig. 6). And, there are $\frac{1}{2}S(x/2) \geqslant \frac{x^{d-1}}{(d-1)!}$ such nodes.

Consider now $r > n$, let $s = \lfloor r/n \rfloor$ and $x \leqslant n$. W.l.o.g., consider the corner $\mathbf{c}$ with $c_i = n$ for $i \leqslant s$, $c_{s+1} = r - sn$, and $c_i = 0$ for $i > s + 1$. Let $\mathbf{c}'$ the image of $\mathbf{c}$ by the translation of a vector $(-x/2, 0, \ldots, 0)$. Each node $\mathbf{y}$ in $\mathcal{S}(\mathbf{c}', x/2)$ such that $y_1 \leqslant n - x/2$ belongs to $\mathcal{B}(\mathbf{0}, r)$ and is at distance x from $\mathbf{c}$. Again, there are $\frac{1}{2}S(x/2) \geqslant \frac{x^{d-1}}{(d-1)!}$ such nodes.

We conclude that for $x \leqslant \min(n, r)$, there are always at least $\frac{x^{d-1}}{(d-1)!}$ nodes at distance x from any corner of $\mathcal{S}(\mathbf{0}, r)$ in $\mathcal{B}(\mathbf{0}, r)$. □

4.2. Analysis of the algorithm

The proof reads along the lines of Section 3. We first show that with constant probability, among $\Theta\left(\frac{\log n}{\log(1+k)}\right)$ nodes, one has a long-range contact at least $\left(\frac{2k}{\log(1+k)}\right)^{1/d}$ times closer to the target.

Lemma 13. *For* $0 < \lambda \leqslant 1/2$, *the probability that a node* $\mathbf{u}$ *at distance* u *from the target, has its jth long-range contact at distance* $\leqslant \lambda u$ *from the target, is greater than*: $\frac{\ln 2}{d^d\ 2^{2d}} \frac{\lambda^d}{H_n}$.

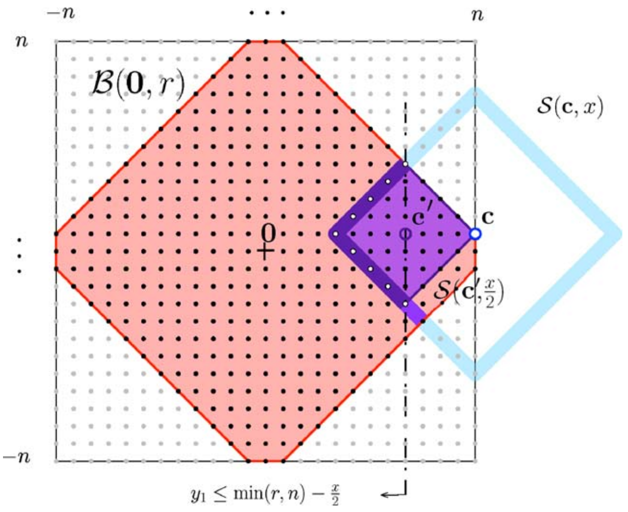
$-n$
$\cdots$
n
n
$\mathcal{B}(\mathbf{0}, r)$
$\mathcal{S}(\mathbf{c}, x)$
$\mathbf{c}'$
$\mathbf{c}$
$\mathbf{0}$
$\vdots$
$\mathcal{S}(\mathbf{c}', \frac{x}{2})$
$-n$
$y_1 \leq \min(r, n) - \frac{x}{2}$

Fig. 6. Illustration of the proof of lemma 12.

Proof. Let P be the probability to evaluate. W.l.o.g., the target is $\mathbf{0}$. From Lemma 11, P is minimized when $\mathbf{u}$ is at a corner of $\mathcal{S}(\mathbf{0}, u)$. W.l.o.g., assume that $\mathbf{u}$ is the corner $\mathbf{u} = (n, \dots, n, u_l, 0, \dots, 0)$, where $0 \leqslant u_l < n$. Let $\mathbf{c} = (n, \dots, n, c_{l'}, 0, \dots, 0)$ the (closest) corner of $\mathcal{S}(\mathbf{0}, \lambda u)$, which is at Manhattan distance exactly $(1-\lambda)u$ from $\mathbf{u}$. P is lower bounded by the sum, for i from 1 to $2\lambda u$, of the probabilities that the jth long-range contact is at distance i from $\mathbf{c}$ and in $\mathcal{B}(\mathbf{0}, \lambda u)$. As long as $i \leqslant n$, Lemma 12 guarantees that there are at least $i^{d-1}/(d-1)!$ nodes in $\mathcal{B}(\mathbf{0}, \lambda u)$ at distance i from $\mathbf{c}$, that is to say, at distance $\leqslant i + (1-\lambda)u$ from $\mathbf{u}$. Summing the corresponding probabilities, for i from $\lambda u/(2d)$ to $\lambda u/d$ ($\leqslant n$) gives then the following lower bound on P. Since for theses values of i, $i + (1-\lambda)u \leqslant \frac{2d}{\lambda} i$, we get

$$P \geqslant \sum_{i=\lambda u/(2d)}^{\lambda u/d} \frac{i^{d-1}/(d-1)!}{H_{n,d}(i + (1-\lambda)u)^d} \geqslant \frac{\lambda^d}{(2d)^d (d-1)! H_{n,d}} \int_{\lambda u/(2d)}^{\lambda i/d} \frac{\mathrm{d}t}{t}$$
$$\geqslant \frac{\lambda^d \ln 2}{(2d)^d (d-1)! H_{n,d}} \geqslant \frac{\ln 2}{d^d\, 2^{2d}} \frac{\lambda^d}{H_n}. \qquad \square$$

Corollary 14. *Given $\gamma > 0$, there is a constant $p'_d > 0$ (independent of n, x, and k), such that, for any subset Γ of $\gamma \cdot \frac{\log n}{\log(1+k)}$ vertices at Manhattan distance in $\left(\left(\frac{\log(1+k)}{2k}\right)^{1/d} \cdot x, x\right]$ to the target, one vertex in Γ (at least) has a long-range contact at lattice distance $\leqslant \left(\frac{\log(1+k)}{2k}\right)^{1/d} \cdot x$ to the target, with probability at least p'_d.*

Proof. According to Lemma 13, with $\lambda = \left(\frac{\log(1+k)}{2k}\right)^{1/d}$, the probability that the jth long-range contact of a given node $\mathbf{u}$ is at distance $\leqslant \lambda u$ to the target, is $\geqslant \ln 2/d^d\, 2^{2d+1} \log(1+k)/kH_n$. The probability that all the $k|\Gamma|$ long-range contacts of the nodes in Γ are at distance $> \left(\frac{\log(1+k)}{2k}\right)^{1/d} \cdot x$ to the target, is then less than $\left(1 - \frac{\ln 2}{d^d\, 2^{2d+1}} \frac{\log(1+k)}{kH_n}\right)^{\gamma k \log n/\log(1+k)} \leqslant \mathrm{e}^{-\gamma \log 2/(d^d\, 2^{2d+1})} =_{\mathrm{def}} 1 - p'_d < 1$. $\square$

We now lower bound the growth of the underlying branching process in $A \cup C$, by first lower bounding as before the probability of creating a new branch by $\alpha_d^- = \Theta(\log x/\log n)$.

Lemma 15. *There exists a constant $c_1 \geqslant 0$, depending only on d such that, for any node $\mathbf{u}$ at Manhattan distance u to the target $\mathbf{t}$, the probability that its jth long-range contact $\mathbf{v}$ is good is at least $(H_u - c_1)/((d-1)!H_{n,d})$.*

Proof. From Lemma 11 the probability P that $\mathbf{v}$ is good, is minimized when $\mathbf{u}$ is a corner of $\mathcal{S}(\mathbf{t}, u)$. From Lemma 12, the number of nodes in $\mathcal{B}(\mathbf{t}, u-1)$ at lattice distance i from a corner of $\mathcal{S}(\mathbf{t}, u-1)$ is greater than $i^{d-1}/(d-1)!$, then

$$P \geqslant \sum_{i=1}^{\min(u,n)} \frac{i^{d-1}}{(d-1)!} \cdot \frac{1}{i^d H_{n,d}} \geqslant \frac{H_{\min(u,n)}}{(d-1)! H_{n,d}} \geqslant \frac{H_u - c_1}{(d-1)! H_{n,d}},$$

for $c_1 = \ln d + 1$, using $u \leqslant nd$. □

Lemma 16. *Let $1 \leqslant j \leqslant k$, $\mathbf{u}$ a node at Manhattan distance u to the target, $\mathbf{v}$ its jth long-range contact, Q a set of q forbidden nodes, and r an integer. The probability that $\mathbf{v}$ is good and at distance $\geqslant r$ from any node in Q, is $\geqslant \frac{H_u - 2^d H_{rq^{1/d}} - c_3}{2^d H_n + c_4}$, for two constants $c_3, c_4 \geqslant 0$, that only depend on d.*

Proof. Let $\mathcal{E}$ be the event that $\mathbf{v}$ is good and is at distance $\geqslant r$ from any node of Q. $\mathcal{E}$ is the event that $\mathbf{v}$ is good and does not belong to any of the q balls of radius r 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 all the forbidden nodes are as close as possible to $\mathbf{u}$, that is to say, when the forbidden nodes are contained in a ball centered on $\mathbf{u}$ of minimum radius δ, where $\delta \leqslant c_2 q f^{1/d}$, for some constant $c_2 > 0$, is the solution to $V(\delta) = q \cdot V(r)$.

We conclude that: $\Pr\{\mathcal{E}\} \geqslant \Pr\{\mathbf{v} \in \mathcal{B}(\mathbf{t}, u-1)\} - \Pr\{\mathbf{v} \in \mathcal{B}(\mathbf{u}, \delta)\} \geqslant \frac{H_u - c_3}{(d-1)! H_{n,d}} - \frac{H_{\delta,d}}{H_{n,d}}$, from Lemma 15, thus: $\Pr\{\mathcal{E}\} \geqslant \frac{H_u - 2^d H_{rq^{1/d}} - c_3}{2^d H_n + c_4}$, for some constants c_3 and c_4 that only depend on d. □

The following lemma corresponds to Lemma 7 in dimension 1.

Lemma 17. *There exists a constant n_d, independent of x, n, and k, such that if $n \geqslant n_d$, during any exploration step, for any unvisited node $\mathbf{u}$ at Manhattan distance $u \geqslant \left(\frac{\log(1+k)}{2k}\right)^{1/d} \cdot x$ to the target, the probability $\alpha_{u,d}$ that the jth long-range contact $\mathbf{v}$ of $\mathbf{u}$, is good and is at Manhattan distance $\geqslant h^d_{\max}(x) + g(x)$ from any node in F, is greater than $\frac{\log x}{2^{d+2} \log n} =_{\text{def}} \alpha_d^-$.*

Proof. Let $c = 1/2^{d+2}$, so that $h^d_{\max}(x) = \frac{\log\log x - \log\log\log n}{\log(1 + kc\log x/\log n)}$. At any time of the exploration, F contains less than $b_{\max}(x)$ nodes. By Lemma 16,

$$\alpha_{u,d} \geqslant \frac{H_u - 2^d H_{(h^d_{\max}(x)+g(x))b_{\max}(x)^{1/d}} - c_3}{2^d H_n + c_4}.$$

By concavity, $\log(1 + kz) \geqslant z\log(1+k)$ for $z \in [0, 1]$, so $h^d_{\max}(x) \leqslant \frac{\log n}{c\log x} \frac{\log\log x}{\log(1+k)}$. Then, $h^d_{\max}(x) + g(x) \leqslant \frac{2\log n}{c\log x} \frac{\log\log x}{\log(1+k)}$. Thus,

$$(h^d_{\max}(x) + g(x))b_{\max}(x)^{1/d} \leqslant (h^d_{\max}(x) + g(x))b_{\max} \leqslant \frac{2\log n}{c\log(1+k)}.$$

Since $u \geqslant (\frac{\log(1+k)}{2k})^{1/d} x$, we get

$$\alpha_{u,d} \geqslant \frac{H_u - 2^d H_{2\log n/c\log(1+k)} - c_3}{2^d H_n + c_4} \geqslant \frac{\log\left((\log(1+k)/2k)^{1/d} x\ (c^{2^d} \log^{2^d}(1+k)/2^{2^d} \log^{2^d} n)\right) - c_3/\ln 2}{2^d \log n + c_4/\ln 2}.$$

But $x > \text{stop}_d = k^{2/d} \log^{2^{2d+2}} n$, then we can write $x = (\sqrt{x})^2 \geqslant x^{1/2} \cdot k^{1/d} \log^{2^{d+1}} n$. Since also $\log(1+k) \geqslant \log 2 = 1$, this yields

$$\alpha_{u,d} \geqslant \frac{\log\left(x^{1/2}\ (c\log n/2)^{2^d}\right) - c_3/\ln 2}{2^d \log n + c_4/\ln 2} = \frac{\log x + 2^d \log(c\log n/2) - c_3/\ln 2}{2^{d+1}\log n + 2c_4/\ln 2}.$$

Then, there exists a constant $n_d \geqslant 2^{2/c}$, independent of n, k, and x such that for $n \geqslant n_d$,

$$\alpha_{u,d} \geqslant \frac{\log x}{2^{d+2}\log n} = \frac{c \log x}{\log n}. \qquad \square$$

Since the parameters are chosen so that $(1 + k\alpha_d^-)^{h^d_{\max}(x)} = (1 + k\,\frac{\log x}{2^{d+2}\log n})^{h^d_{\max}(x)} = \log x/\log\log n = b_{\max}(x)$, we are now in the exact same framework as in dimension 1. We shall then only state the lemmas, the proofs are exactly the same as in dimension 1, and therefore omitted.

Lemma 18. *For $n \geqslant n_d$, at the end of any exploration step, with probability at least $\frac{1}{5}$, there is a node in A at Manhattan distance $\leqslant \left(\frac{\log(1+k)}{2k}\right)^{1/d} \cdot x$ to the target or there are more than $\frac{\log n}{2\log(1+k)}$ distinct nodes in C.*

Proof. As in Corollary 9, we use Section 3.1 to prove that, with probability at least $\frac{1}{5}$, either there exists a node in A at Manhattan distance $\leqslant \left(\frac{\log(1+k)}{2k}\right)^{1/d} \cdot x$ to the target, or $|B_{h_{\text{stop}}}| \geqslant \log x/(2\log\log n)$. $\square$

The following proposition is the last step to Theorem 1 in dimension $d > 1$ (as Proposition 10 in dimension 1).

Proposition 19. *There exist two constants $p_d > 0$ and n_d, independent of n and $\mathbf{x}$, such that, for $n \geqslant n_d$, at the end of any exploration step, with probability $\geqslant p_d$, there is a node $\mathbf{u}$ in $A \cup C$ such that $\mathbf{u}$ or one of its long-range contact is at distance $\leqslant \left(\frac{\log(1+k)}{2k}\right)^{1/d} \cdot x$ to the target.*

We can now conclude with the proof of Theorem 1 in dimension $d > 1$.

Proof (*of Theorem 1*). Let N and M be the solutions to $\left(\frac{2k}{\log(1+k)}\right)^{N/d} = s$ and $\left(\frac{2k}{\log(1+k)}\right)^{M/d} = \text{stop}_d$. Note that $N \sim \frac{d\,\log s}{\log(1+k)}$ and $M \sim \frac{d\,2^{2d+2}\,\log\log n}{\log(1+k)}$. As in dimension 1, we decompose the execution of $\mathcal{A}$ in N phases. Let Z_i and Y_i denote, respectively, the random variable for the number of steps and for the number of nodes visited in phase $0 \leqslant i \leqslant N$, and $x_i = \left(\frac{2k}{\log(1+k)}\right)^{i/d}$. We get the formulae:

$$\sum_{i=0}^{N} \mathbb{E}[Z_i] \leqslant B\,\frac{\log n\,\log(\text{stop}_d)}{k} + \frac{1}{p_d} \sum_{N<i\leqslant M} \left(h^d_{\max}(x_i) + g(x_i)\right)$$

and

$$\sum_{i=0}^{N} \mathbb{E}[Y_i] \leqslant B\,\frac{\log n\;\log(\text{stop}_d)}{k} + \tfrac{1}{p_d} \sum_{N<i\leqslant M} \left(h^d_{\max}(x_i) + g(x_i)\right) b_{\max}(x_i),$$

where $B > 0$ is a constant (independent of n, s and k). With our choice of parameters stop_d and $h^d_{\max}(x)$, we have $h^d_{\max}(x_i) + g(x_i) \leqslant \frac{2^{d+3}\log n\,\log\log n}{\log x_i\,\log(1+k)}$, and also $\log x_i \geqslant \frac{i\,\log(1+k)}{2d}$ we get

$$\begin{aligned}\sum_{i=0}^{N} \mathbb{E}[Z_i] &\leqslant B\,2^{2d+2}\,\frac{\log n\,(\log\log n + \log k)}{k} + \frac{d\,2^{d+3}}{p_d}\,\frac{\log n\,\log\log n}{\log^2(1+k)} \sum_{M<i\leqslant N} \frac{1}{i} \\ &= \mathrm{O}\left(\log n \left(\frac{\log\,\log n}{\log(1+k)}\right)^2\right), \\ \sum_{i=0}^{N} \mathbb{E}[Y_i] &= \mathrm{O}\left(\left(\frac{\log n}{\log(1+k)}\right)^2\right).\end{aligned}$$

Finally, the algorithm uses $\mathrm{O}\big(\log n \cdot (b_{\max} + h^d_{\max}(x) + g(x))\big) = \mathrm{O}\big(\frac{\log^2 n}{\log(1+k)}\big)$ bits for the last of each exploration step. $\square$

Remark. The parameters for the d-dimensional network have been set based on worst case analysis; the parameters may possibly be optimized. Practical implementations of our routing algorithm on d-dimensional networks may try to reduce the 2^d factors in $h^d_{\max}(x)$ and stop_d.

5. 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. It has been proved later on by [12] that the diameter of Kleinberg's network is $\Theta(\log n)$ w.h.p. when $k = O(1)$, but their proof does not provide a decentralized algorithm that computes a path of that expected length. An interesting question is to prove the existence of a decentralized algorithm that could route message along optimal paths.

References

[1] J. Aspnes, Z. Diamadia, G. Shah, Fault-tolerant routing in peer-to-peer systems, in: Proc. of ACM third Symp. on Princ. of Distr. Comp. (PODC 2002), Vol. 31, 2002, pp. 223–232.

[2] L. Barrière, P. Fraigniaud, E. Kranakis, D. Krizanc, Efficient routing in networks with long range contacts, LNCS Proc. of 15th Internat. Symp. on Distributed Computing (DISC '01), Vol. 2180, 2001, pp. 270–284.

[3] I. Benjamini, N. Berger, The diameter of long-range percolation clusters on finite cycles, Random Structures and Algorithms 19 (2001) 102–111.

[4] B. Bollobás, F. Chung, The diameter of a cycle plus random matching, SIAM J. Discrete Math. 1 (1988) 328–333.

[5] D. Coppersmith, D. Gamarnik, M. Sviridenko, The diameter of a long range percolation graph, Random Structures Algorithms 21 (2002) 1–13.

[6] P. Fraigniaud, C. Gavoille, C. Paul, Eclecticism shrinks even small worlds, in: Proc. of 23rd ACM Symp. on Principles of Distributed Computing (PODC), 2004, 169–178.

[7] S. Janson, T. Łuczak, A. Ruciński, Random Graphs, Wiley, New York, 2000.

[8] J. Kleinberg, The small-world phenomenon: an algorithmic perspective, proc. 32nd ACM Symp. on Theory of Computing, 2000, 163–170.

[9] J. Kleinberg, Small-world phenomena and the dynamics of information, Advances in Neural Information Processing Systems, Vol. 14, MIT Press, Cambridge, MA, 2002.

[10] G.S. Manku, M. Bawa, P. Raghavan, Symphony: distributed hashing in a small world, in: Proc. of Fourth Usenix Symp. on Internet Technologies and Systems, 2003.

[11] G.S. Manku, M. Naor, U. Wieder, Know thy neighbor's neighbor: the power of lookahead in randomized P2P networks, in: Proc. of 36th ACM Symp. on Theory of Computing (STOC), 2004, 54–63.

[12] C. Martel, V. Nguyen, Analyzing Kleinberg's (and other) small-world models, in: Proc. of 23rd Annual ACM Symp. on Principles of Distributed Computing (PODC), 2004, 179–188.

[13] S. Milgram, The small world problem, Psychology Today 61 (1967).

[14] M.E.J. Newman, Models of the small world, J. Statist. Phys. 101 (2000).

[15] M.E.J. Newman, The structure and function of complex networks, SIAM Rev. 45 (2003) 167–256.

[16] M.E.J. Newman, C. Moore, D.J. Watts, Mean-field solution of the small-world network model, Phys. Rev. Lett. 84 (2000) 3201–3204.

[17] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, H. Balakrishnan, Chord: a scalable peer-to-peer lookup service for internet applications, in: Proc. of ACM SIGCOMM 2001, 2001.

[18] D. Watts, S. Strogatz, Collective dynamics of small-world networks, Nature 393 (1998).

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