Distance Estimation and Object Location via Rings of Neighbors

Aleksandrs Slivkins*

February 7, 2005

Abstract

We approach several problems on distance estimation and object location using a common technique called *rings of neighbors*. Using this technique on metrics of low doubling dimension, we obtain significant improvements for low-stretch routing schemes [22, 1], distance labeling [22, 21, 19], searchable small worlds [10, 11], and triangulation-based distance estimation [12, 21]. Apart from improving the previously known bounds for these problems, our contributions include extending Kleinberg's small world model to metrics of low doubling dimension, and a short proof of the main result in [1]. Doubling dimension is a combinatorial (non-geometric) notion of dimensionality that has recently become popular in theoretical CS literature [9, 14, 15, 22, 12, 21, 1].

A collection of rings of neighbors is a sparse distributed data structure that captures the distances and routing information. The idea is that every node u stores pointers to some nodes called 'neighbors'; these pointers are partitioned into several 'rings', so that for some increasing sequence of balls $\{B_i\}$ around u, the neighbors in the i-th ring lie inside B_i; the radii of these balls and the distribution of neighbors in a given ring depend on the specific application. For metrics of low doubling dimension it has been particularly helpful to combine the following two collections of rings: in the first collection, the cardinalities of the balls B_i grow exponentially, and the neighbors are distributed randomly; in the second collection the *radii* of the B_i's grow exponentially, and the neighbors are distributed geographically. Although used implicitly in several contexts, rings of neighbors have not been articulated as a general proof technique.

Keywords: routing schemes, small-world networks, distance labeling, doubling dimension.

Page count: title page + 10 pages + appendices.

Eligible for the best student paper award.

*Department of Computer Science, Cornell University, Ithaca, NY 14853.
Email: *slivkins@cs.cornell.edu*. Phone: 607-255-5578. Fax: 607-255-4428.

1 Introduction

We approach several problems on distance estimation and object location using a common technique called *rings of neighbors*. Using this technique on metrics of low doubling dimension, we obtain significant improvements for low-stretch routing schemes [22, 1], distance labeling [22, 21, 19], searchable small worlds [10, 11, 17, 18], and triangulation-based distance estimation [12, 21]. Apart from improving the previously known bounds for these problems, our contributions include extending Kleinberg's small world model to metrics of low doubling dimension, and a short proof of the main result in [1].

A collection of rings of neighbors is a sparse distributed data structure that captures the distances and routing information. The idea is that every node u stores pointers to some nodes called 'neighbors'; these pointers are partitioned into several 'rings', so that for some increasing sequence of balls $\{B_i\}$ around u, the neighbors in the i-th ring lie inside B_i; the radii of these balls and the distribution of neighbors in a given ring depend on the specific application. For metrics of low doubling dimension it has been particularly helpful to combine the following two collections of rings: in the first collection, the cardinalities of the balls B_i grow exponentially, and the neighbors are distributed randomly; in the second collection the *radii* of the B_i's grow exponentially, and the neighbors are distributed geographically. Although simpler forms of rings of neighbors have been used implicitly in several contexts (e.g. [10, 11, 21]), rings of neighbors have not been realized as a general proof technique.

Doubling dimension is a combinatorial (non-geometric) notion of low dimensionality that has recently become popular in theoretical CS literature [9, 14, 15, 22, 12, 21, 1] in many different contexts, including metric embeddings, traveling salesman and compact data structures; in particular, in [12, 21] it was used to model the structural properties of the Internet distance matrix in the context of distributed algorithms for embeddings and distance estimation.

Doubling dimension is defined as the smallest α such that every ball $B_u(r)$ can be covered by 2^α balls of radius $r/2$. Metrics of low doubling dimension are a non-trivial generalization of low-dimensional Euclidean metrics [9].[1] Moreover, they subsume the family of metrics with bounded *growth ratio*,[2] which has been widely considered in the context of object location in peer-to-peer networks (see [6] for a short survey). Unlike a weaker notion of dimensionality based on the growth rate of balls, the doubling dimension is robust, in the sense that the dimension of a subset cannot be larger than the dimension of the entire metric.

Low-stretch routing schemes. Informally, a *routing scheme* on a network is a distributed algorithm that takes packets from source to destination. Formally, in a routing scheme every node u is assigned a *label* and a *routing table*; moreover, there is an efficient algorithm that does the following two things:

- given the packet header and the routing table of the current node determines the next hop, and
- given the routing table of a source node s and the routing label L of target t, forms the packet header (containing L) so that the packet reaches t starting from s.

We assume that the underlying connectivity graph of the network is fixed, i.e. cannot be changed by our algorithm. The routing scheme has *stretch* $1 + \delta$ if for any source-target pair the packet follows a path of length at most $1 + \delta$ times that of the shortest path. There has been a number of results for compact routing schemes, see [23, 4] for a survey. In particular, the seminal paper of Plaxton et al. [24] started a line of work on routing schemes for peer-to-peer overlay networks, with provable guarantees for metrics with bounded growth ratio (see [6] for a short survey). More relevantly to this paper, Talwar [22] considered routing schemes for a more general class of metrics – metrics of low doubling dimension; his result has been improved by Chan et al. [1].

[1]More precisely, any finite point set in a k-dim Euclidean metric has doubling dimension $O(k)$.

[2]Growth ratio is defined as $\max_{(u,r)} |B_u(2r)|/|B_u(r)|$, where $B_u(r)$ is the closed ball of radius r around node u.

	routing table size, bits	packet header size, bits
Talwar [22]	$O(\frac{1}{\alpha\delta})^{\alpha}(\log^{2+\alpha}\Delta)$	$O(\alpha\log\Delta)$
Chan et al. [1]	$(\frac{\alpha}{\delta})^{O(\alpha)}(\log\Delta)(\log D_{\text{out}})$	$O(\alpha\log\frac{1}{\delta})(\log\Delta)$
Thm. 2.1	$(\frac{1}{\delta})^{O(\alpha)}(\log\Delta)(\log D_{\text{out}})$	same as above
Thm. 3.5a	$(\frac{1}{\delta})^{O(\alpha)}(\log\Delta)(\phi\log n)$	$2^{O(\alpha)}(\phi\log n)$
Thm. 3.5b	$(\frac{1}{\delta})^{O(\alpha)}(N_\delta+\phi\log n)(\log D_{\text{out}})$	$O(N_\delta+\phi\alpha\log n)(\log D_{\text{out}})$

NOTATION: doubling dimension α, aspect ratio Δ, maximal out-degree D_{out}, $\phi=\log(\frac{1}{\delta}\log\Delta)$.
For any two nodes uv there exists a $(1+\delta)$-approximate shortest uv-path with at most N_δ hops.
The $(\log D_{\text{out}})$ factors are negligible since nodes keep $\Omega(D_{\text{out}})$ state to keep track of their out-links.

Table 1: $(1+\delta)$-stretch routing schemes for metrics of low doubling dimension.

OUR CONTRIBUTIONS: Using rings of neighbors, we give a significantly shorter and simpler proof of the result in [1], with slightly improved guarantees (Thm. 2.1). We also note that these guarantees are unsatisfactory for metrics with large (e.g. exponential) aspect ratio Δ (which is the ratio between the diameter and the smallest distance), and provide two extensions (Thm. 3.5ab) where we alleviate the dependency on Δ; see Table 1 for details. The first extension removes the factor of $(\log\Delta)$ from the packet header size; the second one is tailored to the case when there exist near-shortest paths with small hop-counts. Both extensions combine the ideas from Thm. 2.1 with those from our result on distance labeling that we discuss below.

Searchable small-world networks. The small-world networks have been an active topic in many branches of social and natural sciences. The 'small-world phenomenon', also known as the 'six degrees of separation' have been discovered in a seminal work of Milgram [20] and recently confirmed by Dodds et al. [2]. Motivated by Watts and Strogatz [26], Kleinberg [10] has articulated another striking aspect of 'small worlds': that a greedy search algorithm can find short paths to most targets using only local information. Kleinberg went on to suggest several mathematical models where this happens [10, 11] (see also [17, 16, 18]). In particular, for any metric with bounded growth-rate of balls he proved that if every node chooses $O(\log^2 n)$ neighbors from a fairly natural distribution, then with high probability a local greedy search algorithm finds $O(\log n)$-hop paths for every query.

OUR CONTRIBUTIONS: We extend Kleinberg's model to metrics of low doubling dimension. While it is relatively straightforward to achieve out-degree $O(\log n)(\log\Delta)$ and $O(\log\Delta)$-hop paths, where Δ is the aspect ratio of the metric, it is quite non-trivial to handle the case of super-polynomial aspect ratio. We obtain $O(\log n)$-hop paths even if Δ is exponential in n. In our first result the out-degree is (still) proportional to $\log(\Delta)$, suggesting that it is a natural lower bound. However, our second (and much more complicated) result breaks this barrier, achieving the out-degree $O(\log^2 n)\sqrt{\log\Delta}$. This result uses a local search algorithm that jumps 'sideways' whenever it cannot make good progress towards the target. To the best of our knowledge this is the first small-world model with a *non-greedy* local search algorithm.

Distance labeling. In a distance labeling scheme, each node is assigned a short label so that the distance between any two nodes can be efficiently approximated just by looking at their labels. Formally, a $(1+\delta)$-approximate labeling scheme for a class $\mathcal{M}$ of metrics consists of a polynomial-time computable real-valued function $f(x,y)$ and, for each metric $M\in\mathcal{M}$, an assignment of labels L_u to nodes u of M such that for each node pair uv, $f(L_u,L_v)$ is within factor of $1+\delta$ of the true uv-distance.

It is known that distance labeling scheme on general graphs requires $\Omega(n)$-bit labels, which improves to $\Omega(n^{1/3})$ bits on planar graphs (see [23, 4] for a survey on distance labeling). Major improvements are possible for metrics of low doubling dimension. Denote $K = [O(\frac{1}{\delta})]^{O(\alpha)}$, where α is the doubling dimension and Δ is the aspect ratio. Talwar [22] proved that every metric has a $(1+\delta)$-approximate distance labeling scheme with $K(\log\Delta)$ bits per label. Slivkins [21] observed that since $(\log\Delta)$ can be arbitrarily large with respect to n, it is desirable to replace it with (poly-)$\log n$; he gave a construction with $K(\log^2 n)\log(n\log\Delta)$ bits per label.[3] This has been improved by a factor of $(\log n)$ in Mendel and Har-Peled [19].

OUR CONTRIBUTION: We obtain the result in [19] as a simple corollary of our result on triangulation. We improve it to $O(K\log n)(\log\log\Delta)$ bits per label using the ideas from our first result on routing schemes. This matches the lower bound proved in [19].[4]

Triangulation. Motivated by the systems of [7, 8, 13, 3] for estimating Internet latencies via the triangle inequality, Kleinberg et al. [12] introduced the notion of *triangulation* on a metric. Triangulation [12, 21] of order k is defined as a labeling of the nodes such that a label of a given node u consists of distances from u to each node in a set S_u of at most k other nodes. Then given the labels of two nodes u and v, one can use the triangle inequality to upper-bound the uv-distance by $D^+_{uv} = \min(d_{ub}+d_{vb})$, and lower-bound it by $D^-_{uv} = \max|d_{ub}-d_{vb}|$, where the max and min are taken over all $b \in S_u \cap S_v$. An (ϵ,δ)-*triangulation* is a triangulation such that $D^+_{uv}/D^-_{uv} \leq 1+\delta$ for all but an ϵ-fraction of node pairs uv. In particular, this inequality holds whenever there exists some node $b \in S_u \cap S_v$ that lies within distance $\delta d_{uv}/3$ from u or v. Note that if it holds then either bound can be seen as a $(1+\delta)$-approximate estimate on the uv-distance, and, moreover, these bounds provide a "quality certificate" for the estimate.

Papers [12, 21] considered distributed algorithms for constructing low-order (ϵ,δ)-triangulations on metrics of low doubling dimension. An obvious flaw of such triangulations is that they provide no guarantees for a constant fraction of node pairs. Accordingly, Slivkins [21] also considered $(0,\delta)$-triangulations and gave a construction of $(0,\delta)$-triangulation of order $O_{\alpha,\delta}(\log^2 n)$, where α is the doubling dimension.

OUR CONTRIBUTION: We construct a $(0,\delta)$-triangulation of order $O_{\alpha,\delta}(\log n)$.

Further directions. Rings of neighbors can be used in a distributed system as a layer that supports various applications. In particular, this is the framework used theoretically in [21] for distributed approaches to embedding and distance estimation, and practically in *Meridian* [25], a system for nearest-neighbor and multi-range queries in a peer-to-peer network. While this framework has already lead to significant results, we can define the rings that are much 'better' than the ones that we know how to construct and maintain in a distributed fashion. Bridging this gap leads to a set of intriguing open questions.

Preliminaries. Denote the uv-distance by d_{uv}. Let $B_u(r)$ be the closed ball of radius r around u. Let $r_u(\epsilon)$ be the radius of the smallest closed ball around u that contains at least ϵn nodes. For $k \in \mathbb{N}$ define $[k]$ as the set $\{0, 1 \ldots k-1\}$. Define an *enumeration* of a finite set S as a bijection $[k] \to S$, where $k = |S|$. Throughout the paper, n denotes the number of nodes, α denotes the doubling dimension, and Δ denotes the *aspect ratio*, which is the ratio of diameter to the smallest distance. Without loss of generality we'll assume that the smallest distance is 1, and the diameter is Δ.

Say a measure is *s-doubling* [5] if for any ball $B_u(r)$ its measure is at most s times larger than that of $B_u(r/2)$. It is known [5] that for any metric there exists a 2^α-doubling measure μ. Intuitively, a doubling measure is an assignment of weights to nodes that makes a metric look growth-constrained; in particular, for *exponential line*, a one-dimensional set $\{2^i : i \in [n]\}$, we have $\mu(2^i) = 2^{i-n}$.

[3] The conference version of [21] erroneously claims $(K\log^2 n)(\log\log\Delta)$ bits per label; this has been corrected in full version.

[4] The current version of [19] erroneously claims the lower bound of $\Omega(\log n)\log(n\log\Delta)$ bits per label for metrics of unit doubling dimension; what they actually prove is $\Omega(\log n)(\log\log\Delta)$ bits per label.

For $r > 0$ an *r-net* on a metric is a set S such that any point of the metric is at distance at most r from S, and any two points in S are at distance at least r. It is easy to see that for a finite metric such set exists and can be constructed greedily. The following easy fact is well-known:

Lemma 1.1 *Any r-net has at most $O(r'/r)^{O(\alpha)}$ elements in any ball of radius $r' \geq r$.*

2 A low-stretch routing scheme for metrics of low doubling dimension

In this section we'll use rings of neighbors to derive a significantly shorter and simpler standalone proof of the main result in Chan et al. [1]. The ideas from this proof will be used in the subsequent results.

Recall that the underlying connectivity graph is fixed; let D_{out} be its maximal out-degree. Note that nodes need $\Omega(D_{\text{out}})$ state in order to keep track of their out-links, so the $(\log D_{\text{out}})$ factor in the size of our routing tables is negligible.

Theorem 2.1 *Consider a metric with doubling dimension α and aspect ratio Δ. Then for any $\delta \in (0, \frac{1}{2}]$ there is a $(1+\delta)$-stretch routing scheme with $O(\alpha \log \frac{1}{\delta})(\log \Delta)$-bit packet headers and routing tables of $[O(\frac{1}{\delta})]^{O(\alpha)}(\log \Delta)(\log D_{out})$ bits. Moreover, such scheme can be efficiently computed.*

Proof: For each $j \in [\log \Delta]$ let $\mathcal{G}_j$ be some $\Delta/2^j$-net; let $r_j = 4\Delta/\delta 2^j$ and define the j-th ring of neighbors as $Y_{uj} = B_u(r_j) \cap \mathcal{G}_j$. Note that by Lemma 1.1 each node has at most $K = [O(\frac{1}{\delta})]^{O(\alpha)}$ j-ring neighbors. Let f_{uj} be some fixed element of Y_{uj} that lies within distance $\Delta/2^j$ from u. The sequence $\{f_{uj} : j \in [\log \Delta]\}$ is called the *zooming sequence* of u.

Claim 2.2 *$f_{vj} \in Y_{uj}$ whenever $j \leq \log(\Delta/\delta d_{uv})$. In particular, $f_{vj} \in Y_{fj}$, where $f = f_{(v,j-1)}$.*

Proof: By definition $f_j(v) \in \mathcal{G}_j$; it is easy to check that $f_j(v)$ lies within distance r_j from u. □

DATA STRUCTURES. To reduce the size of the routing tables, we do not use $\lceil \log n \rceil$-bits global node identifiers. Instead, each node u maintains some enumeration $\varphi_{uj}(\cdot)$ of each ring Y_{uj}. Since the rings Y_{u0} coincide for all nodes u, without loss of generality so do the enumerations φ_{u0}. For each $v \in Y_{uj}$ node u will need to translate between the *host enumeration* φ_u and the *foreign enumeration* φ_v. Specifically, for each $j \in [\log \Delta]$ node u will store the translation function $\zeta_{uj} : [K] \times [K] \to [K]$ such that

$$\zeta_{uj}\left(\varphi_{uj}^{-1}(v), \varphi_{vj}^{-1}(w)\right) = \varphi_{uj}(w) \text{ whenever } v \in Y_{uj} \text{ and } w \in Y_{uj} \cap Y_{vj},$$

and *null* otherwise. Clearly, each such function can be stored using $K^2 \log K$ bits.

Each node u also fixes some enumeration $\phi_u(\cdot)$ of all outgoing links. For two nodes uv, define the *first-hop pointer* from u to v as $\phi_u(w)$ such that uw is the first edge of some shortest uv-path. The *routing table* of u consists of the first-hop pointers to all its neighbors; each such pointer can be stored using only $\lceil \log D_{\text{out}} \rceil$ bits.

Encode each zooming sequence via host enumerations of its elements as follows: let $n_{u0} = \phi_{u0}(f_{u0})$, and for each $j \geq 1$ let $n_{uj} = \phi_{fj}(f_{uj})$, where $f = f_{(u,j-1)}$. Let n_u be the sequence of all n_{uj}'s, and let j_u be the smallest j such that $f_{uj} = u$. The *routing label* of u is defined as a pair (n_u, j_u). It is easy to see that each routing label can be stored using $O(\log K)(\log \Delta)$ bits, as claimed.

Fix a node pair ut. By Claim 2.2 for $j = \lfloor \log(\Delta/\delta d_{ut}) \rfloor$ node u has a neighbor $v = f_{tj} \in Y_{uj}$ that lies within distance δd_{ut} from t. Essentially, if node u needs to send a packet to t, it wants to forward the packet to this node v. However, we know such v exists, but it is non-trivial to identify it since we do not have global node ids. In our context, to *identify* a neighbor $v \in Y_{uj}$ of u means to find $\varphi_{uj}(v)$.

We'll identify this v using the zooming sequence of t. Let j_{ut} be the maximum j such that $f_{ti} \in Y_{ui}$ for each $i \leq j$. Let g_{utj} be the first-hop pointer from u to f_{tj}. We'll need the following claim:

Claim 2.3 *Given the routing table of u and the routing label of t we can find j_{ut} and g_{utj} for each $j \leq j_{ut}$.*
Proof: Let $m_j = \varphi_{uj}(f_{tj})$. We'll use induction on j to compute m_j for all $j \leq j_{ut}$. Host enumerations φ_{u0} coincide for all nodes u, so $m_0 = n_{u0}$. Suppose for some $j < j_{ut}$ we know m_j and we'd like to compute m_{j+1}. Let $f = f_{tj}$ and $v = f_{(t,j+1)}$. Since we know $m_j = \varphi_{uj}(f)$ and $n_{tj} = \varphi_{fj}(v)$, we can find $m_{j+1} = \varphi_{(u,j+1)}(v)$ using the translation function ζ_{uj}. We iterate the above procedure while we can, i.e. while $v \in Y_{(u,j+1)}$. We stop exactly at $j = j_{ut}$. □

ROUTING ALGORITHM. For a packet with target t, the header consists of the *primary* and *intermediate* targets. The primary target is t itself; the header stores the routing label of t. The intermediate target is f_{tj} for some j; the header stores this j. When some node u initially sends the packet to target t, it computes $j = j_{ut}$ and g_{utj}, chooses f_{tj} to be the intermediate target, and sends the packet along the hop g_{utj}.

Suppose node u receives a packet with primary target t and intermediate target f_{tj}. Then u computes j_{ut}; we'll prove that $j_{ut} \geq j$. Then u computes g_{utj} and, in particular, checks whether the intermediate target is u itself. If not, u just forwards the packet along the hop g_{utj}. Else either $u = t$ (i.e. $j = j_t$) and we are done, or u needs to select a new intermediate target. In the latter case u resets the current level to $j = j_{ut}$, recomputes g_{utj} and forwards the packet along this hop. This completes the routing algorithm.

Claim 2.4 *Here are some key properties of the routing algorithm:*
(a) *each intermediate target is at least $\frac{1}{\delta}$ times closer to the target than the previous one.*
(b) *if node v receives a packet with primary target t and intermediate target f_{tj} then $j_{vt} \geq j$.*
(c) *each packet follows a shortest path to each intermediate target.*

Proof: **(a)** The next intermediate target is chosen when the current intermediate target u is reached; it is defined as f_{tj} such that $j = j_{ut}$. By Claim 2.2 $j \geq \lfloor \log(\Delta/\delta d_{ut}) \rfloor$, so f_{tj} lies within distance δd_{ut} from t.

(b) Let P be this packet. We'll use induction on the path traversed by P. This path starts when some node u chooses $w = f_j(t)$ as an intermediate target; then the current level is set to j_{ut}. For the induction step, assume node v receives P from some node u such that $j_{ut} \geq j$; we need to show that $j_{vt} \geq j$, too. Since $w \in Y_{uj}$, and by the specs $d_{vw} < d_{uw} \leq r_j$, it follows that $w \in Y_{vj}$. It remains to show that $f_{ti} \in Y_{ui}$ for every $i < j$. Indeed, by the triangle inequality $d_{vt} \leq d_{vw} + d_{wt} < r_j + \Delta/2^j$. It is easy to check that $d_{vt} + \Delta/2^i \leq r_i$. Therefore, $f_{ti} \in B_t(\Delta/2^i) \subset B_v(r_i)$.

(c) More precisely, we need to show that if node u sends a packet P with intermediate target w then P reaches w and traverses path of total length d_{uw}. Indeed, by part (b) node u sends P along the first hop of some shortest uw-path. Therefore before reaching w the distance to w decreases on every hop, so P never visits the same node twice. Since the network is finite, P reaches w eventually.

Let $\rho(u)$ be the path traversed by P from u to w. We'll prove that $\rho = d_{uw}$ using induction on this path. Indeed, suppose P is directly forwarded from u to some node $v \neq w$. Then uv is the first hop of a shortest uw-path, so $d_{uw} = d_{uv} + d_{vw}$; by induction hypothesis $\rho(v) = d_{vw}$. □

PROOF OF CORRECTNESS. Consider a packet send by node u to target t. By Claim 2.4b the algorithm is well-defined. By Claim 2.4c the packet reaches each intermediate target, and by Claim 2.4a it reaches t. The distance from the i-th intermediate target to t is at most $\delta^i d_{ut}$ by Claim 2.4a, so by Claim 2.4c the total path length is at most $\sum_{i=0} \delta^i d_{ut}(1+\delta) \leq d_{ut}(1+O(\delta))$, theorem proved. □

3 Triangulation, distance labeling, and improved routing schemes

We start this section with our result on triangulation; then we elaborate it using the ideas from the previous section and achieve an optimal $(1+\delta)$-approximate distance labeling scheme; then we use the techniques from this distance labeling scheme to obtain improved routing schemes for metrics with large aspect ratio.

Our result on triangulation builds on the following generalization of r-nets which is implicit (but never articulated) in Section 5 of [21]:

Lemma 3.1 *Consider a complete (and possibly infinite) metric of doubling dimension α, equipped with a probability measure μ. Let $r_u(\epsilon)$ be the radius of the smallest ball around u that has measure ϵ. Then for any $\epsilon > 0$ there exists an (ϵ, μ)-*packing*: a family $\mathcal{F}$ of disjoint balls, of measure at least $\epsilon/2^{O(\alpha)}$ each, such that for any node u the ball $B_u[6r_u(\epsilon)]$ contains some ball from $\mathcal{F}$.*

It is easy to see that if μ is a doubling measure then locally for every node u this (ϵ, μ)-packing $\mathcal{F}$ has the two useful properties of an r-net, $r = 6r_u(\epsilon)$: firstly, the ball $B_u(r)$ contains at least one element of $\mathcal{F}$, and secondly, for any k the ball $B_u(kr)$ contains at most $k^{O(\alpha)}$ elements of $\mathcal{F}$. Therefore (ϵ, μ)-packings can be seen as a generalization of r-nets to arbitrary probability measures. In this section we will use (ϵ, μ)-packings, where μ is the counting probability measure $\mu(S) = |S|/n$.

Theorem 3.2 *For any $\delta \in (0, 1)$ any metric has a $(0, \delta)$-triangulation of order $[O(\frac{1}{\delta})]^{O(\alpha)} \log n$, where α is the doubling dimension. Moreover, such triangulation can be efficiently computed.*
Proof: The label of every node u will consist of distances to a subset of nodes which we call the *neighbors* of u. These neighbors will be partitioned into two types of rings: there will be X_i*-neighbors* and Y_i*-neighbors*, $i \in [\log n]$. All X_i-neighbors and all Y_i-neighbors of u will be contained in the open balls $B_{(u,i-1)}$ and $B_u(12r_{ui}/\delta)$, respectively, where $r_{ui} = r_u(2^{-i})$ and $B_{ui} = B_u(r_{ui})$. This is the construction:

- For each $i \in [\log n]$ let $\mathcal{F}_i$ be a $(2^{-i}, \mu)$-packing guaranteed by Lemma 3.1, where μ is the counting probability measure. Fix one point $h_B \in B$ for every ball $B \in \mathcal{F}_i$. Define the X_i-neighbors of u as all nodes h_B such that $B \subset B_{(u,i-1)}$.
- Let Δ be the aspect ratio of the metric, and for each $j \in [\log \Delta]$ let $\mathcal{G}_j$ be some 2^j-net. For each $i \in [\log n]$ define the Y_i-neighbors of u as all nodes in $B_u(12r_{ui}/\delta)$ that lie in $\mathcal{G}_j$ for some $j \in J_{ui}$, where J_{ui} is the set of all integers between $\lfloor \log(\delta r_{ui}/4) \rfloor$ and $\lceil \log(6r_{ui}) \rceil$.

Let's bound the number of neighbors of some fixed node u. Since each ball $B_{(u,i-1)}$ contains at most $2^{O(\alpha)}$ balls $B \in \mathcal{F}_i$, there are at most $2^{O(\alpha)}$ X_i-neighbors for each i. Moreover, by Lemma 1.1 each $\mathcal{G}_j$ contributes at most $(12r_{ui}/\delta 2^j)^{O(\alpha)}$ Y_i-neighbors. Summing this up for all $j \in J_{ui}$ upper-bounds the total number of Y_i-neighbors by $[O(1/\delta)]^{O(\alpha)}$. It remains to prove that our construction is indeed a $(0, \delta)$-triangulation. First we need a basic fact about the radii r_{ui}:

Claim 3.3 *For any edge uv and any $i \in [logn]$ we have $|r_{ui} - r_{vi}| \leq d_{uv}$.*
Proof: Since $B_v(r_{vi}) \subset B_u(d + r_{vi})$, the latter ball contains at least $n/2^i$ nodes, so it follows that $r_{ui} \leq d_{uv} + r_{vi}$. Similarly, since $B_{ui} \subset B_v(d_{uv} + r_{ui})$ it follows that $r_{vi} \leq d_{uv} + r_{ui}$. □

Fix a node pair uv and let $d = d_{uv}$. We need to show that a ball of radius δd around either u or v contains a common neighbor of both u and v. Suppose there is no such node. Let $r = (1 + \delta)d$ and choose i such that $r_{ui} < r + d \leq r_{(u,i-1)}$. We choose i with respect to u, but by Claim 3.3 this yields some bounds on r_{vj}'s as well; specifically, $r_{(v,i-1)} \geq r$ and $r_{vi} \leq r + 2d$.

First we make use of the X_i-neighbors. The ball $B_v(6r_{vi})$ contains some $B \in \mathcal{F}_i$, so in particular it contains some node $w \in X_i$. If $6r_{vi} \leq \delta d$ then $B_v(6r_{vi})$ is contained in both $B_{(u,i-1)}$ and $B_{(v,i-1)}$, hence is an X_i-neighbor of u and v, contradiction. Similarly, $6r_{ui} < \delta d$ then $B_u(6r_{ui})$ contains some $w \in X_i$ which is an X_i-neighbor of u and v, contradiction. Therefore letting $x = \delta d/6$ we have $x \leq r_{ui} \leq r + d$ and $x < r_{vi} \leq r + 2d$. We will use (all of) these four conditions to show that the Y_i-neighbors give us the desired common neighbor.

Indeed, consider the ball $B = B_v(\delta d)$ and let $j = \lfloor \log(\delta d) \rfloor$. Then there exists a node $w \in \mathcal{G}_j \cap B$. Now since $r_{ui} \geq x$ it follows that $B \subset B_u(12 r_{ui}/\delta)$ and $j \leq \log(6 r_{ui})$; moreover, $j \geq \lfloor \log(\delta r_{ui}/4) \rfloor$ since $r_{ui} \leq r + d$. Therefore by definition w is a Y_i-neighbor of u. Similarly, w is a Y_i-neighbor of v, contradiction.[5] Theorem proved. □

Our $(0,\delta)$-triangulation can be extended to a $(1+\delta)$-approximate distance labeling scheme where each label consists of $[O(\frac{1}{\delta})]^{O(\alpha)}(\log n)\log(n \log \Delta)$ bits, matching the result of Mendel and Har-Peled [19]. Indeed, if we assign each node u a unique $\lceil \log n \rceil$-bit identifier $\mathrm{ID}(u)$, then each neighbor u of v can be stored as a pair $(\mathrm{ID}(u), d_{uv})$, and it suffices to store d_{uv} as a $O(\log \frac{1}{\delta})$-bit mantissa and $O(\log\log\Delta)$-bit exponent. Moreover, with some extra work we can get rid of the $(\log n)$-bit node identifiers, shaving off (essentially) a factor of $(\log n)$ from the label length and matching the lower bound in [19].

Theorem 3.4 *For any $\delta \in (0,1)$ any metric has a $(1+\delta)$-approximate distance labeling scheme where each label consists of $[O(\frac{1}{\delta})]^{O(\alpha)}(\log n)(\log\log\Delta)$ bits, where α is the doubling dimension and Δ is the aspect ratio. Moreover, such scheme can be efficiently computed.*

Proof Sketch: We'll elaborate the construction in the proof Thm. 3.2 using the ideas from the proof of Thm. 2.1. Specifically, we'll use the zooming sequences and the host/foreign enumeration technique.

Keep the notation from the proof of Thm. 3.2. Recall that for each $j \in [\log \Delta]$ we fix some 2^j-net $\mathcal{G}_j$. For each node u and each $i \in [\log n]$ fix a node $f_{ui} \in \mathcal{G}_l$, $l = \lfloor \log(r_{ui}/4) \rfloor$, that lies within distance $\delta r_{ui}/4$ from u. Such node is a Y_i-neighbor of u by definition of the Y_i-neighbors; it is possible that $f_{ui} = u$. Call the sequence $\{f_{ui} : i \in [\log n]\}$ a *zooming sequence*, and denote it f_u. Moreover, for each node u fix some enumeration $\varphi_u(\cdot)$ of all its neighbors.

From the proof of Thm. 3.2 we know that for any given node pair uv there exists a node w_0 within distance δd_{uv} from u or v such that w_0 is a common neighbor of u and v; recall that distances from w_0 to u and v give us a desired estimate. However, we know such w_0 exists, it is non-trivial to identify it since we do not have global node ids. In our context, to *identify* a common neighbor w of u and v means to find $\varphi_u(w)$ and $\varphi_v(w)$.

Suppose w_0 is within distance δd_{uv} from v; then, essentially, we identify it by zeroing in on v via the sequence f_v. We'll be able to identify, sequentially, all f_{vi} from $i = 0$ to some i_0 such that $f = f_{vi_0}$ lies "reasonably close" to v; each f_{vi} will help us identify $f_{(v,i+1)}$. Then f will help us identify w_0.

The problem is that $f_{(v,i+1)}$ might not be a neighbor of f_{vi}, and w_0 might not be a neighbor of f, so we cannot use the host/foreign enumeration technique the way it is used to prove Thm. 2.1. Instead, for every node we'll define another set of nodes called *virtual neighbors*, so that each $f_{(v,i+1)}$ is a *virtual* neighbor of f_{vi}, and w_0 is a *virtual* neighbor of f. These virtual neighbors are used for enumeration only: for each node u fix some enumeration $\psi_u(\cdot)$ of all its virtual neighbors, and define a pointer from one 'true' neighbor u to another 'true' neighbor v as $\psi_u(v)$. It follows that virtual neighbors are cheap: if every node has at most N of them, then each pointer between the 'true' neighbors will use only $\log N$ bits. We'll actually have $N = O_{\alpha,\delta}(\log n)\log(\Delta)$.

The crux of the proof is to define the virtual neighbors and prove that they have the desired properties. This is quite non-trivial even using this relatively large value of N; see Appendix A for the details. □

We note that the bounds in Thm. 2.1 are unsatisfactory for metrics with large aspect ratio, and suggest two extensions that alleviate the dependency on Δ. Here we only sketch the proofs; full proofs are in Appendix A. Note that given Thm. 3.4, the proof of Thm. 3.5a is remarkably short and simple.

We need some notation. Let $\phi = \log(\frac{1}{\delta}\log\Delta)$; let D_{out} be the maximal out-degree. Let N_δ be the smallest N such that for any two nodes uv there exists a $(1+\delta)$-approximate shortest uv-path with at most N hops. The second result is tailored to the case when N_δ is small.

[5] Note that similarly we can also prove that u and v have a common Y_i-neighbor in the ball $B_u(\delta d)$.

Theorem 3.5 *In Thm. 2.1, we can also construct a* $(1+\delta)$*-stretch routing scheme with*
(a) $2^{O(\alpha)}(\phi \log n)$*-bit packet headers and routing tables of* $[O(\frac{1}{\delta})]^{O(\alpha)}(\phi \log n)(\log \Delta)$ *bits.*
(b) $O(K)$*-bit headers and* $K(\frac{1}{\delta})^{O(\alpha)}$*-bit routing tables, where* $K = (N_\delta + \phi\alpha \log n)(\log D_{out})$.

Proof Sketch: **(a)** We plug Thm. 3.4 into the construction in Thm. 2.1. Specifically, we'll redefine the packet headers: instead of the zooming sequence f_t, where t is the target, we'll store the label of t in a $3/2$-approximate distance labeling scheme guaranteed by Thm. 3.4. The routing algorithm is similar to the one in Thm. 2.1, except we choose new intermediate targets by looking at the corresponding distance labels.

(b) We'll combine the ideas of Thm. 3.4 and Thm. 2.1, and add some new tricks. In particular, we'll use (i) the basic rings of neighbors, (ii) zooming sequences and intermediate targets, (iii) the first-hop pointers, and (iv) host/virtual enumerations. We'll use the rings, the zooming sequences, and the enumerations as defined in Thm. 3.4. In fact, we'll just use all definitions from the proof Thm. 3.4, for the same value of δ, including the sets of X_i- and Y_i-neighbors. We also need a unique $\lceil \log n \rceil$-bit identifier $\textsc{id}(u)$ for every node u.

The routing will have two *modes*. One is essentially from Thm. 2.1: if the intermediate target is not set, consider the smallest i such that the element $f = f_{ti}$ of zooming sequence f_t is sufficiently close to the target t. If it is possible to *identify* this f (i.e. to find $\varphi_u(f)$, where u is the current node), then we designate f as the intermediate target, put i and $\textsc{id}(t)$ in the packet header, and try to route to f using the first-hop pointers. However, this routing algorithm might fail since it might not be possible to identify f at u or at some intermediate node. We'll see that this can happen only if for the current node u and some i such that $r_{ui} < 1.25 d_{ut} \leq r_{(u,i-1)}$ it is the case that r_{ui} is much smaller than d_{ut}. In this case we'll go into the *second routing mode*, which we describe below.

By Lemma 3.1 there exists a ball $B \in \mathcal{F}_i$ of cardinality at least $n/2^{i+O(\alpha)}$ such that $B \subset B_u(6r_{ui})$. Let $h = h_B$ be the node selected from B in Thm. 3.2 (e.g. without loss of generality it could be the center of B). It is easy to see that the ball $B' = B_{(h,i-1)}$ contains target t. The nodes in B will collectively store the routes to all nodes in B'; specifically, each node in B will store full routes to $2^{O(\alpha)}$ nodes in B' so that exactly one node in B is responsible for each node in B'. Moreover, the nodes in B will maintain a labeled shortest-paths tree rooted at h, such that given $\textsc{id}(t)$ it is possible to route from h to the node v_t that stores a path to t. Here it is crucial that we are free to choose which node in B is responsible for a given node in B'; we'll choose so that for a given link in the shortest-paths tree it suffices to specify a single *range* of target ids for which a packet should take that link.

This is how the packet will reach t. First the node h (which is a neighbor of u) is designated as the intermediate target, and the packet is routed to h via the first-hop pointers. From h the packet is routed to v_t via the shortest-paths tree. Then v_t puts the full route to t into the packet header and send the packet to t. More precisely, v_t will store a $(1+\delta)$-approximate shortest path to t with the smallest hop count, which is at most N_δ by definition of N_δ. Each hop in this path can be encoded by $\lceil \log D_{\text{out}} \rceil$ bits, so the entire path can be stored using at most $N_\delta \lceil \log D_{\text{out}} \rceil$ bits.

This was the second routing mode; it is easy to see that it causes a detour of length at most $O(\delta d_{ut})$. Moreover, we'll show that the *total* path length from source to target is within the claimed stretch $1 + O(\delta)$ even if we switch to the second mode in the middle of a path to some intermediate target. □

4 Searchable small-world networks

In this section we extend Kleinberg's small world networks [10, 11] to metrics of low doubling dimension. While it is relatively straightforward to achieve out-degree $O(\log n)(\log \Delta)$ and $O(\log \Delta)$-hop paths, where Δ is the aspect ratio of the metric, it is quite non-trivial to handle the case of super-polynomial aspect ratio. We obtain $O(\log n)$-hop paths even if Δ is exponential in n. In our first result the out-degree is (still) proportional to $\log(\Delta)$, suggesting that it is a natural lower bound. However, our second result breaks this barrier.

Formally, a *small-world model* on the metric (V, d) consists of two items: a distribution over directed graphs on V and a local search algorithm, with the following two constraints:

- the out-links of a given node u are chosen independently for different u;
- the search algorithm selects the next hop by looking only at distances from the current node to its *neighbors* (out-links), and distances from the current node and its neighbors to the target.

A very natural local search algorithm is the *greedy* algorithm which selects the neighbor that is closest to the target. This is the algorithm used in the seminal Kleinberg's small-world models [10, 11]. Some improvements over Kleinberg's results were obtained in [16, 17] using a version of the greedy algorithm that also looks at neighbors' neighbors. Here we break the $\log(\Delta)$ barrier using the first small-world model with a *non-greedy* local search algorithm. In this search algorithm, if we are at node u and cannot make a sufficient progress towards the target t in one hop, then we jump *sideways*, to a neighbor v that is closest to u (not to t!) subject to the constraint $d_{uv} \leq d_{ut}$.

Theorem 4.1 *Let α be the doubling dimension, and let Δ be the aspect ratio.*

(a) For any metric there is a small-world model with out-degree $2^{O(\alpha)}(\log n)(\log \Delta)$ and a greedy search algorithm such that with high probability all queries complete in $O(\log n)$ hops.

(b) For any metric there is a small-world model with out-degree $2^{O(\alpha)}(\log n)^2(\log \Delta)^{1/2}$ and a local search algorithm such that with high probability all queries complete in $O(\log n)$ hops.

On metrics with bounded growth ratio $B_u(2r)/B_u(r)$ both models coincide with Kleinberg's small world.

Proof Sketch: A relatively straightforward solution is to use $\log \Delta$ rings of neighbors so that the radii of the rings grow exponentially, and the neighbors are distributed with respect to the doubling measure (these are the Y-neighbors in the full proof). It is easy to make sure that the greedy algorithm reduces the distance by at least a factor of two at each step, so the query will take $(\log n)$ steps to complete.

To achieve $O(\log n)$-hop paths, we need the following property. Let $B_{ui} = B_u(r_{ui})$, where $r_{ui} = r_u(2^{-i})$; if the target is node t and for some i we are at a node $u \in B_{(t,i-1)} \setminus B_{ti}$, then we want to get inside B_{ti} in at most a constant number of hops. This is non-trivial when r_{ti} is much smaller than d_{ut}.

In part (a) we can keep the Y-neighbors. It turns out that if throw in another collection of rings of neighbors defined with respect to cardinalities (the X-neighbors in the full proof), then we get inside B_{ti} using only one intermediate hop to some node within distance $d_{ut}/4$ from t.

In part (b), however, storing all Y-neighbors becomes too expensive, so we'll need to prune them (see full proof for details). From part (a) it will follow that as long as we get within distance $d_{ut}/4$ from t, we get inside B_{ti} in at most two hops. However, u might not have a neighbor that is sufficiently close to t. To handle this case, we introduce yet another family of neighbors (called Z-neighbors in the full proof), and we'll need to jump *sideways*, to a neighbor v that is closest to u (not to t!) subject to the constraint $d_{uv} \leq d_{ut}$. We'll see that there will be a Z-neighbor such that the ratio d_{uv}/d_{ut} is sufficiently large, and that from any such neighbor we can get sufficiently close to t. □

Acknowledgments We'd like to thank Jon Kleinberg and Mark Sandler for useful discussions and help with the manuscript.

References

[1] H.T-H. Chan, A. Gupta, B.M. Maggs and S. Zhou, ”On hierarchical routing in bounded-growth metrics,” *SODA* 2005. Full and updated version available as a CMU Tech Report, Dec 2004.

[2] P. Dodds, R. Muhamad and D. Watts, ”An experimental study of search in global social networks,” *Science*, 301: 827-829, 2003.

[3] P. Francis, S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt and L. Zhang, “IDMaps: a global Internet host distance estimation service,” *IEEE/ACM Trans. on Networking*, 2001.

[4] C. Gavoille, D. Peleg, ”Compact and localized distributed data structures,” *J. Distrib. Comput.*, **16**, 2003.

[5] J. Heinonen, *Lectures on analysis on metric spaces*, Springer Verlag, Universitext 2001.

[6] K. Hildrum, J. Kubiatowicz and S. Rao, ”Object location in realistic networks,” *SPAA* 2004.

[7] S. Hotz, ”Routing information organization to support scalable interdomain routing with heterogeneous path requirements,” Ph.D. Thesis, Univ. of Southern California, 1994.

[8] J.D. Guyton, M.F. Schwartz, “Locating nearby copies of replicated Internet servers,” *SIGCOMM* 1995.

[9] A.Gupta, R. Krauthgamer and J.R. Lee, “Bounded geometries, fractals, and low-distortion embeddings,” *FOCS* 2003.

[10] J. Kleinberg, ”The small-world phenomenon: an algorithmic perspective,” *STOC* 2000.

[11] J. Kleinberg, ”Small-world phenomena and the dynamics of information,” *NIPS* 2001.

[12] J. Kleinberg, A. Slivkins and T. Wexler, ”Triangulation and embedding using small sets of beacons,” *FOCS* 2004.

[13] C. Kommareddy, N. Shankar, B. Bhattacharjee, “Finding close friends on the Internet,” *ICNP* 2001.

[14] R. Krauthgamer, J.R. Lee “Navigating nets: Simple algorithms for proximity search,” *SODA* 2004.

[15] R. Krauthgamer, J.R. Lee, M. Mendel and A. Naor, “Measured descent: A new embedding method for finite metrics,” *FOCS* 2004.

[16] G. Manku, M. Naor and U. Wieder, ”Know thy neighbor’s neighbor: the power of lookahead in randomized P2P networks,” *STOC* 2004.

[17] Ch. Martel and V. Nguyen, ”Analyzing Kleinberg’s (and other) small-world models,” *PODC* 2004.

[18] Ch. Martel and V. Nguyen, ”Analyzing and characterizing small-world graphs,” *SODA* 2005.

[19] M. Mendel and S. Har-Peled, ”Fast construction of nets in low dimensional metrics, and their applications,” Manuscript, 2004.

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

[21] A. Slivkins, ”Distributed approaches to triangulation and embedding,” *SODA* 2005.

[22] K. Talwar, ”Bypassing the embedding: approximation schemes and compact representations for growth restricted metrics,” *STOC* 2004.

[23] D. Peleg, ”Distributed computing: a locality-sensitive approach,” *SIAM*, 2000.

[24] C.G. Plaxton, R. Rajaraman and A. W. Richa, ”Accessing nearby copies of replicated objects in a distributed environment,” *Theory Comput. Syst.*, 32(3), pp. 241280, 1999.

[25] B. Wong, A. Slivkins and E.G. Sirer, ”Meridian: A lightweight framework for position-based node selection without virtual coordinates,” Cornell CS dept Tech Report, 2005.

[26] D.J. Watts, S.H. Strogatz, ”Collective dynamics of ’small-world’ networks,” *Nature*, 393:440-42, 1998.

Appendix A: Distance labeling and routing schemes for metrics with large aspect ratio.

In this section we prove Thm. 3.4 on distance labeling, and Thm. 3.5 on routing schemes.

Proof of Thm. 3.4: Keep the notation from the proof of Thm. 3.2. Let X_{ui} and Y_{ui} denote the sets of X_i- and Y_i-neighbors of u, respectively; let $X_u = \cup_i X_{ui}$ and $Y_u = \cup_i Y_{ui}$. Recall that for each $j \in [\log \Delta]$ we fix some 2^j-net $\mathcal{G}_j$. For each node u and each $i \in [\log n]$ fix a node $f_{ui} \in \mathcal{G}_l$, $l = \lfloor \log(r_{ui}/4) \rfloor$, that lies within distance $\delta r_{ui}/4$ from u. Such node lies in Y_{ui} by definition of the Y_i-neighbors; it is possible that $f_{ui} = u$. Call the sequence $\{f_{ui} : i \in [\log n]\}$ a *zooming sequence*, and denote it f_u.

For each node u we define the sets

$$\begin{aligned} Z_{uj} &= B_u(2^j) \cap \mathcal{G}_l, \text{ where } l = \max(0, \lfloor \log(2^j \delta/64) \rfloor). \\ T_u &= X_u \cup Z_u \cup [\cup_{v \in X_u} Z_v], \text{ where } Z_u = \cup_{j=1}^{\log \Delta} Z_{uj}. \end{aligned}$$

The elements of T_u will be called the *virtual neighbors* of u. We'll need the following crucial facts about virtual neighbors:

Claim A.1 *Fix node u and $i \in [\log n]$; let $r = r_{(u,i-1)}$ and $f = f_{(u,i-1)}$. Then*

(a) if $r_{ui} \leq r/12$ then the nearest X_i-neighbor of u is an X_i-neighbor of f.

(b) if $x \in [\frac{\delta}{4} r_{ui};\, 6r_{ui}]$, $x \leq r/2$ then any node $w \in \mathcal{G}_{\lfloor \log x \rfloor} \cap B_u(x)$ is a virtual neighbor of f.

(c) in particular, node f_{ui} is a virtual neighbor of f.

Proof: For part (a), note that the nearest X_i-neighbor of u is some node $h_B \in B \in \mathcal{F}_i$ such that the ball B is a subset of $B_u(6r_{ui})$. Letting $z = r_{(f,i-1)}$ we have $z - d_{uf} \geq r - 2d_{uf} \geq r/2 \geq 6r_{ui}$ and consequently $B_u(6r_{ui}) \subset B_u(z - d_{vf}) \subset B_f(z)$. Part (a) follows by definition of the X_i-neighbors.

For part (b), it is easy to check that if $r_{ui} > r/12$ then $w \in Z_{fj}$ for $j = \lceil \log(x + d_{uf}) \rceil$. Now suppose $r_{ui} \leq r/12$ and let v be the nearest X_i-neighbor of u. Then $d_{uv} \leq 6r_{uv}$, so it is easy to see that $w \in Z_{vj}$ for $j = \lceil \log(d_{uv} + x) \rceil$. Since $w \in Z_{vj}$ and by part (a) $v \in X_{ui}$ it follows that $w \in T_u$.

Finally, part (c) follows from (b) with $x = r_{ui}/4$. □

Let's define the labels of nodes. Fix some enumeration $\varphi_u(\cdot)$ of each set $X_u \cup Y_u$; call it a *host enumeration* of u. Since any ball B_{u0} contains all nodes, the sets X_{u0} conside for all u, and so do the sets Y_{u0}. Therefore without loss of generality all host enumerations coincide on $X_{u0} \cup Y_{u0}$. Fix some enumeration $\psi_u(\cdot)$ of each set T_u; call it a *virtual enumeration* of u.

Fix node u and let $N(i) = X_{ui} \cup Y_{ui}$. Whenever $v \in N(i)$, the label of u will include the translation between the host enumeration of u and the virtual enumeration of v. Specifically, for each $i \in [\log n]$ we define the translation function ζ_{ui} on pairs of integers, so that

$$\zeta_{ui}\left(\varphi_u^{-1}(v),\, \psi_v^{-1}(w)\right) = \varphi_u^{-1}(w) \text{ whenever } v \in N(i) \text{ and } w \in N(i+1) \cap T_v,$$

and *null* otherwise.

The label of u will contain distances to all its neighbors (but not to its virtual neighbors). These distances are stored as an array whose j-th entry is the distance to $\varphi_u(j)$, encoded as a $O(\log \frac{1}{\delta})$-bit mantissa and $O(\log \log \Delta)$-bit exponent. Moreover, the label will contain the maps ζ_{ui}: each ζ_{ui} is represented by an ordered set of triples (x, y, z) such that $\zeta_{ui}(x, y) = z \neq null$. Finally, it will contain the sequence f_{ui}'s; specifically, we store $\varphi_u(f_{u0})$, and each $f_{(u,i+1)}$ is represented by its number in the virtual enumeration of f_{ui}. This completes the definition of the labels.

By the proof of Thm. 3.2, the cardinality of each X_{ui}, Y_{ui} and Z_{uj} is upper-bounded by some $K = [O(1/\delta)]^{O(\alpha)}$. Therefore each node has at most $K \log n$ neighbors, and each map ζ_{ui} is represented by at most K^2 triples, each triple taking at most $2\log K + \log|T_u|$ bits to store. Since $|T_u| < O(K^2)\log(n)\log(\Delta)$, the label size is within the claimed bounds.

Let's show how to estimate distances from the labels. As in the proof of Thm. 3.2, fix a node pair uv, denote $d = d_{uv}$, let $r = (1+\delta)d$ and choose i such that $r_{ui} < r + d \leq r_{(u,i-1)}$. It follows that $r_{(v,i-1)} \geq r$ and $r_{vi} \leq r + 2d$. We know that there exists a node w_0 within distance δd from u or v such that w_0 is a common neighbor of u and v; recall that distances from w_0 to u and v give us a desired estimate. However, we know such w_0 exists, but we don't know how to identify it: this is non-trivial since we do not have global node ids. In our context, to *identify* a common neighbor w of u and v means to find $\varphi_u(w)$ and $\varphi_v(w)$.

Essentially, if w_0 is close to v then we identify it by zeroing in via the sequence of f_{vj}'s, and vice versa. First we need a basic claim about f_{vj}'s:

Claim A.2 *For any $j \geq i-1$ we have $f_{uj} \in Y_{vj}$ and $f_{vj} \in Y_{uj}$.*
Proof: Let $w = f_{vj}$ and note that $w \in \mathcal{G}_l$ for $l = \lfloor \log(r_{vj}/4) \rfloor$. Since $r_{uj} \geq r + d$, by Claim 3.3 it follows that $r_{uj}/2 < r_{vj} < 2r_{uj}$ and $d_{uw} \leq d + r_{vj}/4 \leq 1.5 r_{uj}$. Therefore $w \in Y_{uj}$ by definition of Y_{uj}. Similarly, we can show that $f_{uj} \in Y_{vj}$. □

In particular, for any $j \leq i-1$ nodes f_{uj} and f_{vj} are common neighbors of u and v. Moreover, we can *identify* them sequentially using the translation maps ζ_{uj} and ζ_{vj}. For instance, it is easy to identify f_{u0} since it is numbered the same in any host enumeration. Then, inductively, suppose that we have identified some f_{uj}, $j \leq i-2$ and we need to identify $f = f_{(u,j+1)}$. Then by Claim A.1c f is a virtual neighbor of f_{uj}, so we can find $\varphi_u(f)$ using map ζ_{uj} and (by Claim A.2) we can find $\varphi_v(f)$ using map ζ_{vj}.

Now, assuming w_0 is closer to v than to u, we will identify it using $f = f_{(v,i-1)}$: it suffices to show that w_0 is a virtual neighbor of f. (If w_0 is closer to u, we can identify it similarly using $f_{(u,i-1)}$.) According to the proof of Thm. 3.2, letting $x = \delta d$ we can assume that either

(a) $r_{vi} \leq x/6$ and w_0 is the nearest X_i-neighbor of v, or

(b) $r_{vi} > x/6$ and $w_0 \in \mathcal{G}_l$ such that $l = \lfloor \log x \rfloor$.

In case (a) $w_0 \in T_f$ by Claim A.1a since $r_{(v,i-1)} > d \geq 2x$; in case (b) $w_0 \in T_f$ by Claim A.1b since x matches the conditions in the claim. Theorem proved. □

Proof of Thm. 3.5a: For every node u, let $\mathrm{ID}(u)$ be its unique $\lceil \log n \rceil$-bit ID. Fix a $3/2$-approximate distance labeling scheme guaranteed by Thm. 3.4; for each node u let L_u be the label of u in this scheme, and let $D(\cdot,\cdot)$ be the non-contracting distance function on labels. Without loss of generality assume that L_u contains $\mathrm{ID}(u)$. Each packet header consists of L_t and $\mathrm{ID}(t')$ where t is the target and t' is the *intermediate target*. The routing table of node u contains labels L_v of some nodes v which we call *neighbors* of u; we'll specify them later. For each such v we also store the first node $g_u(v)$ on some shortest path to v.

The routing algorithm is simple. To send a packet to node t, node u initiates the intermediate target to u. Suppose node u creates or receives a packet with target t and intermediate target t'. If $t' = u$ then node u selects a neighbor v such that $D(L_v, L_t)$ is minimal, makes v the new intermediate target, and forwards the packet to $g_u(v)$. Else, as we'll see, t' is a neighbor of u, so node u just forwards the packet to $g_u(t')$.

Let's define the neighbors: for each $j \in [\log \Delta]$ let F_j be some 2^j-net; let $r_j = 2^{j+2}/\delta$ and $F_j(u) = B_u(r_j) \cap F_j$. Elements of $F_j(u)$ are called *j-level neighbors* of u; by Lemma 1.1 each node has at most $[O(\frac{1}{\delta})]^{O(\alpha)}$ j-level neighbors for each j.

Now we can proceed with the proof of correctness. We claim that for any pair ut of nodes, letting $d = d_{ut}$, node u has a neighbor $v \in B = B_t(\delta d)$. Indeed, pick j such that $2^j \leq \delta d < 2^{j+1}$. Then on one hand B contains some node $v \in F_j$, and on the other hand $r_j > 2d$, so $B \subset B_u(r_j)$, so v is a j-level neighbor of u, claim proved. It follows that $D(L_t, L_v) \leq \delta d(1+\delta)$. So when node u selects a new intermediate target for a packet with final target t, it selects a neighbor v within distance $\delta(1+\delta)d$ from t.

Suppose an intermediate target t' for packet P has been set by the node u. Then $t' \in F_j(u)$ for some j. We claim that $t' \in F_j(v)$ for every node v visited by P after u and before reaching t'. Indeed, let's use induction: if $t' \in F_j(v)$ then P goes from v to $w = g_v(t')$, so $d_{wt'} < d_{vt'} \leq r_j$, so $t' \in F_j(w)$, claim proved.

Now Claim 2.4c holds: each packet follows a shortest path to each intermediate target. To reach the i-th intermediate target, $i \geq 1$, the packet traverses path at most $d_{ut}(1+2\delta)\delta^{i-1}$. Therefore the total path length is at most $d_{ut}[1 + O(\delta)]$, theorem proved. □

Proof of Thm. 3.5b: We'll combine the ideas of Thm. 3.4 and Thm. 2.1, and add some new tricks. In particular, we'll use (i) the basic rings of neighbors, (ii) zooming sequences and intermediate targets, (iii) the first-hop pointers, and (iv) host/virtual enumerations. We'll use the rings, the zooming sequences, and the enumerations as defined in Thm. 3.4. In fact, we'll use all definitions from the proof of Thm. 3.4, for the same value of δ, including the sets of X_i- and Y_i-neighbors. We also need a unique $\lceil \log n \rceil$-bit identifier $\text{ID}(u)$ for every node u.

The *routing label* of u represents the sequence f_{ui}'s; specifically, we store $\varphi_u(f_{u0})$, and each $f_{(u,i+1)}$ is represented by its number in the virtual enumeration of f_{ui}. As in Thm. 2.1, packet headers will contain a routing label of the target node t, and, possibly, a *intermediate target* f_{ti} for the appropriately chosen i.

Let L_u be the label of node u in a δ-approximate distance labeling scheme from this theorem. The *routing table* of each node u includes L_u; moreover, as in Thm. 2.1, for every neighbor v of u, the routing table of u includes the first-hop pointer from u to v, which we can store using only $\lceil \log D_{\text{out}} \rceil$ bits.

The routing will have two *modes*. One is essentially from Thm. 2.1: if the intermediate target is not set, consider the smallest i such that the element $f = f_{ti}$ of zooming sequence f_t is sufficiently close to the target t. If it is possible to *identify* this f (i.e. to find $\varphi_u(f)$, where u is the current node), then we designate f as the intermediate target, put i and $\text{ID}(t)$ in the packet header, and try to route to f using the first-hop pointers. However, this routing algorithm might fail, since it might not be possible to identify f at u or at some intermediate node. The good news is that this can happen only if for the current node u and some i such that $r_{ui} < 1.25 d_{ut} \leq r_{(u,i-1)}$ it is the case that r_{ui} is much smaller than d_{ut}, specifically $6r_{ui} < \delta d_{ut}$. (This can be shown using the techniques from Thm. 3.4 and Thm. 2.1; we omit the details from this version of the paper.) In this case we'll go into the *second routing mode*, which we describe below.

By Lemma 3.1 there exists a ball $B \in \mathcal{F}_i$ of cardinality at least $n/2^{i+O(\alpha)}$ such that $B \subset B_u(6r_{ui})$. Let $h = h_B$ be the node selected from B in Thm. 3.2 (e.g. without loss of generality it could be the center of B). It is easy to see that the ball $B' = B_{(h,i-1)}$ contains target t. The nodes in B will collectively store the routes to all nodes in B'; specifically, each node in B will store full routes to $2^{O(\alpha)}$ nodes in B'. Moreover, the nodes in B will maintain a shortest-paths tree T_B rooted at h. We label the edges of T_B so that given $\text{ID}(t)$, $t \in B'$ it is possible to route from h to the node $v_t \in B$ that stores a path to this $t \in B'$. Specifically, we label each edge $uv \in T_B$ with a range such that if a packet is at u and $\text{ID}(t)$ is within this range, then the packet is forwarded to v.

It is crucial that we are free to assign the ranges to edges of T_B, and to choose the mapping from B' to B (i.e. to choose v_t for each $t \in B'$). We do it using a top-to-bottom construction on the tree T_B. For technical convenience, extend T_B as follows: for every node $u \in B$ add a distinct node l_u and edge (u, l_u), so that each node has a corresponding leaf. We start from the root which is assigned the full range $[\log n]$. For a node $u \in B$ with a given range, partition this range into subranges R_v, $uv \in T_B$ such that $|R_v|$ is proportional to the cardinality of the subtree of T_B rooted at v. For each leaf $l = l_u$, we assign to u all nodes t such that $\text{ID}(t) \in R_l$.

This is how the packet will reach target t. First the node h (which is a neighbor of u) is designated as the intermediate target, and the packet is routed to h via the first-hop pointers. From h the packet is routed to v_t via the shortest-paths tree. Then v_t puts the full route to t into the packet header and send the packet to

t. More precisely, v_t will store a $(1+\delta)$-approximate shortest path to t with the smallest hop count, which is at most N_δ by definition of N_δ. Each hop in this path can be encoded by $\lceil \log D_{\text{out}} \rceil$ bits, so the entire path can be stored using at most $N_\delta \lceil \log D_{\text{out}} \rceil$ bits.

This was the second routing mode; it is easy to see that it causes a detour of length at most $O(\delta d_{wt})$, where w is the node where we switch to this mode. Moreover, we claim that the *total* path length is within the claimed stretch $1 + O(\delta)$; this is even if we switch to this mode in the middle of a path to some intermediate target v. Indeed, suppose the packet originated at node u and let $d = d_{ut}$. Let ρ_{xy} be the distance traversed by the packet on its path from node x to node y. Recall that $v \in B_t(\delta d)$. It is easy to see that the first-mode routing satisfies the nice properties in Claim 2.4ac, with a similar proof; in particular, node w lies on some shortest path from u to v, and the packet followed this path from u to w. Therefore,

$$\begin{aligned}
\rho_{wt} &\leq (1+O(\delta))d_{wt} \leq (1+O(\delta))(d_{wv}+d_{vt}) \leq d_{wv} + O(\delta d) \\
\rho_{ut} &\leq \rho_{uw} + \rho_{wt} = d_{uw} + d_{wv} + O(\delta d) = d_{uv} + O(\delta d) = d + O(\delta d),
\end{aligned}$$

as claimed. □

Appendix B: Searchable small worlds.

Proof of Thm. 4.1: Let $B_{ui} = B_u(r_{ui})$, where $r_{ui} = r_u(2^{-i})$; fix an absolute constant c to be specified later. Let μ be the $2^{O(\alpha)}$-doubling measure. For each $i \in [\log n]$ select a node independently and uniformly at random from the ball B_{ui}; repeat this $c \log n$ times, and let X_{ui} be the set of selected nodes. Let $X_u = \cup X_{ui}$.

(a) For a given node u, select the neighbors as follows. For each $j \in [\log \Delta]$ select a node independently from the ball $B = B_u(2^j)$ according to the probability distribution $\mu(\cdot)/\mu(B)$; repeat this $(2^{c\alpha} \log n)$ times, and let Y_{uj} be the set of selected nodes. Let $Y_u = \cup Y_{uj}$, and define the set of neighbors of u as $X_u \cup Y_u$. Since $\log(\Delta) \geq \frac{1}{\alpha}(\log n)$ the out-degree of u is within the claimed bound.

We need to prove that the greedy search algorithm always finds a path of $O(\log n)$ hops. We'd like to show that if the target is node t and for some i we are at a node $u \in B_{(t,i-1)} \setminus B_{ti}$, then we'll be inside B_{ti} in at most a constant number of hops; in this case we'll show that *two* hops suffices. Indeed, let $d = d_{ut}$ and note that for $j = \lceil \log(1.25d) \rceil$ the set Y_{uj} contains a node $w \in B_t(d/4)$.[6] If $r_{ti} \geq d/4$ then we are done. Now suppose $r_{ti} < d/4$. Since $r_{(t,i-1)} > d$, it follows that $r_{(w,i-1)} > \frac{3}{4}d$, so $B_{ti} \subset B_t(d/4) \subset B_{(w,i-1)}$. Therefore with high probability $X_{(x,i-1)}$ contains a node in B_{ti}, and we are done.

(b) For a given node u, select the neighbors as follows. Let $x = \sqrt{\log \Delta}$ and $\rho_j = 2^{(1+1/x)^j}$. For each j such that $\rho_j \leq \Delta$ pick a node z_{uj} uniformly at random from $B_u(\rho_j) \setminus B_u(\rho_{j-1})$, provided that this set is not empty; else let z_{uj} be the closest node to u that lies outside $B_u(\rho_j)$. Let $Z_u = \cup_j z_{uj}$. For each $i \in [\log n]$ and each integer j such that $|j| \leq (3x+3)\log\log\Delta$ and $r_{(u,i+1)} < r_{ui} \cdot 2^j < r_{(u,i-1)}$ select a node independently from the ball $B = B_u(r_{ui} \cdot 2^j)$ according to the distribution $\mu(\cdot)/\mu(B)$; repeat this $(2^{c\alpha} \log n)$ times and let Y_{uij} be the set of selected nodes. Let $Y_u = \cup_{ij} Y_{uij}$, and define the set of neighbors of u as $X_u \cup Y_u \cup Z_u$.

The local search algorithm is simple. Suppose we are at node u and t is the target. If u has a neighbor within distance $d_{ut}/4$ from t then greedily choose the neighbor that is closest to t. Else we do a non-greedy

[6] More precisely, by Chernoff Bounds for large enough constant c with high probability this happens for all ut pairs simultaneously. In the rest of the proof we will omit these trivial applications of Chernoff Bounds.

step: we go to a neighbor whose distance to u is most similar to d_{ut}; specifically, we go to a neighbor w that is closest to u subject to the constraint $d_{uw} \geq d_{ut}$.

This completes the specification of our small-world model; now we need to prove that our search algorithm indeed finds short paths. We'd like to show that if the target is node t and for some i we are at a node $u \in B_{(t,i-1)} \setminus B_{ti}$, then we'll be inside B_{ti} in at most a constant number of hops; in this case we'll show that *three* hops suffices. Indeed, let $d = d_{ut}$ and note that as proved in part (a), if we get within distance $d/4$ from t then in at most one more hop we are done. Suppose u does not have a neighbor in $B_t(d/4)$ and choose l such that $r_{ul} \leq d \leq r_{(u,l-1)}$. It is easy to see that $r_{ul} \cdot 8^{x+1} < 1.25d < r_{(u,l-1)}/8^{x+1}$. Indeed, if the first inequality fails then for $j = \lfloor \log(d/r_{ul}) \rfloor$ some node from Y_{ulj} lies in $B_t(d/4)$, contradiction; if the second inequality fails, similar contradiction arises.

Now choose j such that $\rho_j \leq d < \rho_{j+1}$ and consider $z = z_{uj}$. It follows that $\rho_{j-1} \leq d_{uz} \leq d$ and

$$d/d_{uz} \leq \rho_{j+1}/\rho_{j-1} = (\rho_{j-1})^{3/x} \leq \Delta^{3/x} = 8^x.$$

Therefore our search algorithm will choose some neighbor z such that $d/8^x \leq d_{uz} \leq d$. In particular, it follows that $4r_{ul} < d_{uz} < r_{(u,l-1)}/4$.

Now that we are at z we will be able to make progress towards t. To be consistent with our search algorithm, the next hop should get us from z to within distance $d_{tz}/4$ from t. So, since by our assumption $w \notin B_t(d/4)$, it suffices to get inside $B_t(d/16)$, and then in at most one more hop we are done. (Note that if the search algorithm is allowed to remember the previous move, then getting inside $B_u(d/4)$ is sufficient, too.) We will achieve the desired progress using some neighbor in Y_{zlj} for the appropriately chosen j.

We claim that $d_{uz} - r_{ul} \leq r_{zl} \leq d_{uz} + r_{ul}$. Indeed, the second inequality follows since the ball $B_z(d_{uz} + r_{ul})$ contains B_{ul} and therefore has cardinality at least $n/2^l$. If the first inequality fails then the balls B_{zl} and B_{ul} are disjoint; since both balls lie inside $B_u(d_{uz} + r_{zl})$, the latter ball has cardinality at least $n/2^{l-1}$, which contradicts the definition of $r_{(u,l-1)}$ since $d_{uz} + r_{zl} \leq 2d_{uz} + r_{ul} < r_{(u,l-1)}$. Claim proved.

It follows that $r_{zl}/d_{uz} < (\frac{4}{3}, \frac{5}{3})$. Therefore

$$r := d_{zt} + d/16 \leq 1.07d + d_{uz} \leq d_{uz}(1.07 + 8^x) < 2^{3x+1} r_{zl}$$

Choose j such that $2^{j-1} < r/r_{zl} \leq 2^j$; obviously, $j \leq 3x + 2$ and $B_t(d/16) \subset B_{zlj}$. The radii of these two balls are within a constant factor since $d_{zt} \leq d + d_{uz} \leq 2d$. Therefore it follows that with high probability the ball $B_t(d/16)$ contains a node from Y_{zlj}.

$\square$