

# Geographic Routing Using Hyperbolic Space

Robert Kleinberg

Computer Science Division

UC Berkeley

Berkeley, CA 94720

Email: rdk@eecs.berkeley.edu

**Abstract**—We propose a scalable and reliable point-to-point routing algorithm for ad hoc wireless networks and sensor networks. Our algorithm assigns to each node of the network a virtual coordinate in the hyperbolic plane, and performs greedy geographic routing with respect to these virtual coordinates. Unlike other proposed greedy routing algorithms based on virtual coordinates, our embedding guarantees that the greedy algorithm is always successful in finding a route to the destination, if such a route exists. We describe a distributed algorithm for computing each node’s virtual coordinates in the hyperbolic plane, and for greedily routing packets to a destination point in the hyperbolic plane. (This destination may be the address of another node of the network, or it may be an address associated to a piece of content in a Distributed Hash Table. In the latter case we prove that the greedy routing strategy makes a consistent choice of the node responsible for the address, irrespective of the source address of the request.) We evaluate the resulting algorithm in terms of both path stretch and node congestion.

## I. INTRODUCTION

This paper proposes a scalable and reliable approach for point-to-point routing in ad hoc wireless networks and sensor networks. Unlike Internet routing — which is based on forwarding tables populated by blocks of IP addresses — sensor network routing must deal with the absence of a standardized, precomputed address space, while also obeying the resource limitations of the individual nodes. This typically excludes implementations which maintain a large amount of state (e.g. forwarding tables) at each node. To overcome this limitation, *geographic routing* techniques [1], [2] assign coordinates to the nodes of the network, e.g. based on their locations in the underlying physical space in which the sensors are deployed. Nodes then forward messages according to the coordinates of the destination and the coordinates of their neighbors.

The simplest geographic routing techniques are *greedy*, in the sense that nodes always forward messages to the neighbor which is closest to the destination. Greedy routing is susceptible to getting stuck in an infinite loop when there exists a node which is nearer to the destination than all of its neighbors. The earliest approaches for overcoming this obstacle [3], [1] were

based on *face routing*, an idea discussed further in Section I-A below. Recently, more robust solutions have been proposed based on the idea of assigning to each node a *virtual coordinate* in some metric space  $X$ , and routing messages greedily in this space  $X$  [4], [5].

While the efficacy of greedy routing based on virtual coordinates has been experimentally confirmed [6], [4], [5], there is a paucity of theorems identifying circumstances under which this strategy is guaranteed to work. In fact, for each of the aforementioned approaches it is known that greedy routing may fail to find a route for at least a small percentage of source-destination pairs. Abstractly, then, we are interested in the following question: given a graph  $G$  and a metric space  $X$ , when can we compute an embedding of  $G$  into  $X$  such that greedy routing in  $X$  succeeds in defining a path between every pair of nodes? To make the question more precise, we may introduce the following definition.

**Definition I.1 ([7]).** A *greedy embedding* of an undirected graph  $G$  in a metric space  $(X, d)$  is a mapping  $f : V(G) \rightarrow X$  with the following property: for every pair of distinct vertices  $s, t \in V(G)$  there exists a vertex  $u$  adjacent to  $s$  such that  $d(f(u), f(t)) < d(f(s), f(t))$ .

This notion of greedy embedding was defined by Papadimitriou and Ratajczak [7], who studied the special case when  $X$  is the plane with the Euclidean (i.e.  $\ell_2$ ) metric. They provided examples of graphs which do not admit a greedy embedding in the Euclidean plane. In fact for every finite-dimensional normed vector space  $V$  there exist graphs which have no greedy embedding in  $V$ . (This follows, e.g., from Theorem 1 below.) However, Papadimitriou and Ratajczak conjectured that every 3-connected planar graph admits a greedy embedding in the Euclidean plane. These initial results and conjectures motivate the following questions.

- 1) Is there a normed vector space  $V$  which admits a greedy embedding of every graph with  $n$  nodes? If so, how small can we make the dimension of  $V$ ?
- 2) Is there a finite-dimensional Riemannian manifold

$M$  which admits a greedy embedding of every finite graph? If so, how small can we make the dimension of  $M$ ?

It is trivial to settle Question 1 affirmatively with  $\dim(V) = n$ ; here we improve this bound to  $\dim(V) = O(\log n)$  and prove that this is the best possible dependence on  $n$ .

**Theorem 1.** *If  $V$  is a  $d$ -dimensional normed vector space which admits a greedy embedding of every graph with  $n$  nodes, then  $d = \Omega(\log n)$ . If  $d = \lceil \log_{3/2}(n) \rceil$  then the space  $\ell_\infty^d$  admits a greedy embedding of every graph with  $n$  nodes.*

The proof of Theorem 1 follows easily from standard metric embedding techniques [8]. However, in light of the first half of the theorem it is perhaps surprising that Question 2 *also* has an affirmative answer: there exists a finite-dimensional manifold which admits a greedy embedding of every finite graph.

**Theorem 2.** *Every connected finite graph has a greedy embedding in the hyperbolic plane.*

As will become apparent from the proof of Theorem 2, our embedding technique is well suited to distributed implementations, since the task of computing virtual coordinates requires only the following sub-tasks:

- 1) Computing a spanning tree of the network.
- 2) Determining this tree's maximum degree,  $d$ .
- 3) Naming each node of the network with a node of the infinite  $d$ -regular tree, by identifying the spanning tree with a subtree of the infinite  $d$ -regular tree,

One of the chief reasons for requiring a point-to-point routing service in a sensor network is to enable data-centric storage [9], in which a Distributed Hash Table (DHT) is implemented in the sensor network, and data is stored at the nodes of the network using the DHT. Our embedding and routing technique is well-suited to data-centric storage, using a generalization of the Content Addressable Network (CAN) paradigm [10]. More precisely, the `put` and `get` operations of the DHT can be implemented by hashing the key to a point  $p$  in the hyperbolic plane, greedily routing to a node  $x$  whose virtual coordinate is nearer to  $p$  than any of its neighbors, and either storing or retrieving the value from this node. In order for this to constitute a reliable DHT implementation, one must ensure that the choice of the node  $x$  does not depend on the source of the `put` or `get` operation. We prove this in Theorem 3.

## A. Related Work

1) *Geometric routing in sensor networks:* The earliest papers on geometric routing in sensor networks [3], [1] assumed that the nodes of the network know their own positions in the underlying physical space which the network inhabits, and they proposed routing strategies based on greedy routing with respect to these coordinates. As we have already observed, greedy routing does not guarantee point-to-point connectivity in the network because messages can become trapped in local minima of the distance-to-destination function. (Such local minima are sometimes referred to as “lakes” or “voids”, since they tend to arise when there is a large empty space in the region occupied by the nodes.) To deal with this inadequacy of greedy routing, the algorithms in [3], [1] and many subsequent papers use *face routing* to escape from such local minima by routing around the perimeter of a face in a planar subgraph of the network until greedy routing can be resumed. This is guaranteed to succeed in defining a route between every pair of nodes, provided that nodes know their own locations and that there is a distributed algorithm to compute a connected planar subgraph of the network. In [1] this “planarization” step is accomplished by assuming that the underlying network is a unit disk graph and using the *relative neighborhood graph* or the *Gabriel graph*. An alternative planarization of unit disk graphs with better stretch properties is introduced in [11]. For more sophisticated cost measures, [2] presents a combination of greedy routing and face routing which is asymptotically optimal for bounded-degree unit disk graphs. Surprisingly, the first distributed planarization algorithm for *general* graphs was discovered only very recently [12].

All of the foregoing work assumed that nodes were aware of their physical locations, an assumption which is often violated in practice for various reasons. (For instance, the most widely used node, the Berkeley mote [13], has no GPS antenna.) In addition, implementations of face routing are either based on non-rigorous heuristics or on complicated planarization procedures, in many cases tailored to restrictive assumptions about the nature of the underlying graph (e.g. that it is a unit-disk graph). To overcome these shortcomings, two recent papers [4], [5] proposing routing algorithms which assign virtual coordinates in a metric space  $X$  and forward messages using greedy routing in  $X$ . In [5] the metric space is the Euclidean plane, and virtual coordinates are assigned using a distributed version of Tutte’s “rubber band” algorithm [14] for finding convex embeddings of graphs. In [4] the graph is embedded in

$\mathbb{R}^d$  for some value of  $d$  much smaller than the network size, by identifying  $d$  beacon nodes and representing each node by the vector of distances to those beacons. The distance function on  $\mathbb{R}^d$  used in [4] is actually not a metric; it is a modification of the  $\ell_1$  norm, fine-tuned for the objective of sensor network routing. Both [4] and [5] provide substantial experimental support for the efficacy of their proposed embedding techniques — both algorithms are successful in finding a route from the source to the destination more than 95% of the time — but neither of them has a provable guarantee. Our Theorem 2 complements the results of [5] by proving that one may achieve 100% success rate with greedy routing by assigning virtual coordinates in the hyperbolic plane rather than the Euclidean plane. Our Theorem 1 complements the results of [4] by proving that one may also achieve 100% success rate by embedding in  $\mathbb{R}^d$  where  $d = O(\log n)$  and using the  $\ell_\infty$  norm rather than the  $\ell_1$  norm, though our  $d$ -dimensional embedding (unlike theirs) is not beacon-based and fails to meet the simplicity requirement which is a cornerstone of work in [4].

2) *Greedy embeddings*: Greedy embeddings of graphs were defined by Papadimitriou and Ratajczak in [7], a paper primarily concerned with the question of whether every 3-connected planar graph has a greedy embedding in the Euclidean plane. While this conjecture remains unsolved, their paper contains many interesting related theorems and examples. Most relevant to the present work is their theorem that every 3-connected planar graph has an embedding in  $\mathbb{R}^3$  which satisfies the definition of a greedy embedding when “distance” is defined by the function  $d(\vec{x}, \vec{y}) = -\vec{x} \cdot \vec{y}$ . This is not a greedy embedding because the distance function is not a metric on  $\mathbb{R}^3$ . However, this type of embedding is just as useful as a greedy embedding from the standpoint of geometric routing in sensor networks, since it is based on a distance function which is easily computable from the virtual coordinates of the two points in question. Our Theorem 2 improves on the three-dimensional embedding from [7] in two respects:

- The embedding can be defined for any connected graph  $G$ , not only for graphs which contain a planar 3-connected subgraph.
- The task of computing the embedding easily lends itself to distributed computation, since it only requires computing a spanning tree of the graph and identifying this tree with a subtree of a complete  $d$ -regular tree for some  $d$ . In contrast, the three-dimensional embedding from [7] requires first computing a 3-connected planar subgraph of the

graph, then computing a Koebe-Andre’ev-Steinitz representation of this subgraph, i.e. identifying the subgraph with the edge graph of a 3-dimensional convex polytope whose edges are all tangent to a unit sphere. It is hard to see how these steps could be performed in a decentralized manner.

3) *Metric embeddings*: The study of *low-distortion embeddings* of metric spaces has had a profound impact on algorithmic theory in the last decade. The notion of *greedy embeddings* studied here is related to order-preservation properties of metric embeddings rather than their distortion properties. While these *ordinal embedding* questions have received comparably less attention, a number of prior papers have studied them for their use in nearest-neighbor search, data visualization, compression, and clustering [15], [16]. Bilu and Linial [17] studied *monotone maps* of metric spaces — i.e. those which preserve the relative ordering of distances — and proved that there exist graphs with  $n$  vertices having no monotone embedding into a vector space of dimension  $o(n)$ . In this paper, we have seen that this negative result does not prevent us from basing wireless routing algorithms on low-dimensional ordinal embeddings.

## B. Outline

The rest of this paper is organized as follows. Section II is concerned with greedy routing in hyperbolic space, including the proof of Theorem 2. In Section III we turn our attention to normed vector spaces and prove Theorem 1. Section IV presents the results of an experimental evaluation of our algorithm, to determine its performance in terms of stretch and congestion. We finish with some open questions.

## II. GREEDILY EMBEDDING GRAPHS IN HYPERBOLIC SPACE

In this section we establish Theorem 2, which asserts that every connected finite graph has a greedy embedding in the hyperbolic plane. We start by making two observations which are trivial consequences of the definition of greedy embedding.

**Observation II.1.** If  $H \subseteq G$  is a subgraph containing all vertices of  $G$ , then every greedy embedding of  $H$  is also a greedy embedding of  $G$ .

**Observation II.2.** If  $T$  is a tree and  $T' \subseteq T$  is a subtree, then every greedy embedding of  $T$  in a metric space  $X$  restricts to a greedy embedding of  $T'$ .

*Proof.* Let  $f : T \rightarrow X$  be a greedy embedding. For any two nodes  $s, t \in T'$ , we know that  $s$  and  $t$  may be joined in  $T$  by a path  $s = s_0, s_1, \dots, s_k = t$  such that

the function  $d(f(s_i, t))$  is monotonically decreasing as  $i$  goes from 0 to  $k$ . This path must lie in  $T'$ , since  $T'$  is a subtree of  $T$  and the endpoints of the path belong to  $T'$ . Thus  $s$  has a neighbor  $s_1$  in  $T'$  which satisfies  $d(s_1, t) < d(s, t)$ .  $\square$

Our strategy for proving Theorem 2 is to exhibit, for each  $d \geq 3$ , a greedy embedding of the infinite  $d$ -regular tree into the hyperbolic plane. Since every finite tree is a subtree of the infinite  $d$ -regular tree for sufficiently large  $d$ , this implies (using Observation II.2) that every finite tree has a greedy embedding into the hyperbolic plane. Now, given an arbitrary connected finite graph  $G$ , we can take a spanning tree  $T \subseteq G$ , find a greedy embedding of  $T$  into the hyperbolic plane, and use Observation II.1 to conclude that this is also a greedy embedding of  $G$ . Thus we have reduced Theorem 2 to the problem of constructing a greedy embedding of the infinite  $d$ -regular tree into the hyperbolic plane, for each  $d \geq 3$ .

The next subsection will review some basic facts about the geometry of the hyperbolic plane, which are necessary for the construction of these greedy embeddings. The following subsection will construct a greedy embedding for the infinite 3-regular tree. Finally, in the last subsection, we will cover the case  $d > 3$ .

#### A. The hyperbolic plane and its isometries

In this section we recall some facts about the geometry of the hyperbolic plane  $H$ . For an excellent exposition of these and other properties of  $H$ , see the second chapter of [18].

Two standard models of  $H$  are the upper half-plane model and the Poincaré disk model. In the upper half-plane model  $H$  is represented by the set of points  $(x, y) \in \mathbb{R}^2$  satisfying  $y > 0$ , endowed with the Riemannian metric

$$ds^2 = \frac{dx^2 + dy^2}{y^2}.$$

In the Poincaré disk model,  $H$  is represented by the set of points satisfying  $x^2 + y^2 < 1$ , endowed with the Riemannian metric

$$ds^2 = \frac{4(dx^2 + dy^2)}{(1 - x^2 - y^2)^2}.$$

In both cases, we will refer to points of the hyperbolic plane using complex coordinates, by treating a point  $(x, y) \in \mathbb{R}^2$  as a complex number  $z = x + yi$ . The hyperbolic plane has a boundary circle  $\partial H$  “at infinity” which may be identified with the circle  $|z| = 1$  in the Poincaré model; in the upper half plane model the boundary circle  $\partial H$  is identified with the one-point compactification of the real line. In both cases, the points

of the boundary circle are referred to as “points at infinity” or “ideal points”.

Every geodesic of the hyperbolic plane is represented by an arc of a circle which meets  $\partial H$  in two ideal points  $p, q$ . (A straight line is considered to be a degenerate case of a circle. In the upper half plane model, we treat a vertical line as a circle which intersects  $\partial H$  in two points: the point at infinity and one point of the real line.) For distinct points  $p, q, r \in \partial H$ , the hyperbolic geodesics joining  $p$  to  $q$ ,  $q$  to  $r$ , and  $r$  to  $p$  enclose a region of  $H$  bounded by a curvilinear triangle; we call this region the *ideal triangle* with vertices  $p, q, r$ .

Every orientation-preserving isometry of  $H$  is represented by a Möbius transformation

$$z \mapsto \frac{az + b}{cz + d}$$

for some complex coefficients  $a, b, c, d$  such that  $ad \neq bc$ . Representing this transformation by a complex matrix

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix},$$

it turns out that the composition of two Möbius transformations is represented by the product of the corresponding matrices. This defines a mapping from the group  $GL_2(\mathbb{C})$  of invertible 2-by-2 complex matrices to the group of Möbius transformations of  $\mathbb{C}$ . The kernel of this mapping consists of all scalar multiples of the identity, so the group of Möbius transformations is isomorphic to the quotient of  $GL_2(\mathbb{C})$  by the subgroup of scalar multiples of the identity. This quotient is denoted by  $PGL_2(\mathbb{C})$ .

A Möbius transformation is an isometry of the upper half-plane model of  $H$  if and only if it preserves the upper half-plane as a point set. It is easy to check that every 2-by-2 real matrix with determinant 1 represents a Möbius transformation preserving the upper half-plane, and in fact the converse is true: every Möbius transformation preserving the upper half-plane is represented by an element of  $GL_2(\mathbb{R})$  with determinant 1. Thus the isometry group of  $H$  is identified with the group  $PSL_2(\mathbb{R})$  of two-by-two real matrices with determinant 1, modulo  $\{\pm I\}$ . This group acts transitively on the set of clockwise-ordered triples of points of  $\partial H$ .

The mapping  $z \mapsto \frac{1+iz}{1-iz}$  defines an isometry from the upper half-plane model of  $H$  to the Poincaré disk model of  $H$ . Denoting this Möbius transformation by  $M$ , the inner automorphism  $i_M$  of the Möbius group  $PGL_2(\mathbb{C})$  maps  $PSL_2(\mathbb{R})$  to another subgroup which is the isometry group of the Poincaré disk. We will identify isometries of the Poincaré disk with elements of  $PSL_2(\mathbb{R})$  using this inner automorphism of  $PGL_2(\mathbb{C})$ .

The subgroup  $PSL_2(\mathbb{Z}) \subset PSL_2(\mathbb{R})$  consists of all transformations represented by two-by-two *integer* matrices. This subgroup is generated by the matrices

$$a = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}; \quad b = \begin{bmatrix} 0 & 1 \\ -1 & 1 \end{bmatrix}.$$

These matrices satisfy  $a^2 = b^3 = -I$ , so they represent elements of order 2 and 3, respectively, in  $PSL_2(\mathbb{Z})$ . (Recall that  $-I$  is identified with  $I$  in  $PSL_2(\mathbb{Z})$ .) In the Poincaré disk model, these transformations  $a, b$  are represented by the mappings

$$a : z \mapsto -z$$

and

$$b : z \mapsto \frac{(1+2i)z+1}{z+(1-2i)}.$$

If  $\Delta$  is the ideal triangle with vertices  $0, 1, i$ , then  $a$  exchanges  $\Delta$  with its complex conjugate  $\overline{\Delta}$ , and  $b$  preserves  $\Delta$  while cyclically permuting its vertices. As  $g$  ranges over all elements of  $PSL_2(\mathbb{Z})$ , the ideal triangles  $g(\Delta)$  form a tiling of  $H$ . The dual of this tiling is a 3-regular tree  $T$  which may be described as follows. Let  $u = (2 - \sqrt{3})i$  and  $v = -u$ . Let  $e$  be the pair  $(u, v)$ . Then the vertex set of  $T$  is  $\{g(u) : g \in PSL_2(\mathbb{Z})\}$  and the edge set of  $T$  is  $\{g(e) : g \in PSL_2(\mathbb{Z})\}$ . The group  $PSL_2(\mathbb{Z})$  acts both vertex-transitively and edge-transitively on  $T$ . In fact  $PSL_2(\mathbb{Z})$  is isomorphic to the group of automorphisms of  $T$  which preserve the cyclic ordering of the edges around every vertex. The tiling and its dual tree  $T$  are illustrated in Figure 1, which also illustrates the equivalent picture in the upper half-plane model.

### B. A greedy embedding of the infinite 3-regular tree

Recall that Figure 1 illustrates an infinite tiling of  $H$  and an infinite 3-regular tree  $T$  dual to the tiling, both of which are invariant under the action of the discrete isometry group  $PSL_2(\mathbb{Z})$ . We claim that this embedding of the tree  $T$  is a greedy embedding. To verify this, it suffices to show that for every path  $s = s_0, s_1, \dots, s_k = t$  in  $T$ , the inequality  $d(s_0, t) > d(s_1, t)$  is satisfied. Recall that the action of  $PSL_2(\mathbb{Z})$  on  $T$  is edge-transitive, so we may assume without loss of generality that  $s_0 = u$  and  $s_1 = v$ . If we delete edge  $(u, v)$  from  $T$ , this separates  $T$  into two subtrees consisting of the nodes corresponding to points in the upper half-plane and those in the lower half-plane. Since  $v$  lies on the path from  $u$  to  $t$ , we may conclude that  $t$  is in the lower half-plane. To prove our assertion that  $d(u, t) > d(v, t)$ , we will prove the following more general fact: for any  $z$  in the lower half of the unit disk,  $d(u, z) > d(v, z)$ . To prove

it, let  $L$  be a geodesic segment from  $u$  to  $z$  and let  $y$  be the intersection of  $L$  with the real line. We have

$$d(u, z) = d(u, y) + d(y, z).$$

The geodesic segment from  $v$  to  $z$  lies entirely in the lower half-plane, hence it does not contain  $y$ ; consequently

$$d(v, z) < d(v, y) + d(y, z).$$

But  $d(v, y) = d(u, y)$  since complex conjugation is an isometry of  $H$ . Hence  $d(v, z) < d(u, y) + d(y, z) = d(u, z)$ , as claimed.

### C. A greedy embedding of the infinite $d$ -regular tree

To construct a greedy embedding of the infinite  $d$ -regular tree for any  $d > 3$ , we use the same ideas but the symmetry group  $\Gamma$  which plays the role of  $PSL_2(\mathbb{Z})$  is defined implicitly by the geometry rather than by specifying its elements explicitly in terms of their matrix coefficients.

Let  $P$  be the interior of the ideal polygon whose vertices are the complex  $d$ -th roots of unity. Note that  $P$  is preserved by the mapping  $\rho(z) = e^{2\pi i/d}z$ , which generates a cyclic subgroup of order  $d$  in the hyperbolic isometry group  $PSL_2(\mathbb{R})$ . Since  $PSL_2(\mathbb{R})$  acts transitively on clockwise-ordered triples of ideal points, we may take any side of the ideal polygon  $\partial P$  and find a hyperbolic isometry  $\sigma$  which maps its endpoints to 1 and  $-1$  while mapping the midpoint of the arc joining these endpoints to  $-i$ . Now consider the two hyperbolic isometries

$$\begin{aligned} a & : z \mapsto -z \\ b & : z \mapsto \sigma(\rho(\sigma^{-1}(z))). \end{aligned}$$

Isometry  $a$  rotates the disk by a half-turn interchanging  $P$  with its complex conjugate  $\overline{P}$ . Isometry  $b$  preserves the set  $P$  while cyclically permuting its corners. These two transformations generate a discrete subgroup  $\Gamma \subset PSL_2(\mathbb{R})$  such that the sets  $\{g(P) : g \in \Gamma\}$  constitute a tiling of the hyperbolic plane by ideal  $d$ -gons. The dual of this tiling is a complete  $d$ -regular tree  $T$  whose nodes are all the points  $\{g(\sigma(0)) : g \in \Gamma\}$ . The complex numbers  $u = \sigma(0), v = -u$  are pure imaginary, and every edge of  $T$  is the image of the edge  $(u, v)$  under a suitable transformation  $g \in \Gamma$ . See Figure 2 for an illustration of the  $d = 4$  case.

The proof that  $T$  is greedily embedded in the hyperbolic plane works as before: the action of  $\Gamma$  on  $T$  is edge-transitive, so it suffices to consider a path in  $T$  whose first hop is  $(u, v)$  and whose endpoint  $t$  lies in the lower half-plane. The proof that  $d(u, z) > d(v, z)$  for every  $z$  in the lower half-plane is exactly the same.

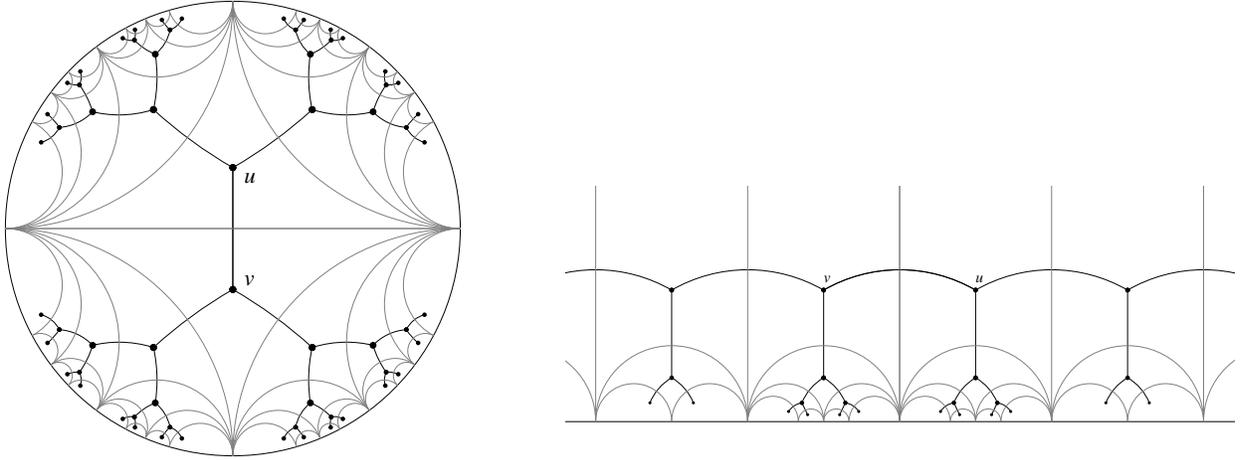


Fig. 1. Greedily embedding a 3-regular tree in the Poincaré disk and upper half plane models of  $H$ .

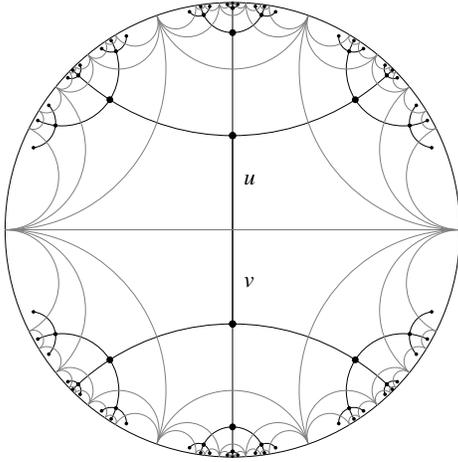


Fig. 2. Greedily embedding a 4-regular tree in  $H$ .

#### D. A distributed algorithm to compute a greedy hyperbolic embedding

In this section we describe a natural and simple distributed algorithm by which nodes can compute their virtual coordinates for the greedy embedding described in Section II-C. We assume that the network is capable of computing a spanning tree rooted at some node  $r$ . (Distributed protocols for computing a spanning tree are abundant in the literature — e.g. [19], [20] — and also in practice.) More specifically, we assume that the network has a distinguished node  $r$  and that each node  $w \neq r$  has chosen one neighbor  $p(w)$  (the “parent” of  $w$ ) such that the edges  $(p(w), w)$  form an arborescence rooted at  $r$ . Let  $d(w)$  be the degree of  $w$ , and let  $0, 1, \dots, d(w) - 1$

be a numbering of the neighbors of  $w$  such that 0 is assigned to  $p(w)$  and  $1, \dots, d(w) - 1$  are assigned to the children of  $w$  in arbitrary order.

Having computed a rooted spanning tree  $T(G)$  of  $G$ , we next compute the maximum degree of  $T(G)$ , via a simple two-pass algorithm in which each node  $w$  reports to its parent the maximum degree of the subtree rooted at  $w$ , and the root computes the maximum degree of  $T(G)$  and broadcasts this information to all other nodes.

Finally, along every edge  $(p(w), w)$  of  $T(G)$ , the parent  $p(w)$  transmits to its child  $w$  the coefficients of a Möbius transformation  $\mu_w$  which maps  $f(p(w))$  to  $u$  and  $f(w)$  to  $v$ , where  $u, v$  are the points of the Poincaré disk defined in Section II-C. The transformations  $\mu_w$  are computed as follows. Recall the Möbius transformations  $a, b$  defined in Section II-C. The root node  $r$  sets  $\mu_r = a$ . For every node  $w$ , including  $r$ , once  $w$  knows the coefficients of the transformation  $\mu_w$ , it computes its virtual coordinate  $f(w) = \mu_w^{-1}(v)$  in the Poincaré disk. It also transmits to each child  $w'$  the coefficients of the Möbius transformation  $\mu_{w'} = b^i \circ a \circ \mu_w$ , where  $i$  is the number which  $w$  assigned to  $w'$  when it labeled its children during the spanning tree computation. To establish the correctness of the algorithm, we must show that for each edge  $(p(w), w)$  of  $T$ , the function  $f$  maps  $p(w)$  and  $w$  to a pair of adjacent nodes in the greedy embedding of the infinite  $d$ -regular tree  $T$  described in Section II-C. Since  $\mu_w$  is an automorphism of  $T$ , it suffices to prove that  $\mu_w(f(p(w)))$  and  $\mu_w(f(w))$  are adjacent nodes of  $T$ . But  $\mu_w(f(w)) = \mu_w(\mu_w^{-1}(v)) = v$  by definition, so it suffices to show that  $\mu_w(f(p(w))) = u$ . Note that  $a \circ \mu_{p(w)}$  maps  $f(p(w))$  to  $a(v) = u$ . Moreover  $u$  is a fixed point of  $b$ , so  $\mu_w = b^i \circ a \circ \mu_w$  maps  $f(p(w))$  to

$u$ , as desired.

### E. Hyperbolic embeddings and data-centric storage

As described in Section I, it is possible to base a Distributed Hash Table (DHT) on the greedy hyperbolic embeddings described above. The implementation of the DHT uses the CAN (Content Addressable Network) paradigm [10], in which a data item is hashed to one or more points in a virtual coordinate space which are used to determine where copies of the data shall be stored and how other nodes may search for the data. In this section we consider a DHT implementation in which the `put` and `get` operations are performed by hashing the key to a point  $p$  in the hyperbolic plane, greedily routing to a node  $x$  whose virtual coordinate is nearer to  $p$  than any of its neighbors, and either storing or retrieving the value from this node. (If redundant storage is desired, one may use a hash function which maps the key to a  $k$ -tuple of points  $p_1, p_2, \dots, p_k$  such that a copy of the value is stored at each of the corresponding nodes  $x_1, x_2, \dots, x_k$ . We will consider the case  $k = 1$  here for simplicity.) In order for this to constitute a reliable DHT implementation, one must ensure that the choice of the node  $x$  does not depend on the source of the `put` or `get` operation. In other words, one needs to prove that there is a unique node  $x$  which is nearer to  $p$  than any of its neighbors. We show that our hyperbolic embedding satisfies this property, for all but a measure-zero subset of points  $p$ .

**Theorem 3.** *Let  $G$  be a graph embedded in the hyperbolic plane  $H$  using the greedy embedding  $f : V(G) \rightarrow H$  described in Section II-C. There is a measure-zero subset  $Y \subseteq H$  such that for every point  $p \in H \setminus Y$ , there is a unique node  $x \in G$  such that  $d(f(x), p) < d(f(y), p)$  for every node  $y$  which is a neighbor of  $x$  in  $G$ .*

*Proof.* Let  $Y$  be the set of all points which are equidistant from two adjacent nodes of the spanning tree  $T$ . This set has measure zero, because for each pair of adjacent nodes, the set of points equidistant from them is isometric to the diameter  $\{\Im(z) = 0\}$  of the Poincaré disk. We claim that if  $p \in H \setminus Y$  and  $x_1, x_2$  are two nodes of  $G$  each of which is nearer to  $p$  than any of its neighbors, then  $x_1 = x_2$ . Assume, by way of contradiction, that  $x_1 \neq x_2$ . Let  $e_1 = (x_1, y_1), e_2 = (y_2, x_2)$  denote the first and last directed hops, respectively, on the path  $P$  in  $T$  from  $x_1$  to  $x_2$ . (Note that  $P$  consists of at least two edges, since  $x_1$  and  $x_2$  cannot be neighbors under our hypothesis that each of them is nearer to  $p$  than any of its neighbors.) We will prove that either

$d(f(x_1), p) > d(f(y_1), p)$  or  $d(f(x_2), p) > d(f(y_2), p)$ , contradicting our hypothesis on  $x_1, x_2$ .

For any pair of adjacent nodes  $s, t$  in  $T$ , let  $N(s, t)$  denote the set of all points  $q$  such that  $d(q, s) < d(q, t)$ . Then  $N(s, t)$  is isometric to the upper half-plane, and its boundary is one of the geodesics separating adjacent tiles of the tiling illustrated in Figure 2. (This is proved by observing that the conclusion is true when  $s = u, t = v$ , that the conclusion is invariant under isometries which preserve  $T$  and preserve the tiling, and that the set of such isometries acts edge-transitively on  $T$ .) Since the geodesics separating adjacent tiles never cross one another, for any two pairs of adjacent nodes in  $T$ , say  $(s_1, t_1)$  and  $(s_2, t_2)$ , we have one of the following relations:

- 1)  $N(s_1, t_1) \cap N(s_2, t_2) = \emptyset$ .
- 2)  $N(s_1, t_1) \cap N(t_2, s_2) = \emptyset$ .
- 3)  $N(t_1, s_1) \cap N(s_2, t_2) = \emptyset$ .
- 4)  $N(t_1, s_1) \cap N(t_2, s_2) = \emptyset$ .

We will apply this when  $(s_1, t_1) = (f(x_1), f(y_1))$  and  $(s_2, t_2) = (f(x_2), f(y_2))$ . Recalling that  $f$  is a greedy embedding of  $T$  and that  $P$  is a path in  $T$  containing at least two edges, we find that  $d(s_1, t_2) < d(s_1, s_2)$ ,  $d(s_2, t_1) < d(s_2, s_1)$ , and  $d(t_2, t_1) < d(t_2, s_1)$ . Hence  $N(s_1, t_1) \cap N(t_2, s_2)$  contains  $s_1$ ,  $N(t_1, s_1) \cap N(s_2, t_2)$  contains  $s_2$ , and  $N(t_1, s_1) \cap N(t_2, s_2)$  contains  $t_2$ . This rules out cases (2)-(4) above. Consequently  $N(s_1, t_1)$  is disjoint from  $N(s_2, t_2)$ , which implies that  $p$  cannot belong to both sets. Hence either  $d(p, f(y_1)) < d(p, f(x_1))$  or  $d(p, f(y_2)) < d(p, f(x_2))$ , contradicting our hypothesis that each of  $f(x_1), f(x_2)$  is closer to  $p$  than any of its neighbors.  $\square$

### III. GREEDY ROUTING IN NORMED VECTOR SPACES

**Lemma III.1.** *If  $X$  is a normed vector space which is admits a greedy embedding of every graph of size  $n$ , then  $\dim(X) = \Omega(\log n)$ .*

*Proof.* Let  $G$  be a star graph consisting of a root vertex  $u$  and  $n - 1$  leaves  $v_1, \dots, v_{n-1}$ , and suppose that  $f : V(G) \rightarrow X$  is a greedy embedding. Assume without loss of generality that  $f(u) = 0$ . For  $i = 1, 2, \dots, n - 1$  let  $x_i = f(v_i)$ ,  $r_i = \|x_i\|$ , and  $w_i = x_i/r_i$ . We claim that  $\|w_i - w_j\| \geq 1$  for  $i \neq j$ . To prove it, suppose  $\|w_i - w_j\| < 1$  and assume without loss of generality that  $r_i \leq r_j$ . We have

$$\begin{aligned} \|x_i - x_j\| &= \|r_i w_i - r_j w_j\| \\ &\leq r_i \|w_i - w_j\| + |r_i - r_j| \|w_j\| \\ &< r_i + (r_j - r_i) = r_j. \end{aligned}$$

Hence  $d(f(v_i), f(v_j)) < d(f(u), f(v_j))$ , contradicting the fact that  $f$  is a greedy embedding.

We conclude that  $\|w_i - w_j\| \geq 1$  for  $i \neq j$ , i.e. the unit sphere in  $X$  contains  $n - 1$  points whose pairwise distances are all at least 1. It follows that a ball of radius  $3/2$  in  $X$  contains at least  $n$  disjoint balls of radius  $1/2$ . Letting  $x$  denote the volume of the unit ball of  $X$ , this implies that  $(3/2)^d x \geq n \cdot (1/2)^d x$ , i.e.  $d \geq \log_3(n)$ .  $\square$

The following lemma is well-known. (For example, see Theorem 5.3 of Linial, London and Rabinovich’s classic paper [8].) The theorem following the lemma is an immediate corollary.

**Lemma III.2.** *For every tree  $T$  with  $L$  leaves, the shortest path metric of  $T$  embeds isometrically in  $\ell_\infty^d$ , where  $d = \lceil \log_{3/2} n \rceil$ .*

**Theorem 4.** *Let  $G$  be a connected graph with  $n$  vertices, and let  $d = \lceil \log_{3/2}(n) \rceil$ . Then there exists a greedy embedding of  $G$  in  $\ell_\infty^d$ .*

#### IV. EVALUATION

Theorem 2 ensures that every finite connected graph has a greedy embedding in the hyperbolic plane, but it does not ensure that this embedding has desirable properties such as low *congestion* (the number of distinct pairs  $s, t$  whose greedy route uses a given node  $v$ ) and low *stretch* (the ratio of the number of hops on a greedy route to the number of hops on the shortest route between the same pair of nodes). In fact, the technique given in Section II for proving that the hyperbolic embedding is a greedy embedding relied only on the fact that the hyperbolic greedy route is no longer than the spanning tree route between two nodes. Hence, a natural hypothesis is that the routes chosen by the greedy algorithm using the hyperbolic embedding may not be very different from the routes in the spanning tree, and they may inherit some of the same undesirable properties, such as high congestion on the popular nodes of the tree. In this section we describe experiments which prove that this is not the case: when the underlying network is a random geometric graph, greedy routes in the hyperbolic embedding take enough short cuts (edges which are not in the spanning tree) to achieve very significant savings on both stretch and congestion.

We generated random unit disk graphs with 50 nodes at independent, uniformly-random locations in a circle of radius  $\sqrt{5}$  (Fig. 3). The average degree of the graphs was 9.8. The failure rate of Euclidean greedy routing was 9.3%. Adding a single barrier — a line segment from  $(-\frac{1}{3}, 0)$  to  $(\frac{1}{3}, 0)$  through which no messages could pass — increased the failure rate of Euclidean greedy routing to 23.3%. We computed a greedy hyperbolic embedding using the minimum spanning tree of the

graph. Of course, hyperbolic greedy routing had failure rate 0 in both cases, in accord with Theorem 2.

The stretch distribution and the congestion distribution for the hyperbolic and Euclidean greedy routing schemes, as well as for routing in the spanning tree, are shown in Fig. 4. Interestingly, hyperbolic greedy routing has very mild stretch, almost as mild as Euclidean greedy routing. For example, the 80th percentile of the stretch distribution is only 1.2 for hyperbolic greedy routing, compared with 1.0 for Euclidean greedy routing and 4.1 for routing in the spanning tree. Similarly favorable results apply to the congestion distribution: the 80th percentile for the hyperbolic routing scheme is only 9.1% more congested than for the Euclidean greedy routing, whereas the 80th percentile of the congestion distribution for the spanning tree routing scheme is 233% more congested.

#### V. OPEN QUESTIONS

Although the experimental results in Section IV confirm that our greedy hyperbolic embedding yields routes with low stretch when applied to typical unit-disk graphs, the worst-case stretch is still linear in the network size. Can every finite graph be embedded in a hyperbolic space of dimension  $O(1)$  — or a normed vector space of dimension  $O(\log n)$  — to yield routes whose stretch is polylogarithmic in  $n$ ? Can constant-stretch routing be achieved using a greedy embedding in a metric space of dimension  $o(n)$ ?

When a node joins or leaves the network, how many operations are required to reconstruct the greedy embedding? On average, our algorithm requires  $O(n)$  operations to respond to each such event. Can one design a distributed greedy embedding algorithm whose amortized complexity, per join/leave event, is  $O(\text{poly} \log n)$  operations? Can one even achieve  $o(n)$  amortized complexity?

Finally, the Papadimitriou-Ratajczak conjecture remains open. Does every 3-connected planar graph admit a greedy embedding in the *Euclidean* plane?

#### REFERENCES

- [1] B. Karp and H. T. Kung, “GPSR: Greedy perimeter stateless routing for wireless networks,” in *Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2000)*, 2000, pp. 243–254.
- [2] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger, “Geometric ad-hoc routing: Of theory and practice,” in *PODC 2003*.
- [3] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, “Routing with guaranteed delivery in ad hoc wireless networks,” *Wireless Networks*, vol. 7, no. 6, pp. 609–616, 2001.
- [4] R. Fonseca, S. Ratnasamy, J. Zhao, C. T. Ee, D. Culler, S. Shenker, and I. Stoica, “Beacon vector routing: Scalable point-to-point routing in wireless sensor networks,” in *Proceedings of the Second USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI 2005)*, 2005.

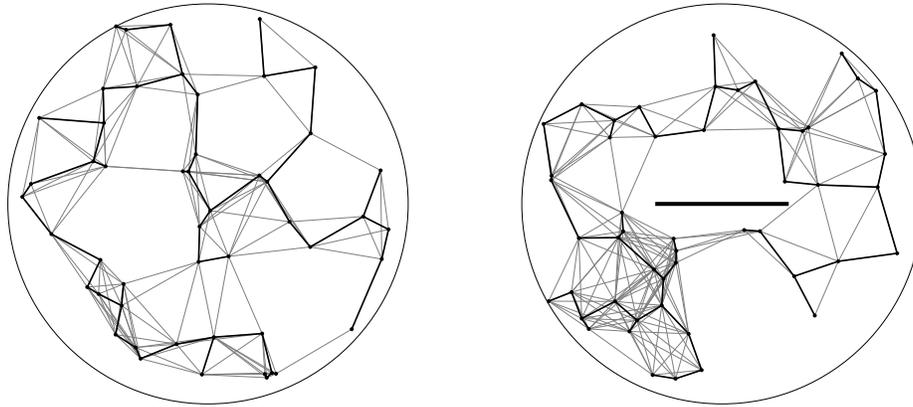


Fig. 3. Random geometric graphs without barrier (left) and with barrier (right).

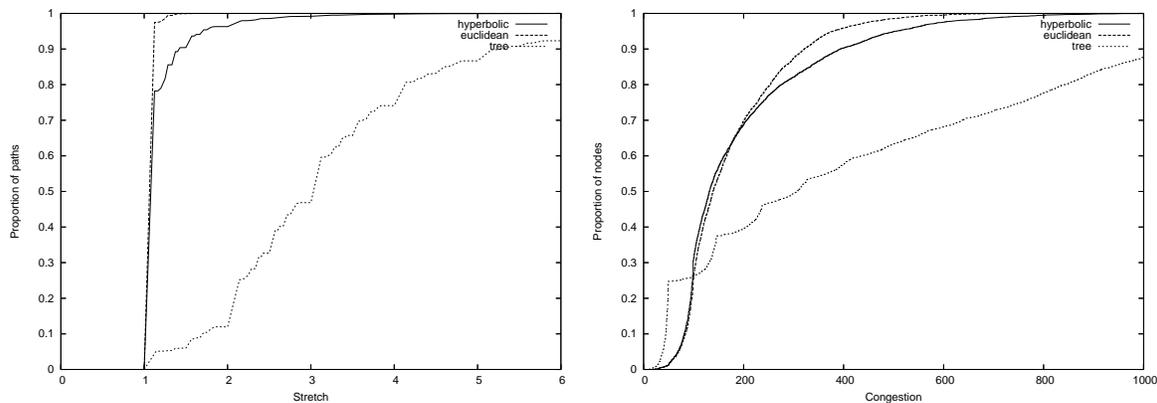


Fig. 4. Stretch and congestion distributions, for hyperbolic and Euclidean greedy routing and for spanning tree routing.

- [5] A. Rao, C. Papadimitriou, S. Shenker, and I. Stoica, "Geographical routing without location information," in *Proceedings of MobiCom 2003*, 2003, pp. 96–108.
- [6] A. Caruso, A. Urpi, S. Chessa, and S. De, "GPS-free coordinate assignment and routing in wireless sensor networks," in *Proceedings of the 24th Conference of the IEEE Communications Society (INFOCOM 2005)*, 2005.
- [7] C. Papadimitriou and D. Ratajczak, "On a conjecture related to geometric routing," *Theoretical Computer Science*, vol. 344, no. 1, pp. 3–14, 2005.
- [8] N. Linial, E. London, and Y. Rabinovich, "The geometry of graphs and some of its algorithmic applications," *Combinatorica*, vol. 15, pp. 215–245, 1995. [Online]. Available: [citeseer.ist.psu.edu/linial94geometry.html](http://citeseer.ist.psu.edu/linial94geometry.html)
- [9] S. Ratnasamy, B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin, and F. Yu, "Data-centric storage in sensornets with GHT, a geographic hash table," *MONET*, vol. 8, no. 4, pp. 427–442, 2003.
- [10] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," in *Proc. ACM SIGCOMM 2001*, 2001.
- [11] J. Gao, L. Guibas, J. Hershberger, L. Zhang, and A. Zhu, "Geometric spanners for routing in mobile networks," in *Proceedings of MobiHoc 2001*, 2001.
- [12] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker, "Geographic routing made practical," in *Proceedings of the Second USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI 2005)*, 2005.
- [13] J. Hill and D. Culler, "Mica: A wireless platform for deeply embedded networks," *IEEE Micro.*, vol. 22, no. 6, pp. 12–24, 2002.
- [14] W. T. Tutte, "Convex representations of graphs," *Proc. London Math. Soc.*, vol. 10, pp. 304–320, 1960.
- [15] N. Alon, M. Badoiu, E. D. Demaine, M. Farach-Colton, M. T. Hajiaghayi, and A. Sidiropoulos, "Ordinal embeddings of minimum relaxation: general properties, trees, and ultrametrics," in *SODA*. SIAM, 2005, pp. 650–659.
- [16] J. P. Cunningham and R. N. Shepard, "Monotone mapping of similarities into a general metric space," *Journal of Mathematical Psychology*, vol. 11, no. 4, pp. 335–364, 1974.
- [17] Y. Bilu and N. Linial, "Monotone maps, sphericity, and bounded second eigenvalue," *Journal of Combinatorial Theory Ser. B*, vol. 95, pp. 283–299, 2005.
- [18] W. P. Thurston, *Three-Dimensional Geometry and Topology*. Princeton University Press, 1997.
- [19] R. Gallager, P. Humblet, and P. Spira, "A distributed algorithm for minimum-weight spanning trees," *ACM Transactions on Programming Languages and Systems*, vol. 5, pp. 66–77, 1983.
- [20] R. Perlman, "An algorithm for distributed computation of a spanning tree in an extended LAN," *ACM SIGCOMM Computer Communication Review*, vol. 15, pp. 44–53, 1985.