Compact Routing for Graphs Excluding a Fixed Minor

(Extended Abstract)

Ittai Abraham[1], Cyril Gavoille[2], and Dahlia Malkhi[3]

[1] School of Computer Science and Engineering,
The Hebrew University of Jerusalem, Jerusalem, Israel.
ittaia@cs.huji.ac.il
[2] Laboratoire Bordelais de Recherche en Informatique,
University of Bordeaux, Bordeaux, France.
gavoille@labri.fr
[3] School of Computer Science and Engineering,
The Hebrew University of Jerusalem, Jerusalem, Israel,
and Microsoft Research, Silicon Valley Center.
dalia@microsoft.com

Abstract. This paper concerns compact routing schemes with arbitrary node names. We present a compact name-independent routing scheme for unweighted networks with n nodes excluding a fixed minor. For any fixed minor, the scheme, constructible in polynomial time, has constant stretch factor and requires routing tables with poly-logarithmic number of bits at each node.

For shortest-path labeled routing scheme in planar graphs, we prove an $\Omega(n^{\epsilon})$ space lower bound for some constant $\epsilon > 0$. This lower bound holds even for bounded degree triangulations, and is optimal for polynomially weighted planar graphs ($\epsilon = 1/2$).

1 Introduction

Consider a distributed network of nodes connected via a network in which each node has an arbitrary network identifier. A routing scheme allows any source node to route messages to any destination node, given the destination's network identifier. The fundamental trade-offs in compact routing schemes is between the *space* used to store the routing table on each node and the *stretch* factor of the routing scheme, the maximum ratio over all pairs between the length of the route induced by the scheme and the length of a shortest-path between the same pair.

The trivial solution to routing on shortest paths (stretch factor 1) is for each node to store a routing table with $\Omega(n)$ entries that contains the next hop of an all pairs shortest-path algorithm. This solution is very expensive as it requires each node to store $\Omega(n \log n)$ bits. Thus, network designers are faced with two conflicting goals: reduce both the stretch factor and the size of the routing tables.

In this paper we assume a network with arbitrary node names. This model is called the *name-independent* model because the designer of the routing scheme has no control over node names. So node names cannot encode any topological awareness, like for instance the X, Y-coordinates in a geographic network. This routing problem may appear daunting: In order to route to a node, we must first somehow gain knowledge about its location in the network, but, in order to have some guarantees on the stretch factor, we must do so without exceeding too much the distance to the target.

A weak variant of this fundamental problem is called *labeled routing*. In this version of the problem, the designer of a solution may pick node names that contain (bounded size) information about their location in the network. This variant is useful in many aspects of network theory, but less so in practice: Knowledge of the labels needs to be disseminated to all potential senders, as these labels are not the addresses by which nodes of an *existing* network, e.g., an IP network, are known. Furthermore, if the network may admit new joining nodes, all the labels may need to be re-computed and distributed to any potential sender. Finally, various recent applications pose constraints on nodes addresses that cannot be satisfied by existing labeled routing schemes. E.g., Distributed Hash Tables (DHTs) require nodes names in the range $[1, n]$, or ones that form a binary prefix.

There is a subtle distinction between a *designer-port* model and a *fixed-port* model. In the *fixed-port* model (also known as the adversarial port model) the names of outgoing links, or ports, from each node may be arbitrarily chosen by an adversary from the set $\{1, \ldots, n\}$. In the *designer-port* model they may be determined by the designer of the routing scheme. Our routing scheme applies to the fixed-port model.

In this paper we are interested in the design of fixed-port name-independent compact routing schemes with low space and stretch, typically with $O(1)$ stretch and $\widetilde{O}(1)$ memory per node[4]. Unfortunately, it is known that, even for the labeled variant, any routing scheme of stretch $O(k)$ applying on all graphs requires $\Omega(n^{1/k})$ bit memory in the worst-case [32,36]. Identifying large realistic families of networks supporting low stretch and memory name-independent routing schemes is a wide open question. Here, we restrict our attention to families of graphs excluding a fixed minor, so including all families closed under taking minors (by The Minor Theorem of Robertson & Seymour). For instance, it includes all graphs of bounded treewidth or bounded genus.

A graph H is a *minor* of G if H is a subgraph of a graph obtained by a series of edge contractions[5] of G. The study of graphs excluding a fixed minor has lead to fundamental graph theory results. In the context of routing, several natural classes of networks can be defined by their forbidden minor. Among them are

---

[4] The notation $\widetilde{O}(\cdot)$ indicates complexity similar to $O(\cdot)$ up to poly-logarithmic factors.

[5] The *contraction* of the edge e with endpoints u, v is the replacement of u and v with a single node whose incident edges are the edges other than e that were incident to u or v [37].

trees [27] (excluding K_3) and *series parallel* networks [18] (excluding K_4) that capture many network backbone structures, and *planar graphs* (excluding K_5 or $K_{3,3}$) that capture the structure of two dimensional maps.

1.1 Related work

The space-stretch trade-off has been extensively studied under various models and extensions. We refer the reader to Peleg's book [30] and to the surveys of Gavoille and Peleg [20,22] for comprehensive background.

There is a large body of work on special families of graphs. For the labeled model: graphs with bounded treewidth [31], bounded chordality [14,15,16] (i.e., graphs for which every induced cycle is bounded), and more recently, graphs (or metrics) with bounded doubling dimension [10,24,33,34] (i.e., graphs for which any radius-$2r$ ball can be covered by a bounded number of radius-r balls). For the name-independent model: trees [1,27], graphs with bounded growth dimension [5]. Observe that planar graphs are not captured by any of these families.

No low memory and stretch name-independent routing schemes are known for planar graphs, however several schemes have been proposed for this family for the labeled variant.

Surprisingly, for stretch 1, the complexity of the size of the routing tables is not known. The best scheme up to date has been proposed by Lu [28]: the node labels range in $[1, n]$, the routing tables are of $7.181n + o(n)$ bits (per node), and each routing decision takes $O(\log^{2+\epsilon} n)$ bit operations for every fixed $\epsilon > 0$. Earlier, it was proved that, actually, genus-g graphs support $n \log g + O(n)$ bit routing tables [21], that provides $8n + o(n)$ bits in the planar case. They also showed in [21] that if each node is forced to route along a shortest-path tree fixed by an adversary, then any routing scheme requires $\Omega(n)$ bits in the worst-case (this actually extends to $\Omega(n \log r)$ bits for a $K_{r,n-r}$, so setting $r = 2$ for planar). However, if the designer of the scheme has freedom to optimize the routes among those of equal costs, then only the $\Omega(\sqrt{n})$ bit lower bound for trees applies [17]. Furthermore, this lower bound concerns only schemes that assign node labels in $[1, n]$ exactly. Better results exist for some particular subclass of planar graphs. In [12], it is proved that quadrangulations[6] where all inner nodes have degree at least 4 (these include the subgraphs of a grid bounded by a circuit) have a labeled routing scheme with $O(\deg(v) \log n)$ bit memory for each node v. For other particular plane graphs, namely the non-positively curved plane graphs, a labeled $O(\log^2 n)$ bit routing scheme exists [11].

To summarize, if poly-log node labels are allowed, then no trivial lower bound on the memory is known for the shortest-path labeled routing scheme problem in general planar graphs, whereas $O(n)$ is the upper bound [21,28].

[6] I.e., plane graphs where all the inner faces are of length 4.

The situation is however quite different if routing schemes with stretches > 1 are considered. For instance routing in Euclidian plane graphs[7] is investigated in [8,9]. It is shown that plane triangulations having the diamond property (which is the case for classical triangulations as the Delaunay, greedy and minimum-weight triangulations) have a constant stretch labeled routing scheme where node labels are the coordinates and the memory requirement of each node v consists only in the coordinates of v and of its neighbors, therefore is $O(\deg(v)\log n)$ bits. The same results hold for all plane graphs possessing both the diamond property and the good convex polygon property. For general planar graphs, Frederickson and Janardan have presented in [19] two schemes. The first one achieve $O(n^{4/3}\log n)$ bits in total (the sum of the routing table size over all the nodes), stretch 3, and uses $O(\log n)$ bit node labels. The second one uses $O((1/\epsilon)n^{1+\epsilon}\log n)$ bit in total, for every fixed $\epsilon > 0$, stretch 7, and $O((1/\epsilon)\log n)$ bit node labels. Recently, Thorup [35] improves the stretch bound thanks to an extension of his *distance oracles* for planar graphs. He obtained a labeled routing scheme with stretch $1+\epsilon$ in which for a fixed $\epsilon > 0$, routing tables and node labels have $O((1/\epsilon)\log^2 n)$ bits. For Euclidean metrics, Hassin and Peleg show in [25] a labeled scheme with $O(\log n)$ out going edges per node, $O(\log n)$-hop routes, and $1+\epsilon$ stretch. The out degree is further reduced to a constant by Abraham and Malkhi [4].

1.2 Our contributions

Our first contribution is a name-independent routing schemes for unweighted graphs excluding a fixed minor. We prove the following.

Theorem 1. *For every n-node unweighted graph excluding a fixed $K_{r,r}$ minor, there is a polynomial time constructible name-independent routing scheme with constant stretch factor, in which every node v requires routing tables of $\widetilde{O}(1)$ bits and $O(\log^2 n/\log\log n)$-bit headers.*

The general result follows since graphs excluding a $K_{r,r}$ minor exclude K_{r+1} minor and thus exclude any fixed graph H on $r+1$ nodes. Note that the result was not known even for trees, i.e., $H = K_3$. For $H = K_{3,3}$, it is important to observe that the $\widetilde{O}(1)$ memory $1+\epsilon$ stretched labeling routing scheme of Thorup [35] for planar graphs cannot be extended to name-independent scheme since in that case a stretch of 3 at least is required if less than $\Omega(n\log n)$ bits are used [3].

Our scheme is based on a cover for graphs excluding a $K_{r,r}$ minor which is a novel variant of the Klein, Plotkin, and Rao [26] partitioning algorithm. While the scheme in [26] gives a partition algorithm with weak diameter bounds, no information is given about the structure of the short paths that bound the diameter. Our covering algorithm gives an explicit structure to the short paths

[7] I.e., a planar graphs embedded in the plane whose edges are weighted by the Euclidean distance between their endpoints.

formed. This explicitness may be of independent interest and is presented in Section 3. Moreover, we show how to utilize this structure for an efficient routing scheme in Section 4.

Our second contribution is a lower bound. As said previously in the introduction, no (trivial) lower bounds are known for stretch 1 labeled routing schemes in planar graphs. Based on the distance labeling lower bound of [23], we prove in Section 5 an $\Omega(n^{\epsilon})$ lower bound on the label length (the length of the local routing tables plus the node label length). This bound holds even in the quite simple case: a bounded degree triangulation.

Theorem 2. *Every shortest-path labeled routing scheme on polynomially weighted n-node planar graphs of bounded degree requires a total label length (the length of all the routing tables and node labels in the graph) of $\Omega(n^{1+1/2})$ bits. Moreover, the (maximum) label length is $\Omega(n^{1/4})$ for some weighted bounded degree triangulations. For unweighted planar graphs, the two bounds are respectively $\Omega(n^{1+1/3})$ and $\Omega(n^{1/6})$.*

Observe that there is a labeled scheme using only $\widetilde{O}(\sqrt{n}\,)$ bit node labels for weighted[8] graphs of treewidth $O(\sqrt{n}\,)$ [31] (thus including planar graphs). So, in a sense Theorem 2 is optimal. However, no shortest-path routing scheme is known to achieve $\widetilde{O}(\sqrt{n}\,)$ routing tables but with $\widetilde{O}(1)$ node labels.

1.3 Outline of techniques

Awerbuch and Peleg introduced *sparse covers* [6] in order to build a hierarchal routing scheme for general graphs [7]. Their scheme is based on tree covers with geometrically increasing radii. Typically each node of the graph belongs to $O(kn^{1/k})$ trees of the cover, where $k \geq 1$ is a parameter. For each radius ρ and source node s, the ball of radius ρ centered at s is contained in some trees of radius $O(k\rho)$. Roughly speaking, the routing task for s consists in seeking the target t in each tree it belongs to. However, this high level description hides many difficulties to implement, and tree routing plays an important role [6,7,1].

The situation for graphs excluding a fixed minor is quite different. It is currently an open question to design a sparse cover for such graphs with trees (or clusters) of strong radius $O(\rho)$ so that each node belongs to $O(1)$ such clusters. Klein, Plotkin, Rao [26] achieve sparse covers in which clusters have only *weak-diameter* $O(\rho)$, i.e., the shortest paths between any two nodes of a cluster can go outside the cluster itself. For our application, it means that a node v might potentially be involved, for the routing between many pairs of nodes, in several clusters that do not contain v. Therefore the naive approach of using [7] with a cover based on [26] may require $\Omega(n)$ nodes to participate in routing for $\Omega(n)$ clusters and hence require $\Omega(n^2)$ bits in total.

Instead, we present a novel partitioning algorithm for graphs excluding a $K_{r,r}$ minor for constant r. Our cover borrows from [26] but has some subtle

[8] With polynomial edge weights.

differences. Using a new analysis we prove explicit properties on the structure of short paths between nodes in the same cluster. Our decomposition creates a set of clusters and trees, such that each node belongs to a constant number of clusters and trees. For every u, v of a same cluster, we construct a path connecting them of length $O(\rho)$ that is "well-structured". It decomposes in at most r particular subpaths, called *tail-connections*: each tail-connection is an upward path towards a root of one tree and a downward path on another tree (see Section 3).

Given this partition, we create an intricate rooted tree for each cluster that takes advantage of the cover properties. Finally, we present a single-source routing scheme in the fixed-port model in order to route on these trees. Our scheme uses the unweighted designer-port construction of [1,2] in a non-trivial manner over several overlapping trees in order to route efficiently both inside and outside of the cluster, and some recent results from graph minor theory [13] (see Section 4).

Due to space constrains some proofs of this extended abstract have been moved in the full version.

2 Preliminaries

Consider an unweighted connected graph $G = (V, E)$ with n nodes. Each node has an arbitrary unique network identifier consisting of $\mathrm{polylog}(n)$ bits. Using standard hashing techniques it is possible to generalize the model and assume nodes have arbitrarily long unique labels.

For a set of nodes $U \subseteq V$, let $G[U]$ be the subgraph induced by U. Given a subgraph $H \subset G$, let $d_H(u, v)$ denote the length of a shortest path in H from u to v. Given $v \in V$, let $d_H(v, U) = \min_{u \in U} d_H(u, v)$, and let $B_H(v, \rho) = \{u \in H \mid d_H(u, v) \leq \rho\}$. We denote by $\mathrm{diam}(H) = \max_{u,v \in H} d_H(u, v)$ the (strong) diameter of H, and $\deg_H(v)$ the degree of v in H.

For an index $j \geq 0$ and a rooted tree T with root $\tau \in T$ let $C(T, j) = \{v \in T \mid j\rho \leq d_T(v, \tau) < (j + 1)\rho\}$, specifically in our construction we will use j's such that either $j \in \mathbb{N}$ or $j + 1/2 \in \mathbb{N}$.

Let T be a tree with root τ, a node $v \in T$, and a number k. The *tail* of v of length k on T, denoted by $\mathrm{tail}(T, v, k)$, is the path of length $\min\{k, d_T(v, \tau)\}$ from v towards τ on T.

For a node $\tau \in G$, let $\mathrm{BFS}(G, \tau)$ be a breadth-first search tree of G rooted at τ. Let $r \in (N)$ be a parameter, and let $R = \{1, 2, \ldots, r\}$.

Definition 1. *(Tail-connected) Given a parameter $\rho \in \mathbb{N}$, cluster $H \subseteq G$, and a collection of rooted trees $\mathcal{T}$, two nodes $u, v \in H$ are* tail-connected *if there are trees $T_1, T_2 \in \mathcal{T}$ such that* $\mathrm{tail}(T_1, u, (r + 2)\rho + 1)$ *and* $\mathrm{tail}(T_2, v, (r + 2)\rho + 1)$ *intersect.*

Definition 2. *(r-Tail-Connected) Nodes $u, v \in H$ are* r-tail-connected *if there exists $2r-2$ nodes $u = x_1, x_2, \ldots, x_{2r-1}, x_{2r-2} = v \in H$ (not necessarily distinct)* *such that for any even i, $d_H(x_i, x_{i+1}) \leq 1$, and for any odd i, x_i, x_{i+1} are tail-connected.*

3 The Weak Diameter Cover

Theorem 3. *For every graph $G = (V, E)$ excluding a $K_{r,r}$ minor, and parameter $\rho > 0$, there exists a polynomial algorithm that constructs a collection of connected components $\mathcal{H}_\rho$ and a collection of trees $\mathcal{T}_\rho$ such that:*

1. *(Cover) For every $v \in V$, there exists $H \in \mathcal{H}_\rho$ such that $B_G(v, \rho/4) \subseteq H$.*
2. *(Sparse clusters) For every $v \in V$, $|\{H \in \mathcal{H}_\rho \mid v \in H\}| \leq 2^r$.*
3. *(Sparse trees) For every $v \in V$, $|\{T \in \mathcal{T}_\rho \mid v \in T\}| \leq 2^{O(r \log r)}$.*
4. *(Weak diameter) For every $H \in \mathcal{H}_\rho$, and every $u, v \in H$, the nodes u, v are r-tail-connected with trees in $\mathcal{T}_\rho$.*

The rest of the section is devoted to the proof of Theorem 3. We first describe a *partitioning algorithm* of a graph G that depends on a parameter $\rho > 0$. It returns a cluster H and implicitly graphs G_i, H_i and trees T_i, for $i \in R$.

Actually, in order to create a partition of the graph into clusters, we apply this algorithm for all possible choices of indices $j_i \geq 0$ (see also the proof of Theorem 3).

Partitioning algorithm: Initially $H_1 = G_1 = G$. Given G_i and H_i set a root $\tau_i \in H_i$ and choose an index j_i. Let $T_i = \mathrm{BFS}(G_i, \tau_i)$. Let H_{i+1} be a connected component of the subgraph graph induced by the nodes

$$H_i \cap C(T_i, j_i)$$

Let G_{i+1} be the connected component of

$$G_i \cap \bigcup_{\ell \in \{j_i - i, \ldots, j_i, \ldots, j_i + i\}} C(T_i, \ell)$$

that contains H_{i+1}. If $i = r$ then return $H = H_{i+1}$ and stop. Otherwise repeat.

For the analysis it will be convenient to use the following notation: for all $u \in G_i$ and $i \in R$, let $\mathrm{tail}_i(u) = \mathrm{tail}(T_i, u, (i+1)\rho + 1)$.

Lemma 1. *Let H be the graph returned after the partitioning algorithm and assume that $j_i > r + 1$ for each $i \in R$. Let $X \subseteq H$ be any r nodes of H. If for each $x \neq x' \in X$ and $i, i' \in R$ the tails $\mathrm{tail}_i(x)$ and $\mathrm{tail}_{i'}(x')$ are pairwise disjoint then G contains a $K_{r,r}$ minor.*

Proof. In order to show that G contains a $K_{r,r}$ minor we construct two sets, each one containing, r connected subgraphs, each subgraph called hereafter left or right super node. The super nodes are chosen to be pairwise disjoint and

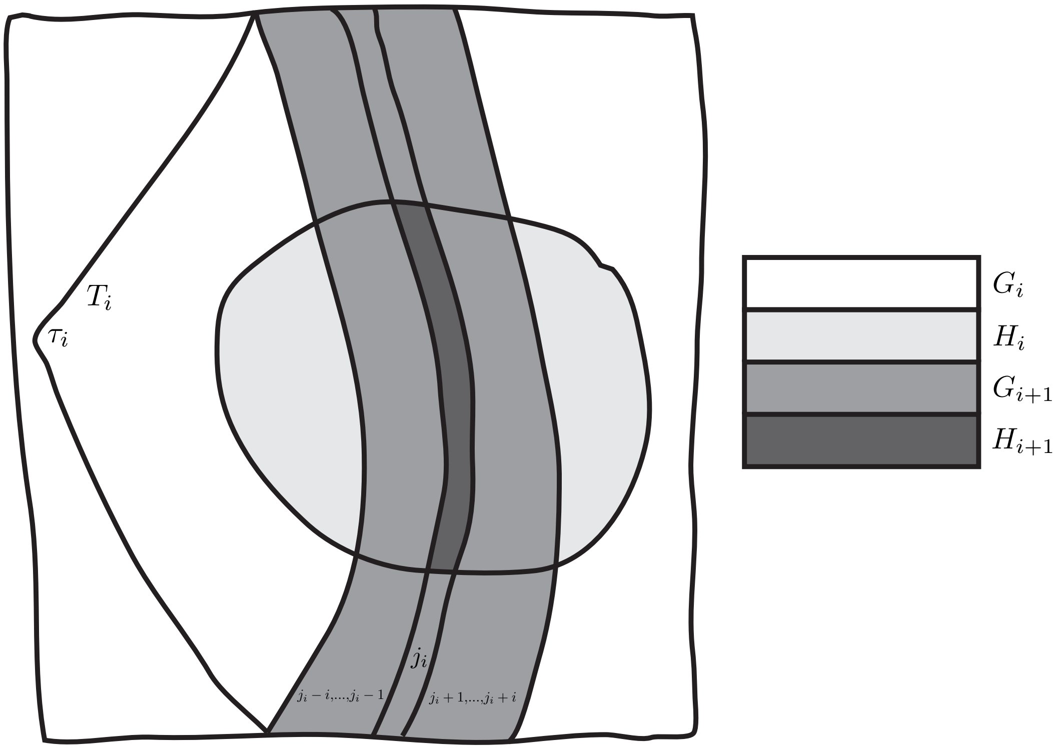

Fig. 1. A schematic drawing of one stage in the algorithm. Note that the paths of T_i from τ_i to nodes in H_{i+1} may use some nodes in G_i outside H_i.

such that each left-right pair of super nodes are connected by an edge with one endpoint in each super node. Then, by contracting all the edges of all the super nodes, and then keeping theses nodes and the left-right edges, it will prove that G has a $K_{r,r}$ minor.

For $x \in X$, let $L_x = \bigcup_{i \in R} \mathrm{tail}_i(x)$. The left super nodes of the $K_{r,r}$ will be the sets $\{L_x \mid x \in X\}$. Observe that each left super node indeed induces a connected subgraph and that theses super nodes are pairwise disjoint by the assumption.

For the right super nodes, for all $i \in R$ let U_i denote the subtree of T_i formed by the paths from each $x \in X$ to τ_i the root of T_i. The right super nodes will be the V_i for $i \in R$, where $V_i = U_i \setminus \bigcup_{x \in X} \mathrm{tail}_i(x)$. Observe that each V_i induces a connected subtree of G. Since $j_i > r + 1$ for all $i \in R$ and $X \subseteq H_{i+1} \subseteq C(T_i, j_i)$ then $|\mathrm{tail}_i(x)| = (i+1)\rho + 1$ hence $d_{G_i}(X, V_i) > i\rho$ for all $i \in R$.

The super edges will be the edges in T_i connecting each V_i with each $\mathrm{tail}_i(x) \in L_x$ for each $i \in R, x \in X$. Since all tails, for distinct $x \in X$, are disjoint, we are left only with showing that all V_i's are pairwise disjoint and disjoint from all the tails.

To do so, we prove that for every $x \in X$ and $i \in R$, the path on T_i from x to τ_i is disjoint from V_j and $\{\text{tail}_j(y) \mid y \in X \setminus \{x\}\}$ for all $i < j$. Seeking a contradiction, assume that for some $x \in X$ and $i \in R$ the path on T_i from x to τ_i intersects some node $v \in G_j$ which is part of another super node (either $v \in V_j \subseteq G_j$ or $v \in \text{tail}_j(y) \subseteq G_j$ for $y \neq x$) for some $i < j$.

Recall that $d_{G_j}(x, V_j) > j\rho$. To arrive to a contradiction we will show that: (1) $v \in V_j$, (2) the path P on T_i from x to v has length $\leq (i+1)\rho$, (3) by induction, that for $i \leq \ell \leq j$ we have $P \subseteq G_\ell$. For $\ell = j$ this is a contradiction since $i < j$ and it implies that $d_{G_j}(x, v) \leq j\rho$.

Observe that $v \in G_j \subseteq \bigcup_{\ell \in \{j_i - i, \ldots, j_i, \ldots, j_i + i\}} C(T_i, \ell)$. Since T_i is a BFS tree then the path from x to the root τ_i on T_i can intersect v only if $v \in \bigcup_{\ell \in \{j_i - i, \ldots, j_i\}} C(T_i, \ell)$ hence $d_{T_i}(x, v) \leq (i+1)\rho$. This completes the first induction base for $\ell = i$ of (3). This also implies that the path P must be a sub-path of $\text{tail}_i(x)$ and that length of P is at most $(i+1)\rho$, proving (2). Since the tails are disjoint by assumption, it must be that $v \in V_j$, proving (1).

For the second base $\ell = i + 1$ observe that $v \in G_{i+1}$ and T_i is a BFS tree so the path P on T_i from x to v is contained inside G_{i+1}.

Assume $P \subseteq G_\ell$ for $i < \ell < j$ we prove $P \subseteq G_{\ell+1}$. Since $x \in H_{\ell+1} \subseteq C(T_\ell, j_\ell)$ and $G_{\ell+1}$ is the connected component that contains x inside the subgraph induced by the nodes $G_\ell \cap \bigcup_{m \in \{j_\ell - \ell, \ldots, j_\ell, \ldots, j_\ell + \ell\}} C(T_\ell, m)$ then $d_{G_\ell}(x, G_\ell \setminus G_{\ell+1}) > \ell\rho \geq (i+1)\rho$. Hence a path in G_ℓ from x to v of length $\leq (i+1)\rho$ does not leave $G_{\ell+1}$. This completes the inductive step.

Hence for $\ell = j$ we have $P \subseteq G_j$ and $|P| \leq (i+1)\rho$ this is a contradiction to $d_{G_j}(x, V_j) > j\rho$ since $i < j$. □

Lemma 2. *If G has no $K_{r,r}$ minor and $j_i > r + 1$ for all $i \in R$ then every two nodes of H are r-tail-connected.*

Proof. Fix $u, v \in H$. If $d_H(u, v) \leq r$ then u, v are r-tail-connected since every node is trivially tail-connected to itself. Let $u = x_1, x_2, \ldots, x_{t-1}, x_t = v$ be a shortest path from u to v on H for some $t > r$.

We recursively define a set of nodes $y_1, \ldots, y_r$ as follows. Let $y_1 = u$. Given $y_1, \ldots, y_i$ let y_{i+1} be the node x_ℓ with highest index ℓ such that y_i and $x_{\ell-1}$ are tail-connected. Hence for any index $\ell \leq m \leq t$ node y_i and x_m are not tail connected. Observe that this process can create at most r nodes $y_1, \ldots, y_r$ until v is reached. Suppose v is not reached at stage $r - 1$ and consider the set $y_1, \ldots, y_{r-1}, v$ then from Lemma 1 there are two nodes that are tail-connected in this set. But by the construction of the sequence only y_{r-1} and v can be tail-connected. □

Proof (of Theorem 3). Creating the cover is done by running the partition algorithm in the following manner. Note that the output of the partition algorithm (a cluster H and implicitly for each $i \in R$ graphs G_i,H_i, trees T_i) depends only on the choice of the roots $\tau_1, \ldots, \tau_t$ of the trees, the indices $j_1, \ldots, j_t$, and the choice of connected components $H_1, \ldots, H_t$. We fix a consistent choice of roots.

Each time a root τ_i is to be chosen from subgraph H_i, we choose it as the node with minimal lexicographic order among the nodes in H_i.

The sets $\mathcal{H}_\rho$ and $\mathcal{T}_\rho$ consist of all the clusters H and trees T_i for all possible choices of connected components and all possible choices of indices $j_1, \ldots, j_r$ such that for each $i \in R$ either $j_i \in \mathbb{N}$ or $j_i + 1/2 \in \mathbb{N}$.

Property 1. *(Cover)*. It follows from the simple observation that for any $v \in G$ and tree BFS tree T spanning G with root τ there must exist some integer $j \in \mathbb{N}$ such that either $B_G(v, \rho/4) \subset C(T, j)$ or $B_G(v, \rho/4) \subset C(T, j + 1/2)$. Then, given any v, we construct by induction a sequence of indices $j_1, \ldots, j_i$ such that $B_G(v, \rho/4) \subseteq H_{i+1}$, and thus, for $i = t$, returning a cluster $H \in \mathcal{H}_\rho$ containing $B_G(v, \rho/4)$.

Property 2. *(Sparse clusters)*. Due to the fact that for every $i \in R$, $j_i \in \mathbb{N}$ or $j_i + 1/2 \in \mathbb{N}$, then a node belongs of at most 2^{i-1} graphs H_i. This is true since for each graph H_i that v belongs to, it belongs to at most two graphs of type H_{i+1}. Hence the number of clusters a node belongs to is at most 2^r.

Property 3. *(Sparse trees)*. For each graph G_i that v belongs to, it belongs to at most $2i+1$ graphs G_{i+1} due to the use of $2i+1$ stripes in the definition of G_{i+1}. Hence for each $i \in R$, by simple induction, a node belongs to $\prod_{1 \le j \le i} 2j + 1 \le (2i + 1)!!$ graphs G_i. Therefore a node belongs to at most $\sum_{i \in R}(2i + 1)!! = 2^{O(r \log r)}$ trees by summing over all G_{i+1} for all $i \in R$.

Property 4. *(Weak diameter)*. When $j_i > r + 1$ for each $i \in R$ this follows directly from Lemma 2. If there is some $i \in R$ for which $j_i \le r + 1$ then any two nodes in H are tail-connected via tree T_i. $\square$

4 Name-Independent Routing Scheme for Weak Diameter Cover

Theorem 1. *For every n-node unweighted graph excluding a fixed $K_{r,r}$ minor, there is a polynomial time constructible name-independent routing scheme with constant stretch factor, in which every node v requires routing tables of $\widetilde{O}(1)$ bits and $O(\log^2 n / \log \log n)$-bit headers.*

The key ingredient of our routing scheme is the following lemma:

Lemma 3. *Let $\mathcal{H}, \mathcal{T}$ be the set of clusters and trees obtained from the cover algorithm with parameter ρ on a graph excluding a fixed $K_{r,r}$ minor. There exists an error-reporting name-independent routing scheme such that*

1. *Each node v stores $\widetilde{O}(1)$ bits per tree of $\mathcal{T}$ it belongs to.*
2. *Each node v stores $\widetilde{O}(1)$ bits per cluster of $\mathcal{H}$ it belongs to.*
3. *For any $H \in \mathcal{H}$ and $s, t \in H$, searching for t from s in H will find t at cost $O(\rho)$.*
4. *For any $H \in \mathcal{H}$ and $s \in H, t \notin H$, searching for t from s in H will cost $O(\rho)$ until an error report is sent back to s.*

Lemma 3 is proven by constructing for every $H \in \mathcal{H}$ a rooted tree T_H. Consider first the case that H has a strong diameter of $O(\rho)$. Hence T_H can simply be a spanning BFS tree of H. In this case we can use the following result for single-source tree routing on unweighted trees on graphs excluding a fixed minor.

Lemma 4. *Let F be a forest (i.e., a set of disjoint trees) of an n-node graph G excluding a fixed minor. Then there exists polynomial time constructible scheme with $\widetilde{O}(1)$ bit routing tables for each $v \in G$ such that for each tree $T \in F$ there is a name-independent single-source error-reporting routing scheme on T (in the fixed port model) with cost $O(\mathrm{diam}(T))$.*

The construction and proof of Lemma 4 is an extension of the scheme in [1,2] with results from graph minor theory [13] that imply that graphs excluding a fixed minor have constant arboricity. The *arboricity* [29] of a graph G is the minimum number of forests into which the edges of G can be partitioned. In the full version we show in the proof of Lemma 4 how to utilize this fact, for routing on any tree $T \in F$ which is a subgraph of G, in order to bound the routing table size when converting from the designer ports used in [1] to the standard fixed port model.

However, H may not have a small strong diameter so routing outside the cluster must occur. At a high level, the idea of the construction is as follows.

According to the partition algorithm, the route between every pair of nodes x, y in H is a route containing at most r pairs of intersecting tails, with at most a single edge from one pair to the next.

In the following we sketch the main ideas for the case of $r = 3$. The high level idea is to build a tree T_H that will span H but will also go outside H. The root of T_h is an arbitrary node $r_1 \in H$. The diameter of T_H will be $O(\rho)$. Each branch may leave H once or twice along tail-connected paths (for the case of $r = 3$).

More specifically, when stepping outside H, we only do it along tail-connected paths, which are composed of two parts: Going upward in some tree T_k, and then downward on some tree T_ℓ. Consider in T_H the case where we have a branch containing $u \rightsquigarrow w \rightsquigarrow v$, such that u and v belong to H, the left $\rightsquigarrow$ stands for a path on tree T_k and the right $\rightsquigarrow$ stands for a path on T_ℓ, and w is a node in the intersection between the two tail paths, $w \in T_k \cap T_\ell$. We collapse the each of the paths $\rightsquigarrow$ into a single edge, entering and leaving a *virtual node* which we denote by $\langle u, k, \ell, z\rangle$ where $z = d_{T_k}(u, w)$.

Now suppose we have built the tree. There remains the problem of building a name-independent scheme for it. A useful result here would be the single-source name-independent scheme we have presented in [1] (Lemma 4). This would give us stretch $O(1)$ with $\widetilde{O}(1)$ memory per node. However, it cannot be employed easily, since some nodes in T_H are outside the cluster H, and may not maintain specific information about T_H (or else, they might need to maintain information about too many trees). Additionally, nodes in T_H may have neighbors outside T_H, which prevents us from using the result in [1] directly.

The solution we develop emulates the virtual tree of T_H. We need to address three issues.

1. Each (regular) node v must distinguish its neighbors in T_H from any other neighbors it may have in G. The solution is based on having bounded arboricity. We use a partition of the edges into a constant number of forests. This allows nodes to store only $\widetilde{O}(1)$ bits of information about all their children in the tree.
2. Each node u must recognize its virtual children, and be able to route to them. To solve this, first note that for any u, there are $O(\rho)$ possible combinations of trees T_k, T_ℓ and nodes w along upward tails within distance $O(\rho)$. Therefore, u utilizes $O(\rho)$ virtual children. We allocate for u a set of ρ nodes, denoted $M(u)$, each of which emulates one virtual child. So whenever an algorithm on T_H calls to use a path from u to w, we first search a node within $M(x)$ that emulates w. Then obtain the necessary information from it and route back to u. This side-track information lookup incurs a cost of $O(\rho)$ hops. Then we use this information to simply route on T_k toward w.
3. The emulation of a virtual node w must be able to reach any child v of w in T_H. The route is along the tree T_ℓ. Our solution is to define for T_H a subtree of T_ℓ that contains H, as well as all nodes in T_ℓ within distance $O(\rho)$ from H. To be more precise, we define a forest $F(\ell, j)$ which is the subgraph of T_ℓ that spans a constant number of consecutive levels $C(T_\ell, \cdot)$. Let $j = j_\ell$ be the index that generated H_ℓ. The diameter each tree in $F(\ell, j)$ is $O(\rho)$ hence we can route on the tree T in $F(\ell, j)$ that contains both w and v. Using the single-source name-independent scheme of [1] on $T \in F(\ell, j)$, node w can route to v via the root of the tree $T \in F(\ell, j)$ at a cost of $O(\rho)$.

We now go back to describe the construction of the tree T_H. We start by choosing a root r_1 arbitrarily. The first component of the tree T_H is simply a BFS tree of a set C_1 containing all nodes reachable within H via length 8ρ paths from r_1.

In the next step, we bring in nodes $r_2, r_3, ...$ that have distance greater than 8ρ from T_H, and are tail-connected with r_1. We do not include in the tree all the nodes along the tails from r_1 to r_i. Rather, we define a virtual node $\langle r_1, k, \ell, z\rangle$, that records the name T_k of the tree of the upward tail, the name T_ℓ of the tree of the downward tail, and the distance z upward on T_k of the intersection with T_ℓ. The virtual node is added to the tree, as well as the edges $(r_i, \langle r_1, k, \ell, z\rangle)$, $(\langle r_1, k, \ell, z\rangle, r_i)$. We also add all nodes with paths of length 4ρ in H from r_i. We then set $T_H = \bigcup_i C_i$. In order to store node r_1's virtual children information, we designate as $M(r_1)$ the ρ closest nodes to r_2.

In the next step, we bring in one at a time nodes r_i that are at distance at least 8ρ from T_H, and that are tail connected to some node x in T_H. For each such node x, we repeat the process for r_1 above.

Finally, we insert into T_H all remaining nodes, which must be at distance at most $O(\rho)$ within H.

Due to space constrains the construction for general r and the proof of Lemma 4, Lemma 3 and Theorem 1 will appear in the full version.

5 Lower Bounds

Theorem 2. *Every shortest-path labeled routing scheme on polynomially weighted n-node planar graphs of bounded degree requires a total label length (the length of all the routing tables and node labels in the graph) of $\Omega(n^{1+1/2})$ bits. Moreover, the (maximum) label length is $\Omega(n^{1/4})$ for some weighted bounded degree triangulations. For unweighted planar graphs, the two bounds are respectively $\Omega(n^{1+1/3})$ and $\Omega(n^{1/6})$.*

Let us consider the problem of labeled routing in planar graphs along shortest-path. To strengthen our lower bound we assume that the routing table and the node label of any target t are merged into a single routing label, denoted by $L(G,t)$ for a graph G. So in this model, the routing decision in the source s is taken with full knowledge of $L(G,s)$ and $L(G,t)$. Observe that in this model, we make no assumptions on headers (rewritable and of arbitrary length). We also assume the designer-port model, i.e., the designer can permute the port numbers in order to optimize the maximum label length, however the ports of v must range in $\{1,\ldots,\deg(v)\}$.

Our proof is based on the planar graph construction of [23]. This graph, denoted G_k for some parameter k, was used to prove lower bounds for *distance labeling schemes* in planar graphs. For this labeling problem each node of a graph receives a label such that distances between any two nodes can be computed from their labels only. There is no general relation between both labeling problems, routing and distance, and thus a lower bound on distance labeling cannot be applied to labeled routing as a black box. Nethertheless, in the full version we show how adapt the G_k construction to exhib a costly routing decision.

References

1. I. Abraham, C. Gavoille, and D. Malkhi, *Routing with improved communication-space trade-off*, in 18^{th} International Symposium on Distributed Computing (DISC), vol. 3274 of Lecture Notes in Computer Science, Springer, Oct. 2004, pp. 305–319.
2. ———, *Routing with improved communication-space trade-off*, Tech. Report RR-1330-04, LaBRI, University of Bordeaux 1, 351, cours de la Libération, 33405 T alence Cedex, France, July 2004.
3. I. Abraham, C. Gavoille, D. Malkhi, N. Nisan, and M. Thorup, *Compact name-independent routing with minimum stretch*, in 16^{th} Annual ACM Symposium on Parallel Algorithms and Architecture (SPAA), ACM PRESS, June 2004.

4. I. Abraham and D. Malkhi, *Compact routing on euclidian metrics*, in Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing (PODC), ACM Press, 2004, pp. 141–149.
5. I. Abraham and D. Malkhi, *Name independent routing for growth bounded networks*, in 17^{th} Annual ACM Symposium on Parallel Algorithms and Architecture (SPAA), ACM Press, July 2005. To appear.
6. B. Awerbuch and D. Peleg, *Sparse partitions*, in 31^{th} Annual IEEE Symposium on Foundations of Computer Science (FOCS), IEEE Computer Society Press, Oct. 1990, pp. 503–513.
7. B. Awerbuch and D. Peleg, *Routing with polynomial communication-space trade-off*, SIAM J. Discret. Math., 5 (1992), pp. 151–162.
8. P. Bose and P. Morin, *Competitive online routing in geometric graphs*, Theoretical Computer Science, 324 (2004), pp. 273–288.
9. ———, *Online routing in triangulations*, SIAM Journal on Computing, 33 (2004), pp. 937–951.
10. H. T.-H. Chan, A. Gupta, B. M. Maggs, and S. Zhou, *On hierarchical routing in doubling metrics*, in 16^{th} Symposium on Discrete Algorithms (SODA), ACM-SIAM, Jan. 2005.
11. V. D. Chepoi, F. F. Dragan, and Y. Vaxes, *Distance and routing labeling schemes for non-positively curved plane graphs*, Journal of Algorithms, (2004). To appear.
12. V. D. Chepoi and A. Rollin, *Interval routing in some planar quadrangulations*, in 8^{th} International Colloquium on Structural Information & Communication Complexity (SIROCCO), Carleton Scientific, June 2001, pp. 89–104.
13. M. DeVos, G. Ding, B. Oporowski, D. P. Sanders, B. Reed, P. D. Seymour, and D. Vertigan, *Excluding any graph as a minor allows a low tree-width 2-coloring*, Journal of Combinatorial Theory, Series B, 91 (2004), pp. 25–41.
14. Y. Dourisboure, *Compact routing schemes for bounded tree-length graphs and for k-chordal graphs*, in 18^{th} International Symposium on Distributed Computing (DISC), vol. 3274 of Lecture Notes in Computer Science, Springer, Oct. 2004, pp. 365–378.
15. Y. Dourisboure and C. Gavoille, *Improved compact routing scheme for chordal graphs*, in 16^{th} International Symposium on Distributed Computing (DISC), vol. 2508 of Lecture Notes in Computer Science, Springer, Oct. 2002, pp. 252–264.
16. F. F. Dragan and I. Lomonosov, *On compact and efficient routing in certain graph classes*, in 15^{th} Annual International International Symposium on Algorithms and Computation (ISAAC), vol. 3341 of Lecture Notes in Computer Science, Springer, Dec. 2004, pp. 402–414.
17. T. Eilam, C. Gavoille, and D. Peleg, *Compact routing schemes with low stretch factor*, Journal of Algorithms, 46 (2003), pp. 97–114.
18. P. Flocchini and F. L. Luccio, *Routing in series parallel networks.*, Theory Comput. Syst., 36 (2003), pp. 137–157.
19. G. N. Frederickson and R. Janardan, *Efficient message routing in planar networks*, SIAM Journal on Computing, 18 (1989), pp. 843–857.
20. C. Gavoille, *Routing in distributed networks: Overview and open problems*, ACM SIGACT News - Distributed Computing Column, 32 (2001), pp. 36–52.
21. C. Gavoille and N. Hanusse, *Compact routing tables for graphs of bounded genus*, in 26^{th} International Colloquium on Automata, Languages and Programming (ICALP), vol. 1644 of LNCS, Springer, July 1999, pp. 351–360.

22. C. Gavoille and D. Peleg, *Compact and localized distributed data structures*, Journal of Distributed Computing, 16 (2003), pp. 111–120. PODC 20-Year Special Issue.
23. C. Gavoille, D. Peleg, S. Pérennès, and R. Raz, *Distance labeling in graphs*, Journal of Algorithms, 53 (2004), pp. 85–112.
24. A. Gupta, R. Krauthgamer, and J. R. Lee, *Bounded geometries, fractals, and low-distortion embeddings*, in 44^{th} Annual IEEE Symposium on Foundations of Computer Science (FOCS), IEEE Computer Society Press, Oct. 2003, pp. 534–543.
25. Y. Hassin and D. Peleg, *Sparse communication networks and efficient routing in the plane*, Distributed Computing, 14 (2001), pp. 205–215.
26. P. Klein, S. A. Plotkin, and S. Rao, *Excluded minors, network decomposition, and multicommodity flow*, in 25^{th} Annual ACM Symposium on Theory of Computing (STOC), ACM Press, 1993, pp. 682–690.
27. K. A. Laing, *Brief announcement: name-independent compact routing in trees*, in 24^{th} Annual ACM Symposium on Principles of Distributed Computing (PODC), ACM Press, 2004, pp. 382–382.
28. H.-I. Lu, *Improved compact routing tables for planar networks via orderly spanning trees*, in 8^{th} Annual International Computing & Combinatorics Conference (COCOON), vol. 2387 of LNCS, Springer, Aug. 2002, pp. 57–66.
29. C. S. J. Nash-Williams, *Edge-disjoint spanning trees of finite graphs*, J. London Math. Soc, 36 (1961), pp. 445–450.
30. D. Peleg, *Distributed Computing: A Locality-Sensitive Approach*, SIAM Monographs on Discrete Mathematics and Applications, 2000.
31. ———, *Proximity-preserving labeling schemes*, Journal of Graph Theory, 33 (2000), pp. 167–176.
32. D. Peleg and E. Upfal, *A trade-off between space and efficiency for routing tables*, Journal of the ACM, 36 (1989), pp. 510–530.
33. A. Slivkins, *Distance estimation and object location via rings of neighbors*, in 24^{th} Annual ACM Symposium on Principles of Distributed Computing (PODC), ACM Press, July 2005. To appear. Also appears as Cornell CIS technical report TR2005-1977.
34. K. Talwar, *Bypassing the embedding: Algorithms for low dimensional metrics*, in 36^{th} Annual ACM Symposium on Theory of Computing (STOC), June 2004, pp. 281–290.
35. M. Thorup, *Compact oracles for reachability and approximate distances in planar digraphs*, in 42^{nd} Annual IEEE Symposium on Foundations of Computer Science (FOCS), IEEE Computer Society Press, Oct. 2001, pp. 242–251.
36. M. Thorup and U. Zwick, *Compact routing schemes*, in 13^{th} Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA), ACM Press, July 2001, pp. 1–10.
37. D. B. West, *Introduction to Graph Theory (second edition)*, Prentice Hall, 2001.