Towards Small World Emergence *

Philippe Duchon, Nicolas Hanusse
LABRI
Domaine universitaire 351, cours de la libération,
33405 Talence Cedex, France
duchon@labri.fr, hanusse@labri.fr

Emmanuelle Lebhar, Nicolas Schabanel
LIP, ENS Lyon
46 allée d'Italie, 69364 Lyon Cedex 07, France
elebhar@ens-lyon.fr,
nschaban@ens-lyon.fr

ABSTRACT

We investigate the problem of optimizing the routing performance of a virtual network by adding extra random links. Our asynchronous and distributed algorithm ensures, by adding a single extra link per node, that the resulting network is a navigable small world, i.e., in which greedy routing, using the distance in the original network, computes paths of polylogarithmic length between any pair of nodes with probability $1 - O(1/n)$. Previously known small world augmentation processes require the global knowledge of the network and centralized computations, which is unrealistic for large decentralized networks. Our algorithm, based on a careful multi-layer sampling of the nodes and the construction of a light overlay network, bypasses these limitations. For bounded growth graphs, i.e., graphs where, for any node u and any radius r the number of nodes within distance $2r$ from u is at most a constant times the number of nodes within distance r, our augmentation process proceeds with high probability in $O(\log n \log D)$ communication rounds, with $O(\log n \log D)$ messages of size $O(\log n)$ bits sent per node and requiring only $O(\log n \log D)$ bit space in each node, where n is the number of nodes, and D the diameter. In particular, with the only knowledge of original distances, greedy routing computes, between any pair of nodes in the augmented network, a path of length at most $O(\log^2 n \log^2 D)$ with probability $1 - O(1/n)$, and of expected length $O(\log n \log^2 D)$. Hence, we provide a distributed scheme to augment any bounded growth graph into a small world with high probability in polylogarithmic time while requiring polylogarithmic memory. We consider that the existence of such a lightweight process might be a first step towards the definition of a more general construction process that would validate Kleinberg's model as a plausible explanation for the small world phenomenon in large real interaction networks.

*The authors were supported by the french CNRS grant Locoglobo.

SPAA'06, July 30–August 2, 2006, Cambridge, Massachusetts, USA.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and design—*Network topology*; C.2.2 [**Computer-Communication Networks**]: Network Protocols—*Routing protocols*; C.2.4 [**Computer-Communication Networks**]: Distributed systems

General Terms

Algorithms, Design, Performance

Keywords

Distributed algorithms, small world, routing algorithms, bounded growth

1. INTRODUCTION

In this paper, we investigate the problem of efficiently preprocessing a large virtual network, in a fully distributed manner, so that the resulting network is a navigable small world. Namely, by adding a single entry to each address book, we obtain a network in which greedy routing computes paths of polylogarithmic expected length between any pair of nodes. Such a scheme is called a *small world augmentation process*. This problem arises as an application of recent investigations on the small world phenomenon in real interaction networks (e.g. social networks, or peer-to-peer networks). This phenomenon consists in the combination of a low diameter *and* the ability, for each node, to discover short paths without the global knowledge of all connections, as exhibited in the Milgram's seminal experiment [15].

The first graph model reproducing the small world navigability was proposed by Kleinberg in 2000 [12] and consists in a 2-dimensional regular grid augmented by a constant number of random links per node, distributed according to the 2-harmonic distribution. Kleinberg shows that greedy routing, which chooses at each step the closest node to the target among its neighbors according to the globally known grid distance, computes paths of polylogarithmic expected length between any two nodes. Further investigations point out the general characteristics of these models, by extending it to d-dimensional tori [3], suggesting a more general model. Recently, several augmentation processes have been proposed for larger graph classes [7, 19, 4, 5], respectively bounded treewidth, bounded doubling dimension, and bounded growth graphs; results are summarized in Table 1.2. These classes are often considered as reasonable approximations of real networks. The key to these results is to create random shortcuts at all distance scales. However, these later processes are essentially centralized (in particular, they require the global knowledge of the network) and therefore could hardly be implemented in the context of large spontaneous networks. In addition, they do not provide a convincing explanation for the omnipresence of the small world phenomenon observed in real interaction networks.

Adding shortcuts to turn a large overlay network into a small-world is a recent general and efficient technique to handle queries in peer-to-peer networks. The design of a peer-to-peer overlay network aims at providing to each node a small address book such that queries as information retrieval can be done using greedy algorithms [20, 8, 17, 18]. However, these overlay networks are often strongly structured (e.g. Hypercube or DeBrujin's like) and the topological properties may be difficult to maintain.

1.1 Small world augmentation problem

We consider the following framework. We are given a graph G representing a virtual network such that one can compute for any pair of nodes u, v the distance $d(u, v)$ in G (one way is to use a distance labeling of G, i.e., each node is given a unique ID which enables one to compute the hop-distance in G between any pair of nodes, see [10, 21, 1, 2, 19]). G represents the "global" knowledge, as the 2-dimensional grid in [12]. The routing scheme studied here is the *greedy routing* that, starting from the source, repetitively moves to the neighbor of the current position that is the closest to the target according to the distance in G.

Our goal is here is to improve greedy routing performance by adding one single private random link per node in the virtual network G (i.e., one entry to each routing table/address book in the virtual network) such that the *maximum* length of the paths computed by greedy routing (using d_G) between any pair of nodes in the resulting randomly augmented network H is at most polylogarithmic in the size of the network n, with high probability, i.e., at least $1 - O(1/n)$. Such a process results in an optimized network H where greedy routing computes polylogarithmic paths between any pair of nodes without having to modify its routing procedure inherited from G, which is known to be an expensive procedure. Such an augmented network H is called a *navigable small world*. The additional random links are called *long range links*, and their destination is the *long range contact* of their origin. This raises two main questions: Can any graph be augmented into a small world? Can it be done efficiently? We address here the second question.

1.2 Our contribution

In this paper, we present the first fully distributed algorithm to augment arbitrary *bounded growth graphs*[1] into navigable small worlds in polylogarithmic time and space. Precisely, we optimize the greedy routing performance in any arbitrary bounded growth graph by running a polylogarithmic time and space distributed procedure that will only add one single entry to each address book, without recomputing the distance labeling of its nodes. The existence of such a lightweight scheme might be considered as a first step toward the explanation of the omnipresence of small world phenomenon in interaction networks. Indeed, while our process may appear unrealistic in the social networks framework, exhibiting a fully distributed augmentation process suggests the possible existence of a more general construction process that would validate Kleinberg's model as a plausible explanation for the small world phenomenon in real networks. As for the computer networks framework, the lightweight scheme proposed here might have promising applications in the design of peer-to-peer overlay networks with efficient routing.

Our process uses a sampling step to construct a relevant approximation of the network structure. Precisely, our scheme first constructs a random tree-shaped overlay network $\mathcal{T}$ which encodes efficiently a good approximation of G's metric. The tree is then used to efficiently select and add relevant long range contacts to the address books. The main result of this paper is stated as follows:

THEOREM 1. *Any bounded growth n-node network of diameter D can be augmented into a navigable small world in a distributed manner in $O(\log n \log D)$ rounds with $O(\log n \log D)$ messages of $O(\log n)$ bits induced by each node and with $O(\log N \log D)$ bits memory per node, with high probability*[2].

More precisely, with high probability, the maximum length of the paths computed by greedy routing between any pair of nodes is at most $O(\log^2 n \log^2 D)$ and the expected length is at most $O(\log n \log^2 D)$.

1.3 Related works

Previous small world augmentation processes for bounded growth graphs [19, 4] rely on pairing each node u to a node v at distance r with probability proportional to $1/b_u(r)$, where $b_u(r)$ is the number of nodes within distance r from u. If the bounded growth property ensures the success of such processes, these are however centralized and time-consumming. A first reasonable implementation has been proposed in [6] which avoids complete flooding by computing an estimation of the graph metric. In this latter scheme, nodes only explore $O(\text{polylog}\, n)$ nodes on average but $\Omega(\log n)$ nodes still explore the whole network, i.e., $\Theta(n)$ nodes.

The overlay network constructed in this paper presents some similarities with known data structures used to represent metrics of bounded doubling dimension, e.g., deformable spanners [9] or navigating nets [14]. Indeed, it similarly uses a hierarchical sampling of the nodes. However, our goal is different, for instance, we encode ball sizes

[1] A graph has *c-bounded growth* if for any node u and any radius r, $|B_u(2r)| \leqslant c|B_u(r)|$, where $B_u(r)$ is the set of nodes within distance r from u.

[2] For a n-node graph, with high probability means with probability at least $1 - O(\frac{1}{n})$.

[3] A *cluster structure* refers to a setting which includes metrics of polynomial ball growth and hierarchies, defined as *group structure* in [13].

[4] The *doubling dimension* of a metric is α if each ball of radius $2r$ can be covered by 2^α balls of radius r. Δ is the aspect ratio of the metric, i.e., the ratio of the largest distance over the smallest one.

[5] A graph has *b-moderate growth* if the ratio of a ball of radius $2r$ is at most $O(\log^b r)$ times the size of the ball of radius r and of same center, and the size of a sphere of radius r is at most $1/r$ times the size of the ball of same center and same radius. This class is included in the class of bounded growth graph, itself included in the class of bounded doubling dimension graphs.

Ref.	Underlying structure	Out-degree	Expected path length	Scheme
[7]	Treewidth k	1	$O(k \log k \log^2 n)$	Centralized description
[13]	Cluster structure[3]	$O(\log^2 n)$	$O(\log^2 n)$	
[19]	α doubling dimension[4]	$O(2^{O(\alpha)} \log n \log \Delta)$	$O(\log n)$	
[4]	b-moderate growth[5]	1	$O((\log n)^{5/2+2b})$	
this paper	bounded growth	1	$O(\log^3 n)$	Decentralized

Table 1: Small world augmentation processes.

of exponentially growing radii, and we do not require a minimum distance between nodes of the same level (the nested property), as in the r-net data structure (see [11]). In addition, as opposed to these data structures, our construction is fully decentralized (e.g., no node has a significantly higher load than another during the construction process). Note also that our goal is also different from the one in [18] which aims at *maintaining* dynamically an overlay network over dynamically changing distances between nodes measured in terms of RTT (Round Trip Time). Our overlay network is only used during the $O(\log n \log D)$ rounds of the optimization process (to compute the random long range links) and does not need to be maintained afterwards.

1.4 Outline of the paper

Section 2 describes the notations and provides the technical probabilistic lemmas that ensure the proper covering properties of our overlay network for c-bounded graphs. In Section 3, we present a distributed construction of an overlay network $\mathcal{O}(G)$ that does not require any global exploration. For any node u and radius 2^i, the overlay network provides an good estimation of $b_u(2^i)$, noted $\tilde{b}_u(2^i)$, from which we sample the long range contacts in Section 4 that will augment the graph in a navigable small world, with high probability.

2. PRELIMINARIES

2.1 Principle of our algorithm

In order to avoid expensive exploration of the graph, we build a tree-shaped overlay network which parsimoniously encodes good approximations of the sizes of the balls centered on each node with exponentially increasing radii (which is enough for our purpose). Indeed, thanks to the bounded growth, in order to pick its long range contact, each node can use the ball of radius 2^{i+1} of its ancestor at level i in the tree as a good approximation to its own ball of radius 2^i. Each node of level i approximates its ball by the union of the subtrees of its sons of level $i-1$ at distance $\leqslant 5.2^i$. Each node starts at level 0 and increases its level with some probability or stops if level $\lceil \log D \rceil$ is reached. The subtrees are disjoint and the probability to join the upper level is suitably adjusted so that the tree covers the graph with high probability, and thus the approximated ball sizes are correct up to a constant factor (which is enough for our purpose). Interestingly enough, our construction allows to maintain a constant factor approximation of balls sizes at every level; indeed, a careful selection of the subtrees to be merged, based on the relative distances of their roots in G, as well as the covering properties ensure that the errors on the estimations do not propagate to the higher levels.

2.2 Notations and model

We consider $G = (V, E)$, a n-node c-*bounded growth* graph of diameter D, where c is the *expansion rate*, i.e., the minimal constant $c > 0$ such that $b_u(2r) \leqslant c \cdot b_u(r)$ for any node u and any radius $r > 0$. For any node u and radius r, let $B_u(r)$ denote the set of nodes within distance at most r from u in G. Our small world augmentation process assigns to each node u in G, a new neighbor L_u, its *long-range contact*. In all the following, $d(u, v)$ stands for the distance in G between nodes u and v (i.e., ignoring the long range links) which is assumed to be computable from the IDs of u and v. We assume that a constant approximation of $\log n$ as well as an upper bound on c are known. Note that satisfying estimations of c and $\log n$ could be obtained by starting with an arbitrary value (1) which is increased (multiplied by 2) if the construction fails, until the construction succeeds (see [6]), combined to an appropriate failure detection scheme.

We measure the performances of our small world augmentation process in terms of time complexity (the number of rounds), of the maximum amount of memory required per node, and of the maximum number of messages induced per node. Since the algorithm is asynchronous, a *round* is defined as the time period between two consecutive messages sent by the slower node. The worst scenario is thus when all nodes work at the same speed. We assume then that sending a message takes one round in our analysis (1-port model).

2.3 Parsimonious covering sampling scheme for bounded growth graphs

The following probabilistic lemmas will be used later to design the sampling of our tree-shaped overlay network on c-bounded growth graphs. They show that using constant factor estimations $(\beta_u)_{u \in V}$ of the ball sizes allows one to sample sparse sub-networks with proper covering properties with high probability, even if the estimations are *correlated* by the construction process. The section gives technical lemmas and can be skipped during the first reading.

LEMMA 1. *Let $G = (V, E)$ be a n-node c-bounded growth graph, a radius $r > 0$, and S a random subset of vertices in which each vertex $u \in V$ is included independently with probability p_u satisfying:*

$$\min\left(1, \frac{C \log n}{b_u(r)}\right) \leqslant p_u \leqslant \frac{\alpha C \log n}{b_u(r)},$$

for some constants $C \geqslant 6c$ and $\alpha \geqslant 1$. With probability $1 - O\left(\frac{1}{n^2}\right)$, for any $u \in G$, $B_u(r)$ contains at least one element of S, and $B_u(5r)$ contains at most $2c^4 \alpha C \log n$ elements of S.

PROOF. Let $u \in G$, $r > 0$ and $B = B_u(r)$. For any $v \in B$, $B_v(r) \subseteq B_u(2r)$ and $B_u(r) \subseteq B_v(2r)$, thus $b_u(r)/c \leqslant$

$b_v(r) \leqslant c \cdot b_u(r)$. Then we have:

$$\frac{C\log n}{cb_u(r)} \leqslant p_v \leqslant \frac{c\,\alpha\,C\log n}{b_u(r)}. \quad (1)$$

The probability that $B_u(r)$ contains no vertex of S is then at most

$$\left(1 - \frac{C\log n}{cb_u(r)}\right)^{b_u(r)} \leqslant e^{-(C/c)\log n} = \frac{1}{n^{C/(c\ln 2)}},$$

which is at most $1/n^3$ since $C \geqslant 6c$. According to the union bound, the probability that it happens for at least one node u is less than $1/n^2$.

We now prove the upper bound on $|B_u(5r) \cap S|$. For any $u \in G$ and $r > 0$, the number of vertices of S in $B_u(5r)$ is the sum of independent Bernoulli trials with total expectation at least $C\log n/c$ and at most $c^4\,\alpha\,C\log n$, since $b_u(r) \leqslant b_u(5r) \leqslant c^3 b_u(r)$. Using the Chernoff bound (see, e.g., Theorem 4.1 in [16] for the version we are using here), the probability that this number is greater than $2c^4\,\alpha\,C\log n$ (i.e., at least twice its expectation) is at most

$$\left(\frac{e}{4}\right)^{C\log n/c} \leqslant \frac{1}{n^{C\log(e/4)/c}} \leqslant \frac{1}{n^3}, \quad \text{since } C \geqslant 6c.$$

From the union bound again, the probability that this happens for at least one node u is at most $1/n^2$. $\square$

During the construction process of our tree-shaped overlay network, nodes at the ith level of the tree are sampled from nodes u at the $(i-1)$th level based on the estimations $(\tilde{b}_u)$ of their ball sizes (see Algorithm 1). Since the construction process of our tree-shaped overlay network induces dependencies between the estimations $(\tilde{b}_u)$ of the ball sizes, we need to extend our lemma to a more general framework where the probabilities for the nodes to join the set S (the next level) are not independent from the construction process. The next lemma solves this issue.

LEMMA 2. *Let $G = (V, E)$ be a n-node c-bounded growth graph, a radius $r > 0$, and $\alpha \geqslant 1$ and $C \geqslant 6c$ two constants. Let $(X_u)_{u\in V}$ be a collection of independent uniform $[0, 1]$ random variables, and a collection of integer valued functions $(\beta_u)_{u\in V}$ of the X_us, such that, with high probability, for all $u \in G$,*

$$b_u(r) \leqslant \beta_u \leqslant \alpha\, b_u(r).$$

Then, for any $u \in G$, $B_u(r)$ contains at least one element of S, and $B_u(5r)$ contains at most $2c^4\,\alpha\,C\log n$ elements of S with probability $1 - O\left(\frac{1}{n^2}\right)$.

PROOF. Consider the two sets S_0 and S_1, which both satisfy conditions of Lemma 1 :

$$S_0 = \left\{v \in V : X_v \leqslant \frac{C\log n}{b_u(r)}\right\}$$
$$S_1 = \left\{v \in V : X_v \leqslant \frac{\alpha\,C\log n}{b_u(r)}\right\}$$

The conditions on $(\beta_u)_{u\in V}$ imply that $S_0 \subset S \subset S_1$ holds with high probability. The lower bound of Lemma 1 (on S_0) implies the lower bound for S, and the upper bound (on S_1) implies the upper bound for S. $\square$

3. THE OVERLAY NETWORK

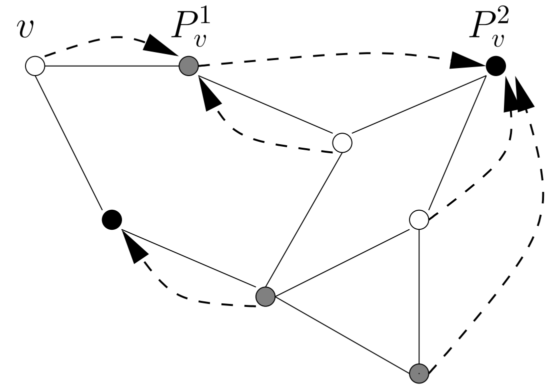

Figure 1: An example of a hierarchy of sampling for level 0 (white nodes), level 1 (grey nodes) and level 2 (black nodes). The dashed arrows show one possible overlay forest $\mathcal{T}$.

3.1 Overlay network construction

Algorithm 1 gives a detailed description of the design of the overlay network $\mathcal{O}(G)$.

Overlay network tree-like structure. The overlay network $\mathcal{O}(G)$ is composed of a tree $\mathcal{T}$ of height $\lceil\log D\rceil$ and a sequence $G_0, \ldots, G_i, \ldots, G_{\lceil\log D\rceil}$ of graphs connecting nodes of the same level. For any i, two nodes of level i are connected in G_i iff their distance in G is $\leqslant 5.2^i$. For each node u, ℓ_u denotes the highest level reached by u. u has one self-copy in the overlay network for each level $0, 1, \ldots, \ell_u$, and each copy is thus connected to further and further neighbors in the graph G_i.

Figure 3.1 illustrates nodes of increasing levels.

Overlay network at a node u. For each level $i \leqslant \ell_u$, u stores:

- its *parent*: one node P_u^i of level $i+1$ such that $d(u, v) \leqslant 2^{i+1}$ ($P_u^i = u$ if $i < \ell_u$);
- its *neighbors* in G_i: the list of nodes $N_u^i = \{w : \ell_w \geqslant i \text{ and } d(u, w) \leqslant 5.2^i\}$ along with the sizes of the subtrees T_w^i rooted on each $w \in N_u^i$. Note that u belongs to N_u^i whenever $i \leqslant \ell_u$. For convenience, we distinguish the *close neighbors* $\tilde{N}_u^i = \{w \in N_u^i : d(u, w) \leqslant 3.2^i\}$.
- its *children*: the list of nodes $C_u^i = \{v : P_v^{i-1} = u\}$.

For each $i \leqslant \ell_u$, we recursively define the tree T_u^i rooted on the copy at level i of u whose subtrees are the T_v^{i-1}, for $v \in C_u^i$.

The function INFORM(u, i) consists in flooding a message from node u to all the nodes at distance at most 3 from u in G_i using a BFS. u waits for an acknowledgment at the end of this flooding, required for u to decide whether it stays at level i.

3.2 Overlay network properties

The correctness of Algorithm 1 is guaranteed by induction, based upon the three following properties:

- $\mathcal{P}_{\mathcal{C}}(i)$, *covering property*: For each node $u \in G$, there exists a node of level i in $B_u(2^i)$ (in particular, if u is of level $i-1$, u's parent, P_u^{i-1}, is properly defined and $d(u, P_u^{i-1}) \leqslant 2^i$)
- $\mathcal{P}_{\mathcal{N}}(i)$, *neighborhood property*: For each node u of level i, all nodes of level i at distance at most $5 \cdot 2^i$

Algorithm 1 Contruction of the overlay network

INPUT: a c-bounded growth graph
for all node u do
  Choose $X_u \in [0,1]$ uniformly at random
  $T_u^0 = \{u\}$, $N_u^0 \leftarrow$ the set of nodes at distance 5 from u (BFS exploration).
  $\tilde{b}_u(1/2) = 1$, $\tilde{b}_u(1) = b_u(1)$, $i = 1$.
  while $X_u < \frac{6c^3 \log n}{\tilde{b}_u(2^i)}$ and $\tilde{b}_u(2^i) > \tilde{b}_u(2^{i-1})$ **do**
    $i \leftarrow i+1$.
    INFORM(u,i) that u is now at level i
    Wait until all the nodes at distance at most 3 hops in G_{i-1} tell if they join (or not) the next level i in order to build N_u^i and $\tilde{N}_u^i$.
    $\tilde{b}_u(2^i) = \sum_{v \in \tilde{N}_u^i} |T_v^i|$.
  $\ell_u \leftarrow i$.

  Upon reception of message Inform$(v, i+1)$ from a neighbor $v \in N_u^i$:
  if u stopped at level $\ell_u = i$ and $d(v,u) \leqslant 2^{i+1}$ **then**
    Choose v as its parent and send "$P_u^i = v$" to v. *(Ties are broken arbitrarily)*

  Upon reception of message "$P_v^i = u$" from a neighbor $v \in N_u^i$: add v to the list of children C_u^{i+1}.

from u in G are at most 3 hops away in G_{i-1} (i.e., all the neighbors of u in G_i are correctly computed).

- $\mathcal{P}_\mathcal{A}(i)$, *approximation property*: For any node u of level i, $B_u(2^i) \subseteq \bigcup_{v \in \tilde{N}_u^i} T_v^i \subseteq B_u(5.2^i)$.

Figure 2 illustrates the covering property.

LEMMA 3. *For any $i \in \{0, \dots, \lceil \log D \rceil\}$, if $\mathcal{P}_\mathcal{C}(j)$ and $\mathcal{P}_\mathcal{N}(j)$ hold for all $j \leqslant i$, then $\mathcal{P}_\mathcal{A}(i)$ holds.*

PROOF. Consider a node u of level i. Let $w \in B_u(2^i)$. If $\ell_w \geqslant i-1$, according to $\mathcal{P}_\mathcal{C}(i)$, w's parent, $v = P_w^{i-1}$, exists and $d(v,w) \leqslant 2^i$. Otherwise, an immediate induction shows that w has an ancestor v of level i such that $d(w,v) \leqslant 2^i + 2^{i-1} + \dots + 1 \leqslant 2^{i+1}$ (by triangle inequalities). Thus, in both cases, $d(u,v) \leqslant 3 \cdot 2^i$, which implies by $\mathcal{P}_\mathcal{N}(i)$ that v belongs to $\tilde{N}_u^i$ and $w \in \bigcup_{v \in \tilde{N}_u^i} T_v^i$.

Consider now $v \in \tilde{N}_u^i$. By definition, $d(u,v) \leqslant 3 \cdot 2^i$, and again according to triangle inequalities, all the nodes in T_v^i are at distance at most $2^i + 2^{i-1} + \dots + 1 \leqslant 2^{i+1}$ from v. Thus, from the triangle inequality, T_v^i is included in $B_u(5 \cdot 2^i)$. $\square$

LEMMA 4. *For any $i \in \{1, \dots, \lceil \log D \rceil\}$, if $\mathcal{P}_\mathcal{C}(i-1)$ and $\mathcal{P}_\mathcal{N}(i-1)$ hold, then $\mathcal{P}_\mathcal{N}(i)$ holds.*

PROOF. Let u, v be two nodes of level i in G such that $d(u,v) = k \leqslant 5 \cdot 2^i$. Consider $u = w_0, w_1, \dots, w_\ell = v$ a shortest path from u to v in G and set $x = w_{\lfloor \ell/2 \rfloor - 2^i}$ and $y = w_{\lceil \ell/2 \rceil + 2^i}$, such that $d(u,x)$, $d(x,y)$ and $d(y,v)$ are all at most 4.2^i. According to $\mathcal{P}_\mathcal{C}(i-1)$, there exists two nodes s_1 and s_2 of level at least $i-1$ in $B_x(2^{i-1})$ and $B_y(2^{i-1})$ respectively. Then, $d(u,s_1)$, $d(s_1,s_2)$ and $d(s_2,v)$ are all at most 5.2^{i-1} and, according to $\mathcal{P}_\mathcal{N}(i-1)$, (u,s_1,s_2,v) is a path of length at most 3 connecting u to v in G_{i-1}, which proves $\mathcal{P}_\mathcal{N}(i)$. The algorithm computes then correctly the neighbors N_u^i for all nodes u of level i. $\square$

PROPOSITION 1. *With high probability, for all $i \in \{0, \dots, \lceil \log D \rceil\}$, $\mathcal{P}_\mathcal{C}(i)$, $\mathcal{P}_\mathcal{N}(i)$ and $\mathcal{P}_\mathcal{A}(i)$ hold.*

PROOF. We prove the proposition by induction on i. $\mathcal{P}_\mathcal{C}(0)$, $\mathcal{P}_\mathcal{N}(0)$ and $\mathcal{P}_\mathcal{A}(0)$ hold with probability 1. Consider now $i > 0$, and assume that $\mathcal{P}_\mathcal{C}(j)$, $\mathcal{P}_\mathcal{N}(j)$ and $\mathcal{P}_\mathcal{A}(j)$ hold for all $j < i$. According to Lemma 4, $\mathcal{P}_\mathcal{N}(i)$ clearly holds.

According to $\mathcal{P}_\mathcal{A}(i-1)$ and the c-bounded growth, for all nodes u of level $i-1$, we have

$$b_u(2^i) \leqslant c\, b_u(2^{i-1}) \leqslant c\, \tilde{b}_u(2^{i-1}) \leqslant c\, b_u(5 \cdot 2^{i-1}) \leqslant c^3 b_u(2^i).$$

Let us define β_u as:

$$\begin{aligned} \beta_u &= c\tilde{b}_u(2^{i-1}) \quad \text{if } u \text{ has level at least } i-1 \\ &= \max(c\tilde{b}_u(2^{\ell_u}), b_u(2^i)) \quad \text{otherwise.} \end{aligned}$$

From above, for all u, we have

$$b_u(2^i) \leqslant \beta_u \leqslant c^3 b_u(2^i).$$

Consider the random set of nodes $S = \{u \in V : X_u \leqslant 6c^4 \log n / \beta_u\}$. Since, for all u of level at most $i-2$, we have

$$X_u \geqslant 6c^3 \frac{\log n}{\tilde{b}_u(2^{\ell_u})} \geqslant 6c^4 \frac{\log n}{\beta_u},$$

the set S corresponds exactly to the set of nodes of level at least i, selected during the ith iteration of the **while** loop. Applying Lemmas 1 and 2 to S with $\alpha = c^3$ and $C = 6c$, ensures that with probability $1 - O(1/n^2)$, for all nodes u, $B_u(2^i)$ contains at least one node of level i, which proves $\mathcal{P}_\mathcal{C}(i)$ w.h.p.

Finally, since $\mathcal{P}_\mathcal{C}(j)$ and $\mathcal{P}_\mathcal{N}(j)$ hold for all $j \leqslant i$ with probability $1 - O(1/n^2)$, $\mathcal{P}_\mathcal{A}(i)$ holds according to Lemma 3 w.h.p.. In order to complete the proof, note that there are at most $\lceil \log D \rceil$ iterations; the total failure probability is then at most $O(\log D / n^2)$, and $\mathcal{P}_\mathcal{C}(i)$, $\mathcal{P}_\mathcal{N}(i)$ and $\mathcal{P}_\mathcal{A}(i)$ hold for any i with probability at least $1 - O(\log D/n^2)$. $\square$

Remark. Note that property $\mathcal{P}_\mathcal{A}$ ensures that for c-bounded growth graphs, $\tilde{b}_u$ encodes efficiently a c^3-approximation for all ball sizes. Indeed, w.h.p., for each node u with $\ell_u \geqslant \lfloor \log r \rfloor + 1$, and each radius r,

$$b_u(2^{\lfloor \log r \rfloor}) \leqslant b_u(r) \leqslant b_u(2^{\lfloor \log r \rfloor + 1}) \leqslant c\, b_u(2^{\lfloor \log r \rfloor}),$$

and

$$b_u(2^{\lfloor \log r \rfloor}) \leqslant \tilde{b}_u(2^{\lfloor \log r \rfloor + 1}) \leqslant c^3 b_u(2^{\lfloor \log r \rfloor}),$$

which implies that w.h.p. for all u and r:

$$\max\left(\frac{\tilde{b}_u(2^{\lfloor \log r \rfloor + 1})}{b_u(r)}, \frac{b_u(r)}{\tilde{b}_u(2^{\lfloor \log r \rfloor + 1})}\right) \leqslant c^3.$$

Hence, our overlay network construction provides an approximation of ball sizes. Note that allowing $O(\log D)$ extra rounds of communications, one can also compute the exact ball size for any center node and radius (details are omitted here).

3.3 Time and space analysis

The following proposition shows that the construction of the overlay network is completed in polylogarithmic time and requires only polylogarithmic size memory at each node.

PROPOSITION 2. *With high probability, the computation of $\mathcal{O}(G)$ is completed after $O(c^8 \log D \log n)$ rounds*

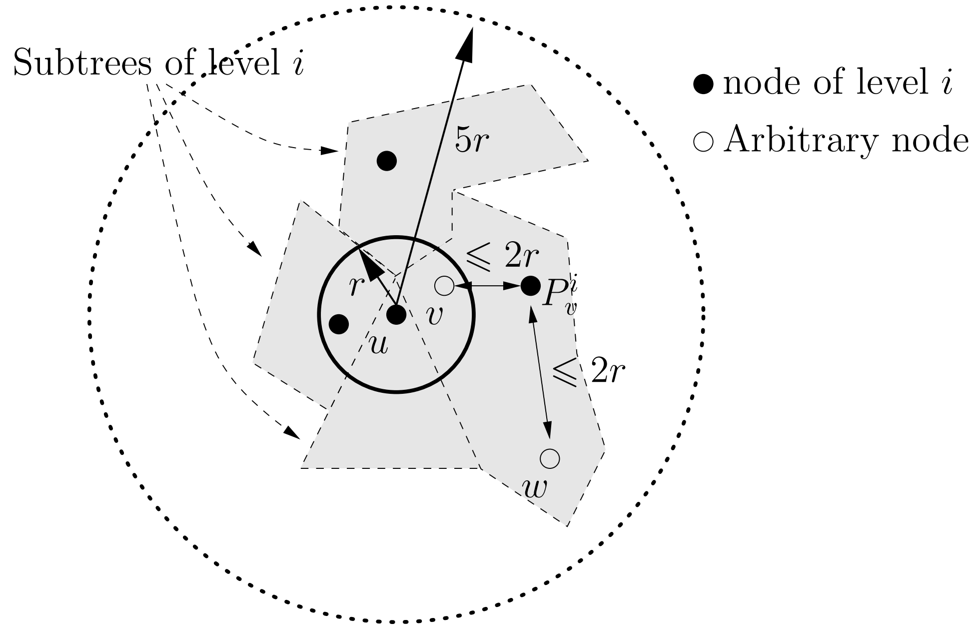

Figure 2: Balls of radius $r = 2^i$ covered by the union of subtrees (of maximal depth $2r$) rooted at close neighbors of level i.

of communications during which $O(c^8 \log n \log D)$ messages of $O(\log n)$ bits are sent per node, and requires $O(c^8 \log n \log D)$ bits of memory at each node.

Proof. Proposition 1 implies that the construction does not fail with probability $O(1 - \log D/n^2)$. At most $\lceil \log D \rceil$ iterations of the **while** loop are executed. The ith iteration consists for each node u of level i to contact its neighbors of level i and its children of level $i-1$ in G_{i-1}. According to Lemmas 1 and 2 with $\alpha = c^3$ and $C = 6c$, with high probability the number of neighbors and children for each node is at most $O(c^8 \log n)$. The ith iteration requires then at most $O(c^8 \log n)$ exchanges of messages of $O(\log n)$ bits per node of level i. The overall number of messages sent per node is then at most $O(c^8 \log n \log D)$. Finally, each node requires to store its neighbors and children IDs at each level it has reached, which requires at most $O(c^8 \log n \log D)$ bits memory. □

4. SMALL WORLD AUGMENTATION PROCESS

While the overlay network described in the previous section provides a set of hierarchical shortcuts, they are inoperative as small world shortcuts. Indeed, greedy routing only uses the original distances in G to navigate, and therefore never follows a link towards a higher level node if this node is not closer to the target in the original graph. However, the overlay network and the approximation of ball sizes obtained enable us to build efficiently, and in a distributed manner, a set of random additional shortcuts such that the augmented graph is a navigable small world.

Algorithm 2 assigns a single long range contact L_u to each node u using the distributed spanning forest

$$\mathcal{T} = \bigcup_{u\,:\,\ell_u = \lceil \log D \rceil} T_u^{\lceil \log D \rceil}$$

of the overlay network as a guideline.

In a first step (Long range contacts construction), each node u computes, for each level it has reached, a list of random nodes $\mathcal{L}_u^i$ which is sent to its descendants in T_u^i. Corollary 1 will show that this list covers all "possible directions" with high probability. That is to say that there exists a node in list $\mathcal{L}_u^i$ that allows to get twice closer to any node within radius 5.2^{i-1} from u.

In the second phase of Algorithm 2, each node u 1) picks uniformly a random length scale $i \in \{1, \ldots, \lceil \log D \rceil\}$ and 2) selects its long range contact L_u uniformly at random among nodes of $\mathcal{L}_s^i$, where s is its ancestor of level i. Lemma 6 will conclude that this correctly creates a small world network with high probability.

Lemma 5. *For all $i \geqslant 3$, for all nodes s of level at least i, and for all $v \in B_s(5.2^{i-1})$,*

$$\mathbf{Pr}\{\mathcal{L}_s^i \cap B_v(2^{i-3}) \neq \varnothing\} \geqslant 1 - 1/n^4.$$

Proof. For each $s' \in N_s^i$, $d(s,s') \leqslant 5.2^i$ and for each $w \in T_{s'}^i$, $d(s',w) \leqslant 2^{i+1}$, thus $\bigcup_{s' \in N_s^i} T_{s'}^i \subset B_s(7.2^i)$. According to triangle inequality,

$$B_s(7.2^i) \subset B_v(7.2^i + 5.2^{i-1}) \subset B_v(2^{i+4}).$$

Therefore,

$$|\bigcup_{s' \in N_s^i} T_{s'}^i| \leqslant b_v(2^{i+4}) \leqslant c^7 b_v(2^{i-3}).$$

Consider now $w \in B_v(2^{i-3})$ and w's ancestor of level i, s'. We have

$$\begin{aligned} d(s,s') &\leqslant d(s,v) + d(v,w) + d(w,s') \\ &\leqslant 5.2^{i-1} + 2^{i-3} + 2^{i+1} \leqslant 5.2^i, \end{aligned}$$

and thus $s' \in N_s^i$. We conclude that $B_v(2^{i-3}) \subset \bigcup_{s' \in N_s^i} T_{s'}^i$.

Algorithm 2 Augmentation process

for all node u **and all** level $i \leqslant \ell_u$ **do**

Long range contacts sampling:

Select a list $\mathcal{L}_u^i$ of $4c^7\lceil\log n\rceil$ leaves, chosen uniformly and independently at random in the forest $\bigcup_{v\in N_u^i} T_v^i$ of the subtrees rooted on u's neighbors in N_u^i. *(This is accomplished distributively by sending $4c^7\lceil\log n\rceil$ messages that independently recursively go down the trees with probability proportional to the size of the subtrees until a leaf is reached which sends its identity back to u)*

Recursively spread the list $\mathcal{L}_u^i$ down to each of its descendants in T_u^i.

Long range contact assignment:

Pick a random level $j \in \{1, \ldots, \lceil\log D\rceil\}$ uniformly at random.

Pick uniformly at random u's long-range contact L_u in the list $\mathcal{L}_s^i$, where s is u's ancestor of level j.

Since s samples in $\mathcal{L}_u^i$, $16c^7\lceil\log n\rceil$ nodes uniformly at random in the forest $\bigcup_{s'\in N_s^i} T_{s'}^i$,

$$\mathbf{Pr}\{\mathcal{L}_s^i \cap B_v(2^{i-3}) = \varnothing\} \leqslant \left(1 - \frac{b_v(2^{i-3})}{|\bigcup_{s'\in N_s^i} T_{s'}^i|}\right)^{4c^7\lceil\log n\rceil} \leqslant \left(1-\frac{1}{c^7}\right)^{4c^7\lceil\log n\rceil} \leqslant \frac{1}{n^4}.$$

□

Since the maximum level is $\lceil\log D\rceil$ for each node, we get the following corollary from the union bound.

Corollary 1. *With probability at least $1-O(\log D/n^2)$, for all $i \geqslant 3$, for all nodes s of level at least i, and for all $v \in B_s(5.2^{i-1})$, $B_v(2^{i-3})$ contains a point of $\mathcal{L}_s^i$.*

We now show that for each node u, u's long range L_u is twice closer than u to a given target v, with polylogarithmic probability.

Lemma 6. *With probability $1-O(\log D/n^2)$, for all pair of nodes (u, v), if $i \geqslant 3$ is such that $2^{i-2} < d(u,v) \leqslant 2^{i-1}$, then*

$$\mathbf{Pr}\{L_u \in B_v(2^{i-3})\} \geqslant \frac{1}{4c^7\lceil\log n\rceil\lceil\log D\rceil}.$$

Proof. Let s be u's ancestor of level i. $d(v,s) \leqslant d(v,u) + d(u,s) \leqslant 2^{i-1} + 2^{i+1} \leqslant 5.2^{i-1}$. By Corollary 1, it follows that with probability $1-O(\log D/n^2)$, for all such pairs, $\mathcal{L}_s^i \cap B_v(2^{i-3}) \neq \varnothing$. Conditioned on this event, the probability that $L_u \in B_v(2^{i-3})$ is then at least the probability that u picked level i (probability $1/\lceil\log D\rceil$) and that L_u has been picked among nodes of $\mathcal{L}_s^i$ (probability $1/|\mathcal{L}_s^i|$), i.e., at least $\frac{1}{4c^7\lceil\log n\rceil\lceil\log D\rceil}$. □

Lemma 6 provides the key element to ensure the efficiency of greedy routing in the augmented network, as shown in the following theorem.

Theorem 2. *The augmented network is a navigable small world, i.e., with probability at least $1-O(\log D/n^2)$, for any pair of nodes (u, v), greedy routing computes a path of length at most $16c^7\lceil\log n\rceil^2\log(d(u,v))\lceil\log D\rceil = O(\log^4 n)$, and $16c^7\lceil\log n\rceil\log(d(u,v))\lceil\log D\rceil = O(\log^3 n)$ on expectation.*

Proof. Assume, with probability $1-O(\log D/n^2)$, that the conclusion of Lemma 6 holds. Consider a pair of source and target (u, v) for greedy routing. Let $i \geqslant 3$ such that $2^{i-2} < d(u,v) \leqslant 2^{i-1}$. From Lemma 6,

$$\mathbf{Pr}\{L_u \in B_v(2^{i-3})\} \geqslant 1/(4c^7\lceil\log n\rceil\lceil\log D\rceil).$$

If $L_u \notin B_v(2^{i-3})$, greedy routing gets to a node u' still satisfying $2^{i-2} < d(u',v) \leqslant 2^{i-1}$. Then, the probability that no node with a long range contact in $B_v(2^{i-3})$ was visited after the first $16c^7\lceil\log n\rceil^2\lceil\log D\rceil$ steps of greedy routing is at most:

$$\left(1 - \frac{1}{4c^7\lceil\log n\rceil\lceil\log D\rceil}\right)^{16c^7\lceil\log n\rceil^2\lceil\log D\rceil} \leqslant \frac{1}{n^4}.$$

This upper bound is independent of the current distance to v. Hence, $\log d(u,v)$ times, the current distance to v is divided by at least 2 in $16c^7\lceil\log n\rceil^2\lceil\log D\rceil$ steps with probability greater than

$$\left(1-\frac{1}{n^4}\right)^{\log d(u,v)} \geqslant 1 - \frac{2\log d(u,v)}{n^4},$$

for $n \geqslant n_0$. Thus, the expected number of steps of greedy routing from u to v is at most $16c^7\lceil\log n\rceil^2\lceil\log D\rceil\log d(u,v)$ with probability greater than $1 - 2\lceil\log D\rceil/n^4$. Applying the union bound, we conclude that with probability at least $1-O(\log D/n^2)$, the maximum length of the greedy path between any pair of nodes (u, v) is at most

$$16c^7\lceil\log D\rceil\lceil\log n\rceil^2\log d(u,v).$$

To get the expectation result, note that, given the pair (u, v), greedy routing visits a node with a long range contact in $B_v(2^{i-3})$ after the first $16c^7\lceil\log n\rceil\log D$ steps with probability greater than a constant. The expected path length is thus $16c^7\lceil\log n\rceil\lceil\log D\rceil\log d(u,v)$ following the same dichotomic argument as above. □

Analysis of Algorithm 2 performances combined to Proposition 2 leads to the following overall performances of our small world augmentation process.

Theorem 3. *Any c-bounded growth n-node graph can be augmented into a navigable small world in a distributed way w.h.p. in $O(c^8\log n\log D)$ rounds, $O(c^8 n\log n\log D)$ messages, and requiring $O(c^8\log n\log D)$ bits memory in each node.*

5. CONCLUSION

One point that remains unaddressed in Kleinberg's model is an explanation of the emergence of the 2-harmonic long range links distribution in the grid. In a sense, our decentralized algorithm is light and fast enough to be considered as a first step towards a validation of Kleinberg's model for real interaction networks. Indeed our algorithm is decentralized, applies to arbitrary bounded growth graphs and requires only polylogarithmic size memory and time complexity. Although our description assume that all the nodes try simultaneously to make a small-worldization process, it can easily be adpated in such a way that a single node decides to start the process with the same performance. Note also that the assumption of an estimation of c and $\log n$ can be bypassed thanks to successive trials, similarly as in [6]. Finally, note that an expansion rate $c = \text{polylog}\, n$ does not change the performances drastically (time, space and message complexity are still polylogarithmic in the size of the network) and our scheme is still an efficient distributed small world augmentation process for polylog(n)-bounded growth graphs.

6. REFERENCES

[1] I. Abraham, C. Gavoille, D. Malkhi, N. Nisan, and M. Thorup. Compact name-independent routing with minimum stretch. In 16^{th} *ACM Symp. on Parallel Alg. and Archi. (SPAA)*, pages 20–24, 2004.

[2] I. Abraham and D. Malkhi. Name independent routing for growth bounded networks. In 17^{th} *Annual ACM Symposium on Parallel Algorithms and Architecture (SPAA)*, pages 49–55, 2005.

[3] L. Barrière, P. Fraigniaud, E. Kranakis, and D. Krizanc. Efficient routing in networks with long range contacts. In *Proc. of 15th Inter. Symp. on Distr. Comp. (DISC)*, volume 2180, pages 270–284, 2001.

[4] P. Duchon, N. Hanusse, E. Lebhar, and N. Schabanel. Brief announcement: Could any graph be turned into a small world? *Proc. of the 19th Inter. Symp. on Distr. Comp. (DISC)*, pages 511–513, 2005.

[5] P. Duchon, N. Hanusse, E. Lebhar, and N. Schabanel. Could any graph be turned into a small world? *Theoretical Computer Science*, 355(1): 96–103, 2006.

[6] P. Duchon, N. Hanusse, E. Lebhar, and N. Schabanel. A fully distributed scheme to turn a network into a small world. Technical Report RR2006-03, LIP - ENS Lyon, 2006.

[7] P. Fraigniaud. Greedy routing in tree-decomposed graphs: a new perspective on the small-world phenomenon. In *Proc. of the 13th European Symposium on Algorithms (ESA)*, pages 791–802, 2005.

[8] P. Fraigniaud and P. Gauron. Brief annoucement: an overview of the content-addressable network d2b. In *Proc of the ACM 22st Symp. on Princ. of Distr. Comp. (PODC)*, page 151, 2003.

[9] J. Gao, L. J. Guibas, and A. Nguyen. Deformable spanners and applications. In *Proc. of the 20th ACM Symp. on Computational Geometry (SoCG)*, pages 190–199, 2004.

[10] C. Gavoille and D. Peleg. Compact and localized distributed data structures. *Journal of Distributed Computing*, 16:111–120, 2003. PODC 20-Year Special Issue.

[11] S. Har-Paled and M. Mendel. Fast construction of nets in low dimensional metrics and their applications. In *Proc. of the 21th ACM Symp. on Computational Geometry (SoCG)*, 2005.

[12] J. Kleinberg. The Small-World Phenomenon: An Algorithmic Perspective. In *Proceedings of the 32nd ACM Symposium on Theory of Computing (STOC)*, pages 163–170, 2000.

[13] J. Kleinberg. Small-world phenomena and the dynamics of information. In *Advances in Neural Information Processing Systems (NIPS) 14*, 2002.

[14] R. Krauthgamer and J. R. Lee. Navigating nets: simple algorithms for proximity search. In *Proc. of the 15th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 798–807, 2004.

[15] S. Milgram. The small world problem. *Psychology Today*, 61(1), 1967.

[16] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[17] S. Ratsanamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. In *Proceedings of the ACM SIGCOMM*, pages 161–170, 2001.

[18] E. G. Sirer, A. Slivkins, and B. Wong. Meridian: A lightweight network location service without virtual coordinates. In *Proceedings of the ACM SIGCOMM*, 2005.

[19] A. Slivkins. Distance estimation and object location via rings of neighbors. In 24^{th} *ACM Symp. on Principles of Distributed Computing (PODC)*, pages 41–50, 2005.

[20] I. Stoica, R. Morris, D. R. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proc. of the ACM SIGCOMM*, pages 149–160, 2001.

[21] M. Thorup and U. Zwick. Approximate distance oracles. *Journal of the ACM*, 52(1):1–24, 2005.