Distance Estimation and Object Location via Rings of Neighbors[*]

Aleksandrs Slivkins[†]

February 2005
Revised: June 2005, Nov 2005, April 2006, Sept 2006

Abstract

We consider four problems on distance estimation and object location which share the common flavor of capturing global information via informative node labels: low-stretch routing schemes [47], distance labeling [24], searchable small worlds [30], and triangulation-based distance estimation [33]. Focusing on metrics of low doubling dimension, we approach these problems with a common technique called *rings of neighbors*, which refers to a sparse distributed data structure that underlies all our constructions. Apart from improving the previously known bounds for these problems, our contributions include extending Kleinberg's small world model to doubling metrics, and a short proof of the main result in Chan et al. [14]. Doubling dimension is a notion of dimensionality for general metrics that has recently become a useful algorithmic concept in the theoretical computer science literature.

1 Introduction

In *node labeling* problems one needs to assign short labels to nodes of a graph so that they capture some (problem-specific) global information about distances and routes in the graph. We consider four problems of this type: low-stretch routing schemes [47], distance labeling [24], searchable small worlds [30], and triangulation-based distance estimation [33].

We approach these problems with a common technique called *rings of neighbors*, which refers to a sparse distributed data structure that underlies all our constructions. The idea is that every node u stores pointers to (i.e. addresses of) 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 selection of neighbors depend on the specific application. For a simple example, consider the structure where each ball B_i has radius 2^i, and the neighbors in the i-th ring are selected independently and uniformly at random in B_i. In effect, rings of neighbors form an overlay network with a certain structure imposed by the balls $\{B_i\}$.[1]

For the problems that we consider, the input is a finite metric space or, more generally, an undirected weighted graph that induces a shortest-paths metric. We focus on metrics of low *doubling dimension*, a notion of low dimensionality for general metrics that has recently become a useful algorithmic concept in the theoretical computer science literature [25, 36, 35, 50, 33, 49, 14] in many different contexts, including metric embeddings, traveling salesman and compact data structures; in particular, in [33, 49] it was used

[*] Preliminary version of this paper has appeared in *24th Annual ACM SIGACT-SIGOPS Symp. on Principles Of Distributed Computing (PODC)*, 2005. This is the full version. The journal version (which excludes Appendix B) will be published in the special issue of *Distributed Computing*.

[†] Department of Computer Science, Brown University, Providence RI, 02906. *slivkins at cs.brown.edu*. This work was done when A. Slivkins was a graduate student at Cornell University and was supported by the Packard Fellowship of Jon Kleinberg.

[1] Note that the term 'neighbor' here refers to the adjacency in this overlay network, not to the proximity in the input graph.

to model the structural properties of the Internet distance matrix in the context of distributed algorithms for metric embedding and distance estimation.

Any point set in a k-dimensional ℓ_p metric has the following property [10]: for $\alpha = k + O(1)$, every set of diameter d can be covered by 2^α sets of diameter $d/2$. (The diameter of a set is the supremum of distances between any two points in this set.) This motivates the following definition: *doubling dimension* of a metric space is the infimum of all α such that the above property holds. Clearly, doubling dimension of any n-node metric is at most $\log n$. *Doubling metrics* are defined as metrics such that the doubling dimension is upper-bounded by a constant.

By definition, doubling metrics generalize constant-dimensional ℓ_p metrics. Doubling metrics is a much wider class of metrics: in particular, there exist doubling metrics on n nodes that need distortion $\Omega(\sqrt{\log n})$ to embed into any ℓ_p, $p \geq 2$ [25]. Moreover, doubling metrics subsume metrics of bounded *grid dimension*[2], which have been considered in the long line of work on Distributed Hash Tables started by Plaxton et al. [48] (see the intro of [27] for a short survey). Again, doubling metrics is a much wider class of metrics: as an example of a doubling metric with high (super-constant) grid dimension, consider the set $\{1, 2, 4, \ldots, 2^n\}$ equipped with the standard distance function $d(x, y) = |x - y|$. Furthermore, unlike grid dimension, the doubling dimension is robust, in the sense that the dimension of a subset is no larger than the dimension of the entire metric.

Now let us discuss each of the four problems in more detail.

Low-stretch routing schemes. A *routing scheme* on a network is a distributed algorithm that provides routing of packets from any node to any other node. The underlying connectivity of the network is expressed by a weighted graph, where weights represent delays on edges. Every node u is assigned a *routing label* and a *routing table*. All routing decisions are local in the sense that they are based on the routing table and the packet header, which includes the label of a target node.

Formally, a routing scheme on a family $\mathcal{G}$ of graphs consists of the following components:

(a) for each $G \in \mathcal{G}$, an assignment of routing labels and routing tables to the nodes of G;

(b) an algorithm that inputs a routing table of the current node, and a packet header, and outputs the next hop for this packet; the next hop must be an edge in G, adjacent to the current node.

(c) an algorithm that inputs the routing table of node u and the routing label of some other node v, and outputs the packet header such that the packet reaches v starting from u.

The algorithms in (b) and (c) must be polynomial-time computable (with respect to the input length). By a slight abuse of notation, we can talk about a routing scheme on a particular graph $G \in \mathcal{G}$ once the underlying family $\mathcal{G}$ of graphs is clear. Such routing scheme consists of routing labels, routing tables, and the algorithms in (b) and (c). [3]

Let d_{uv} be the length of the shortest uv-path in G. Say a uv-path has *stretch* β if its d-length is at most βd_{uv}. A routing scheme on G has stretch β if for any source-target pair the packet follows a β-stretch path. For a given stretch we try to minimize two parameters: storage (the maximal size of a routing table), and communication (the maximal size of a packet header).

In a trivial stretch-1 routing scheme, each node stores full routing table of the all-pairs shortest paths algorithm. However, this routing table takes up $\Omega(n \log n)$ bits, which does not scale well with n. *Compact* low-stretch routing schemes have been introduced in Peleg and Upfal [47], and explored in a number of

[2] The *grid dimension* of a metric is the smallest α such that for any ball B the cardinality of B is at most 2^α times the cardinality of a ball with the same center and half the radius. Note that a k-dimensional grid has this property for $\alpha = k + O(1)$.

[3] A technicality: these algorithms must be the same for all graphs in $\mathcal{G}$, so that one could not encode all of G inside the algorithm.

	routing table size, bits	packet header size, bits
Talwar [50]	$O(\frac{1}{\alpha\delta})^{\alpha}(\log^{2+\alpha}\Delta)$	$O(\alpha\log\Delta)$
Chan et al. [14]	$(\frac{\alpha}{\delta})^{O(\alpha)}(\log\Delta)(\log D_{\text{out}})$	$O(\alpha\log\frac{1}{\delta})(\log\Delta)$
Theorem 2.1	$(\frac{1}{\delta})^{O(\alpha)}(\log\Delta)(\log D_{\text{out}})$	same as above
Theorem 4.1	$(\frac{1}{\delta})^{O(\alpha)}(\log\Delta)(\log n)(\log\log\Delta)$	$2^{O(\alpha)}(\log n)\log(\frac{1}{\delta}\log\Delta)$
Follow-up work [3]	$(\frac{1}{\delta})^{O(\alpha)}(\log\Delta)(\log n)$	$\lceil\log n\rceil$
	$(\frac{1}{\delta})^{O(\alpha)}(\log^4 n)$	$2^{O(\alpha)}(\log^3 n)$

doubling dimension α, aspect ratio Δ, out-degree D_{out}

Table 1: $(1+\delta)$-stretch routing schemes for doubling graphs.

subsequent papers (see [23, 46] for a survey). In particular, for any integer $k \geq 2$ there exists a $(4k-5)$-stretch routing scheme on weighted graphs with $o(k\log^2 n)$-bit packet headers and $\tilde{O}(n^{1/k})$-bit routing tables [52, 51]; this trade-off between the stretch and the size of routing tables is essentially optimal [47]. Moreover, there is no routing scheme on weighted graphs with stretch less than 3 and $o(n)$-bit routing tables [21].

OUR CONTRIBUTIONS: We focus on routing schemes for weighted undirected graphs that induce doubling metrics (for simplicity, let's call them *doubling graphs*). In this setting Talwar [50] has achieved compact $(1+\delta)$-stretch routing schemes, for any given $\delta > 0$; the routing table size in his result has been improved by Chan et al. [14]. Using rings of neighbors, we re-derive the result in [14] via the construction and proof of correctness that are significantly shorter and simpler than the ones in [14]; our guarantees (Theorem 2.1) are slightly improved, too. Moreover, we can give a *really* simple derivation (Theorem 4.1) if we use our result on distance labeling and allow an extra $(\log n)$ factor in the routing table size. The quantitative results are summarized in Table 1. All these results extend to a related model of *routing schemes on metrics*,[4] with poly-logarithmic out-degrees; see Section 4.1 for more details.

We note that the above guarantees are unsatisfactory if the aspect ratio Δ (the largest distance divided by the smallest distance) is very large, e.g. $\Delta = 2^n$. We wish to alleviate the dependency on Δ; we do it by replacing the $(\log\Delta)$ factor with $(\log n)(\log\log\Delta)$. The first step in this direction is Theorem 4.1, where the improvement is for packet headers only. Furthermore, in Section 4 we improve both packet headers and routing tables for routing schemes on metrics, and also (Theorem 4.2) for routing schemes on weighted graphs that contain near-shortest paths with small hop-counts; the latter property is, intuitively, a natural property of a ”good” network topology.

FOLLOW-UP WORK. Following the publication of the conference version of this paper, and building on our techniques, Abraham et al. [3] further alleviate the dependency on the aspect ratio Δ for routing schemes on graphs (see Table 1). In particular, one of their results essentially improves the packet header size in Theorem 4.1 to $\lceil\log n\rceil$. They also provide an extension where they get rid of the dependence on Δ altogether, at the cost of extra poly-log(n) factors in both routing table size and packet header size. This result elaborates on our Theorem 4.2, eliminating the requirement of near-shortest paths with small hop-counts. Abraham et al. [3] also refine our results on doubling metrics, see Section 4 for further details.

Related work on routing schemes. An important version of routing schemes is *name-independent routing* [11, 13], where the routing destination is specified only in terms of its unique $\lceil\log n\rceil$-bit identifier that

[4] A routing scheme on a metric (V, d) is a routing scheme on a directed graph on $G = (V, E)$ such that for any edge $uv \in E$, the weight of this edge is d_{uv}. The crucial point here is that we are free to choose the edge set E (which is, essentially, an overlay network). The out-degree of E becomes another parameter to be optimized.

is given as an input and cannot be changed by our construction. Currently the best known results for arbitrary weighted graphs are: stretch $O(k)$ with $\tilde{O}(n^{1/k}\log\Delta)$-bit tables [4], and stretch 3 with $\tilde{O}(\sqrt{n})$-bit tables [6]; both routing schemes use poly-log packet headers.

For weighted graphs that induce doubling metrics, the extra restriction of name-independence results in more demanding storage requirements: $(1+\delta)$-stretch routing with $o(n)$-bit routing tables is no longer possible for any $\delta < 2$ [3]. However, there *is* a routing scheme with $O(1)$-stretch and polylog storage/headers [3]. Moreover, for any $\delta > 0$ there exists a $(1+\delta)$-stretch routing scheme on low-dimensional *Euclidean* metrics [7], also with polylog storage and headers, which is 'almost' name-independent (node labels include Euclidean coordinates).

A number of results on name-independent routing has focused on the case of bounded grid dimension (see the intro to [27] for a short survey). The best current results [9, 8] achieve $(1+\delta)$-stretch with poly-log storage/headers for routing on metrics and on graphs, respectively.

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', has been discovered in a seminal work of Milgram [44] and recently confirmed by Dodds et al. [15]. Motivated by Watts and Strogatz [54], Kleinberg [30, 29] has articulated another striking aspect of 'small worlds': that a greedy routing algorithm can find short paths to most targets using only local information. Kleinberg went on to suggest several mathematical models where this happens [30, 31]. In particular, he considered a constant-dimensional grid and proved that if every node chooses a constant number of long-range contacts from a fairly natural probability distribution, then in expectation a greedy routing algorithm finds $O(\log^2 n)$-hop paths for every query. The follow-up work (e.g. [37, 41, 40, 18, 42]) has focused on small worlds on hierarchies and grid-like graphs, with versions of the basic greedy routing from [30]. This line of work has also found applications in the design of peer-to-peer systems (e.g. [39]). For more background on small-world networks, refer to a very recent survey by Kleinberg [32].

The following design space emerges. We are given a notion of distance such that every node can locally compute its distance to any given node (e.g. we may assume that node names include informative labels that enable such computation). For this distance function, we need to provide an overlay network of long-range contacts, and specify a routing algorithm which finds short paths to every target using only local information about the contacts. The long-range contacts are usually given as a probability distribution which has the following informal property: if from the point of view of a given node u two nodes v and w are similar, then these two nodes should have a similar probability of being chosen as contacts of u. We would like to minimize the number of long-range contacts (i.e. the out-degree), and the path length.

Most of the previous work has considered the distance induced by a given (possibly directed) unweighted graph of short-range contacts; note that one could start from this notion of distance and recover the short-range contacts as all nodes within distance 1. Abstracting away the useful small-world properties of grids and hierarchies, Kleinberg [31] introduced searchable small worlds on distance functions induced by certain families of node sets. Here we take a somewhat different (and perhaps more basic) approach: we consider distance functions that are metrics, and we wish to extend Kleinberg's small worlds beyond those induced by hierarchies and grid-like graphs. Namely, we extend them to doubling metrics.

We use routing algorithms such that the next hop is chosen by only looking at the current node's contacts, which is a desirable property since (intuitively) this is the minimal amount of information a routing algorithm can be allowed to use. More formally, the next hop is chosen among the current node's contacts, by looking *only* at distances to these contacts and distances from these contacts to the target. Let us call such routing algorithms *strongly local*. The *greedy* algorithm used in [30] is a strongly local routing algorithm that just chooses the contact that is closest to the target.

OUR CONTRIBUTIONS: We extend Kleinberg's model to doubling metrics. 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, it is quite non-trivial to handle the case of super-polynomial Δ. To remedy this, 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 since we need some long-range contacts for each one of the $(\log \Delta)$ distance scales. 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 routing 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* strongly local routing algorithm.

We note in passing that our results trivially extend to a setting where we are given a graph of local contacts, and we add exactly one long-range contact per node; see Section 5.3 for further discussion.

Related work on small-world networks. In the literature on searchable small-world networks several non-greedy routing algorithms have been suggested. In Manku at al. [40] the algorithm looks at all contacts of contacts of the current node, and (greedily) forwards the message to one that is closest to the target. In Martel and Nguyen [41] and Fraigniaud et al. [18] the algorithm looks at several nodes that are closest to the current node u, looks at their contacts, among these contacts chooses one (let us call it v) that is closest to the target, and tries to deliver the message to v by forwarding it to one of the contacts of u. Finally, in Lebhar and Schabanel [37] the algorithm has access to contacts of the previously visited nodes. Note that all these non-greedy algorithms are *not* strongly local.

Following the publication of the conference version of this paper, we became aware that concurrently with our work, two other papers have independently considered extending searchable small worlds to broader classes of graphs. Specifically, Duchon et al. [16] consider graphs of low grid dimension, and Fraigniaud [17] work on graphs of bounded treewidth. An even more recent paper [2] considers weighted minor-excluding graphs. Furthermore, Fraigniaud et al. [19] have recently provided a complementary impossibility result for searchable small-worlds on an infinite family of graphs of large doubling dimension. [5]

Distance labeling. In a distance labeling scheme (DLS), 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 k-approximate DLS 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 k of the true uv-distance. By a slight abuse of notation, we can talk about a DLS on a particular metric $M \in \mathcal{M}$ once the underlying family $\mathcal{M}$ of metrics is clear. Given k, we would like to minimize the maximal bit-length of node labels.

In a trivial DLS, the label of node u would encode the distances to all other nodes, taking up $O(n \log \Delta)$ bits. Exact DLS are known for two families of unweighted graphs: for bounded-genus graphs and for graphs with constant-size separators, with $\tilde{\mathrm{O}}(\sqrt{n})$- and $O(\log^2 n)$-bit labels, respectively [24]. For weighted graphs, approximate DLS with sublinear label length have been introduced by Peleg [45], see [23, 46] for a survey. In particular, for any integer k there exists a $(2k-1)$-approximate DLS on weighted graphs with $\tilde{\mathrm{O}}(n^{1/k} \log \Delta)$-bit labels [52]; a complementary lower bound of $\tilde{\Omega}(n^{1/k})$ is given in [52, 22].

Major improvements are possible for doubling metrics. For any $\delta \in (0, \frac{1}{2})$ Gupta et al. [25] provided an embedding into ℓ_∞ which translates into a $(1+\delta)$-approximate DLS with $(\frac{1}{\delta})^{O(\alpha)}(\log n)(\log \Delta)$ bits per label, where α is the doubling dimension and Δ is the aspect ratio. Using a different technique, Talwar [50] improved this by a factor of $(\log n)$, and gave a lower bound of $(\frac{1}{\delta})^{\Omega(\alpha)}$. Slivkins [49] observed that since the aspect ratio Δ can be arbitrarily large with respect to n, it is desirable to alleviate the dependency on Δ;

[5] The cited result is for the 'one long-range contact per node' setting; note that it trivially extends to our setting, too.

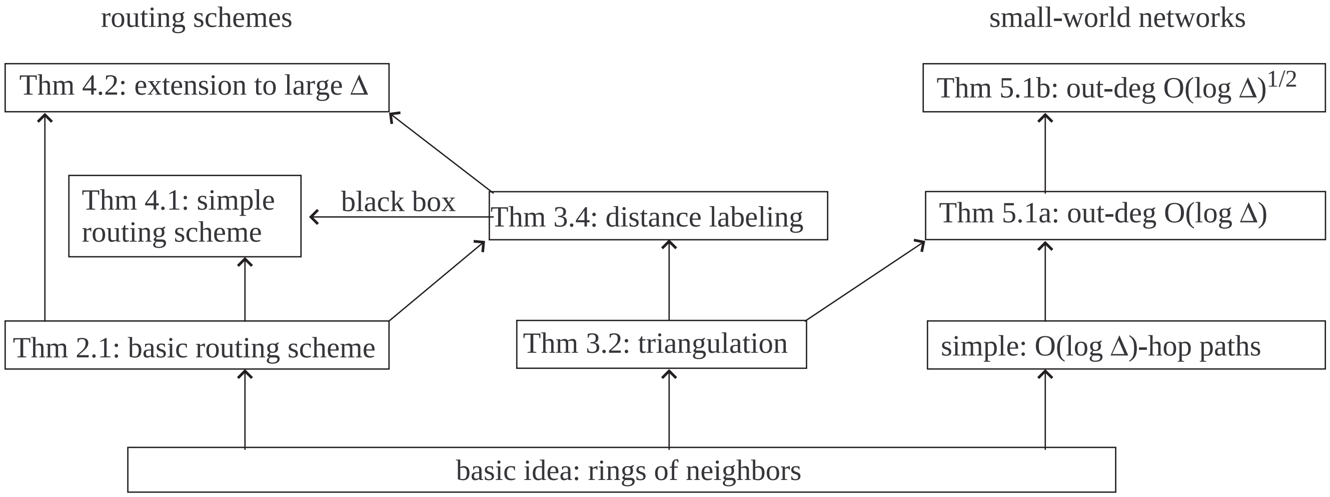
routing schemes
small-world networks
Thm 4.2: extension to large Δ
Thm 5.1b: out-deg O(log Δ)^{1/2}
Thm 4.1: simple routing scheme
black box
Thm 3.4: distance labeling
Thm 5.1a: out-deg O(log Δ)
Thm 2.1: basic routing scheme
Thm 3.2: triangulation
simple: O(log Δ)-hop paths
basic idea: rings of neighbors

Figure 1: Relations between our results: arrows indicate the flow of ideas.

he gave a construction with $(\frac{1}{\delta})^{O(\alpha)}(\log^2 n)(\log n + \log\log\Delta)$ bits per label.[6] This has been improved by a factor of $(\log n)$ in Mendel and Har-Peled [43]; using a construction from [24], for any $\delta \leq .9$ and any $\Delta \geq n^{\log n}$ they derived a lower bound of $\Omega(\log n)(\log\log\Delta)$ bits per label.[7]

OUR CONTRIBUTIONS: We obtain the result in [43] as a simple corollary of our result on triangulation. We improve it to $(\frac{1}{\delta})^{O(\alpha)}(\log n)(\log\log\Delta)$ bits per label using the ideas from our first result on routing schemes. For any $\Delta \geq n^{\log n}$ and bounded α, δ this is optimal up to constant factors.

Triangulation. Motivated by systems for estimating Internet latencies via the triangle inequality [28, 26, 34, 20], Kleinberg et al. [33] introduced the notion of *triangulation* on a metric. Triangulation [33, 49] 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 *beacon 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.

Distributed algorithms for constructing low-order (ϵ, δ)-triangulations on doubling metrics have been developed in [33, 49]; in these triangulations all nodes have the same beacon set. An obvious flaw in these results is that they provide no guarantees for a significant fraction of node pairs. Accordingly, Slivkins [49] considered $(0, \delta)$-triangulations and gave a construction (with distinct beacon sets) that achieves order $(\frac{1}{\delta})^{O(\alpha)}(\log^2 n)$, where α is the doubling dimension.

OUR CONTRIBUTION: We construct a $(0, \delta)$-triangulation of order $(\frac{1}{\delta})^{O(\alpha)}(\log n)$. Using the upper bound D^+ as a distance estimate, we recover the result in [43] on distance labeling.

[6] The conference version of [49] erroneously claimed $(\frac{1}{\delta})^{O(\alpha)}(\log^2 n)(\log\log\Delta)$-bit labels.

[7] This is what they actually prove, although they only claim their result for any $\Delta \geq 4^n$.

The unifying technique. In this paper we present results on four related, yet different problems. These results are unified by a common technique: rings of neighbors. Moreover, these results are intertwined, in the sense that one result elaborates ideas pioneered in another. This flow of ideas is represented in Figure 1. Note that both Theorem 4.1 and Theorem 4.2 build on Theorem 3.4; however, Theorem 4.1 just uses it as a black box, whereas Theorem 4.2 imports its techniques and elaborates on them. In fact, the proof of Theorem 4.2 is the culmination of our techniques for routing schemes, triangulation and distance labeling.

Recall that in *rings of neighbors*, the i-ring neighbors of a given node u lie in a ball B_i around u, for some increasing sequence of balls $\{B_i\}$; the radii of these balls and the distribution of neighbors in a given ring depend on the specific application. One trick that has been particularly useful in our proofs is to combine the following two collections of rings of neighbors. In the first collection, the cardinalities of the balls B_i grow exponentially, and the i-ring neighbors are distributed uniformly on the node set of B_i. In the second collection, the *radii* of the B_i's grow exponentially, and (if one draws on the analogy between doubling metrics and low-dimensional Euclidean metrics) the i-ring neighbors are distributed uniformly in the *space region* that corresponds to B_i. For some applications, e.g. in Section 2, the second collection alone suffices.

In a more abstract view, a collection of rings of neighbors is a tractable representation for the fine structure of the underlying graph. The idea of using a tractable structure-preserving representation as a unifying technique for various problems on graphs is not new; several representations have been suggested in the literature, e.g. [12, 11] for general graphs and [50, 43] for doubling graphs. Our representation seems to be particularly suitable to the problems that we consider in this paper.

Roadmap of the paper. We start with a simple proof of the main result in Chan et al. [14] on routing schemes. In Section 3 we present our results on triangulation and distance labeling. We return to routing schemes in Section 4; in particular, we consider routing schemes on metrics in Section 4.1. In Section 5 we discuss our results on searchable small-world networks. We conclude and discuss open questions in Section 6.

1.1 Preliminaries

Recall that the defining property of a doubling metric is that any set of diameter d can be covered by a constant number of sets of diameter at most $d/2$. We will use this property via a more concrete corollary where we cover with a constant number of *balls*:

Lemma 1.1. *In a metric of doubling dimension α, any set of diameter d can be covered by $2^{\alpha k}$ balls of radius $d/2^k$, for any integer $k \geq 1$. The desired cover can be efficiently constructed.*

Proof. Let α be the doubling dimension. Consider a set S of diameter d and apply the definition of the doubling dimension recursively k times. It follows that S can be covered by $2^{\alpha k}$ sets of diameter at most $d/2^k$. Pick any one point from each of these sets. Then S can be covered with $2^{\alpha k}$ balls of radius $d/2^k$ centered in the selected points. Moreover, it follows that the desired cover can be efficiently constructed by a simple greedy algorithm: select any node $u \in S$, add the ball around u to the cover, delete from S all nodes within distance $d/2^k$ from u, repeat until S is empty. □

Throughout the paper, we denote the underlying metric by d, so that d_{uv} is the distance between nodes u and v. 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 $S \rightarrow [k]$, 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 largest distance divided by the smallest distance. Note that Δ can be arbitrarily large with respect to n and α. For instance, consider a 3-node metric space $\{1, 2, \Delta\}$, equipped with the natural distance function $d(x, y) = |x - y|$. However, it is easy to bound the aspect ratio from below:

Lemma 1.2. $1 + \log \Delta \geq \frac{1}{\alpha} \log n$, *for any metric with aspect ratio* Δ *and doubling dimension* α.

Proof. For simplicity let us divide all distances by the smallest distance. Then the smallest distance is 1, and the diameter is Δ. Recursively applying the definition of the doubling dimension k times, it follows that we can cover the metric with $2^{\alpha k}$ sets of diameter at most $\Delta/2^k$. Taking $k = 1 + \lfloor \log \Delta \rfloor$, we can cover the metric with $2^{\alpha k}$ sets of diameter less than 1. Each of these balls contains at most one node, so $2^{\alpha k} \geq n$. □

Say a measure is s-*doubling* if for any ball $B_u(r)$ its measure is at most s times larger than that of $B_u(r/2)$. Intuitively, a doubling measure μ is an assignment of weights to nodes that makes a metric look growth-constrained; in particular, for the n-node *exponential line*, a one-dimensional set $\{2^i : i \in [n]\}$, we have $\mu(2^i) = 2^{i-n}$. For any finite doubling metric, a doubling measure exists and can be constructed efficiently [53, 56, 43].[8] Quantitatively, the following theorem holds:

Theorem 1.3 ([53, 56, 38, 43]). *For any complete metric of doubling dimension* α *there exists a* 2^α*-doubling measure. If the metric is finite, such measure can be constructed efficiently, in time* $O(2^{O(\alpha)}\, n \log n)$.

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, starting from any (possibly empty) set of points that are at distance at least r from each other. It is often useful to consider r-nets in conjunction with doubling metrics, because of the following simple and well-known fact (e.g. see [25]):

Lemma 1.4. *Any* r*-net has at most* $(4r'/r)^\alpha$ *elements in any ball of radius* $r' \geq r$.

Proof. Let S be an r-net, and let B be a ball of radius $r' \geq r$. Recursively applying the definition of doubling dimension $2 + \lfloor \log r'/r \rfloor$ times, we cover B with at most $(4r'/r)^\alpha$ sets of diameter at most $r/2$. Each of these sets contains at most one node of S. □

2 A low-stretch routing scheme for doubling metrics

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. [14]; the ideas from this proof will be used in the subsequent results.

Theorem 2.1. *Consider a weighted graph* G *with out-degree* D_{out}*. Suppose its shortest-path metric has doubling dimension* α *and aspect ratio* Δ*. Then for any* $\delta \in (0, \frac{1}{4})$ *there is a* $(1+\delta)$*-stretch routing scheme on* G *with* $O(\alpha \log \frac{1}{\delta})(\log \Delta)$*-bit packet headers and routing tables of* $(\frac{1}{\delta})^{O(\alpha)}(\log \Delta)(\log D_{\mathit{out}})$ *bits. Moreover, such routing scheme can be efficiently computed.*

Proof. Let d be the shortest-path metric of G. For each $j \in [\log \Delta]$ let $\mathcal{G}_j$ be some $\Delta/2^j$-net on d; let $r_j = 4\Delta/\delta 2^j$ and define the j*-th ring of neighbors* of node u as the set $Y_{uj} = B_u(r_j) \cap \mathcal{G}_j$. Note that by Lemma 1.4 each node has at most $K = (16/\delta)^\alpha$ j-ring neighbors. The nodes in $\cup_j Y_{uj}$ are called the *neighbors* of u. Intuitively, we think that u has a virtual link to each of its neighbors; note that these virtual links are *not* the physical links in the underlying connectivity graph G.

To connect the virtual links with G, for each neighbor v the routing table of u will contain the *first-hop pointer* from u to v, which is, informally, the first edge of some shortest uv-path in G. We will define the first-hop pointers formally later in the proof.

[8] Bibliographic note: the original existence result (for compact metrics) is due to [53]. The proof has been simplified by [56] and extended to complete metrics in [38]. The algorithmic result builds on the construction from [56] and is due to [43].

Fix some node t; let us think of t as a potential target node. For any given j, by definition of Y_{tj} there exists a j-ring neighbor of t that lies within distance $\Delta/2^j$ from t; let us fix one such neighbor, call it f_{tj}. Consider the sequence $\{f_{tj} : j \in [\log \Delta]\}$. The nodes in this sequence "zoom" in on t as j increases. Let us call this sequence the *zooming sequence* of t.

A routing label of t will contain (a description of) its zooming sequence, which will be used to guide the routing as follows. Suppose node u wants to send a packet to node t. For some j node u has a neighbor $v = f_{tj} \in Y_{uj}$ that lies within distance δd_{ut} from t. Essentially, node u wants to forward the packet to v; here v becomes an intermediate target. In general, u does not have a direct link to v. Instead, the packet is sent via the edge uw which is the first-hop pointer to v. It will turn out that v is also a neighbor of w, so we can again use the first-hop pointer to v, and so on. This way the packet gets delivered to v via a shortest path using the first-hop pointers. Once the packet reaches v, a new intermediate target is selected. Eventually the next intermediate target that we choose will be the actual target t.

We want a routing table of each node u to list all its neighbors. Similarly, we want a routing label of each node t to list its zooming sequence. The simplest way to achieve this is to assign each node a global $\lceil \log n \rceil$-bit identifier, and just list the corresponding identifiers. However, this leads to unwanted extra $(\log n)$ factors in the storage requirements. Later in the proof we will show how to reduce storage using shorter local identifiers. No matter what routing tables and routing labels we use, all we need from them is summarized in the following claim (which is trivial if we use global identifiers).

For any two nodes (u, t), let us define j_{ut} be the maximum j such that $f_{ti} \in Y_{ui}$ for each $i \leq j$. Note that $j_{ut} \geq 0$ since $f_{(t,0)} \in \mathcal{G}_0 = Y_{(t,0)}$. Let g_{utj} be the first-hop pointer from u to f_{tj}, or *null* if $u = f_{tj}$.

Claim 2.2. *Given the routing table of u and the routing label of t we can find j_{ut} and g_{utj} for each $j \leq j_{ut}$.*

Now using this claim we will define the routing algorithm and prove its correctness. Then we provide a more space-efficient way to define routing tables and routing labels which will satisfy Claim 2.2 and lead to the desired storage complexity.

We start with a very useful fact about the zooming sequences: $f_{tj} \in Y_{uj}$ for a sufficiently small j.

Claim 2.3. *For any two nodes (u, t) and any $j \leq \lceil \log(\Delta/\delta d_{ut}) \rceil$ we have $f_{tj} \in Y_{uj}$. In particular, for any node t and any $j \in [\log \Delta]$ letting $f = f_{(t,j-1)}$ we have $f_{tj} \in Y_{fj}$.*

Proof: By definition $f_{tj} \in \mathcal{G}_j$. It is easy to check that f_{tj} lies within distance r_j from u, so $f_{tj} \in Y_{uj}$. The claim applies to $f = f_{(t,j-1)}$ since $d_{ft} \leq \Delta/\delta 2^j$. □

ROUTING ALGORITHM. For a packet with target t, the header consists of the routing label of t and the number $j \in [\log \Delta]$ such that f_{tj} is the current *intermediate target*; recall that the routing label of t contains the description of its zooming sequence such that Claim 2.2 holds. Suppose node u wants to send a packet to target t. Then using Claim 2.2 node u computes $j = j_{ut}$ and g_{utj}, chooses f_{tj} to be the intermediate target, and sends the packet along g_{utj}, the first edge on some shortest path from u to f_{tj}.

Now suppose node u receives a packet with target t and intermediate target f_{tj}. We will prove that in this case we have $j_{ut} \geq j$ (see Claim 2.4b). First node u checks whether it is the target.[9] If not, then via Claim 2.2 it computes j_{ut} and g_{utj} and, in particular, checks whether the intermediate target is u itself. If it is not, i.e. if g_{utj} is not null, then u just forwards the packet along the hop g_{utj}.

If u is indeed the current intermediate target, then it needs to select a new one. Specifically, it resets $j = j_{ut}$ and selects f_{tj} as the new intermediate target. Then it recomputes g_{utj} and forwards the packet along the corresponding 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.

[9] Without loss of generality, the routing table and the routing label of every node contain its global identifier.

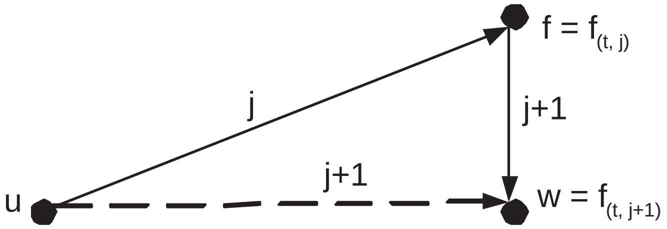

Figure 2: Translation between host enumerations of u and $f = f_{tj}$.

(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 2.3 $j \geq \lceil \log(\Delta/\delta d_{ut}) \rceil$, so f_{tj} lies within distance δd_{ut} from t.

(b) Let P be this packet. We will use induction on the path traversed by P. This path starts when some node u_0 chooses $w = f_{tj}$ as an intermediate target; then the current level is set to $j_{(u_0,\, t)}$. 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.

Indeed, note that $w \in Y_{uj} := B_u(r_j) \cap \mathcal{G}_j$. Since the packet is forwarded along a shortest uw-path, we have $d_{vw} < d_{uw} \leq r_j$. 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 $d_{vt} \leq d_{vw} + d_{wt} < r_j + \Delta/2^j$. Consequently, $d_{vt} + \Delta/2^i \leq r_i$. 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 $w = f_{tj}$ then P reaches w and traverses path of total length d_{uw}. This is because each node x on the route of P forwards it along g_{xtj}, the first hop of some shortest xw-path. □

Now it is straightforward to prove correctness of the routing algorithm:

Claim 2.5. *Every packet reaches its target and follows a path of stretch $1 + O(\delta)$.*

Proof. Consider a packet sent 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 $\sum_{i=0} \delta^i d_{ut}(1+\delta) \leq d_{ut}(1+O(\delta))$. □

It remains to provide space-efficient routing tables and routing labels which satisfy Claim 2.2. Recall that our goal is to replace $\lceil \log n \rceil$-bits global node identifiers with shorter 'local' identifiers.

For each node u, let us fix some enumeration $\varphi_{uj}(\cdot)$ of each ring Y_{uj}; let us call it the j-th *host enumeration* of u. Recall that an enumeration of a set S is a bijection $S \to [k]$, where $k = |S|$. Since the rings Y_{u0} coincide for all nodes u, we can guarantee that so do the corresponding enumerations φ_{u0}.

Consider nodes $f = f_{tj}$ and $w = f_{(t,j+1)}$, for some target t and integer j. Note that by Claim 2.3 we have $w \in Y_{(f,j+1)}$. Consider some node u such that $f \in Y_{uj}$ and $w \in Y_{(u,j+1)}$. For such triangles (u, f, w) (see Figure 2) we will provide a 'translation' between host enumerations of u and f, in the following sense: knowing $\varphi_{uj}(f)$ and $\varphi_{(f,j+1)}(w)$ we will be able to find $\varphi_{(u,j+1)}(w)$.

Specifically, for each $j \in [\log \Delta]$ the routing table of each node u will include the translation function $\zeta_{uj} : [K] \times [K] \to [K]$ such that

$$\zeta_{uj}\left(\varphi_{uj}(f),\ \varphi_{(f,\,j+1)}(w)\right) = \varphi_{(u,\,j+1)}(w) \text{ whenever } f \in Y_{uj} \text{ and } w \in Y_{(u,\,j+1)} \cap Y_{(f,\,j+1)},$$

and *null* otherwise. Clearly, each such function can be stored using $K^2 \lceil \log K \rceil$ bits. Recall that $K = (16/\delta)^\alpha$ is the maximal cardinality of each set Y_{uj}.

Let us formally define the first-hop pointers. For each node u we fix some enumeration $\phi_u(\cdot)$ of all outgoing links in the underlying connectivity graph G. For two nodes uv, we 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; each such pointer can be stored using only $\lceil \log D_{\text{out}} \rceil$ bits.

For every node t, let us encode its zooming sequence via host enumerations of its elements as follows: let us define $n_{t0} = \varphi_{t0}(f_{t0})$, and for each $j \geq 1$ let $n_{tj} = \varphi_{fj}(f_{tj})$, where $f = f_{(t,j-1)}$. This is well-defined because by Claim 2.3 f_{tj} is a j-ring neighbor of node f. It is easy to see that the sequence $\{n_{tj}\}$ can be stored using $O(\log K)(\log \Delta)$ bits.

DATA STRUCTURES: The *routing table* of a given node u consists of the translation functions ζ_{uj} and the first-hop pointers to all its neighbors. The *routing label* of a given node t is the sequence $\{n_{tj}\}$.

Having defined routing tables and routing labels, it remains to prove Claim 2.2. Here the difficulty is that node u needs to "decode" the zooming sequence of t, which is given indirectly: each element is given as an index in the host enumeration of the previous element. The proof follows in a straightforward way from our discussion of the translation functions. Indeed, let $m_j = \varphi_{uj}(f_{tj})$; this is well-defined for all $j \leq j_{ut}$. We will 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 $w = f_{(t,j+1)}$. Since we know $m_j = \varphi_{uj}(f)$ and $n_{tj} = \varphi_{fj}(w)$, we can find $m_{j+1} = \varphi_{(u,j+1)}(w)$ using the translation function ζ_{uj}. We iterate the above procedure while we can, i.e., while $w \in Y_{(u,j+1)}$. We stop exactly at $j = j_{ut}$. This completes the proof of Claim 2.2 and Theorem 2.1. □

3 Triangulation and distance labeling schemes

We start with the result on triangulation, then we elaborate it using the ideas from the previous section and achieve an optimal $(1+\delta)$-approximate distance labeling scheme. We use the following lemma which is implicit (but never articulated) in Slivkins [49] (see Appendix A for a self-contained proof).

Lemma 3.1. *Consider a finite 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 $\mathcal{F}$. Moreover, such $\mathcal{F}$ can be efficiently computed.*

It is easy to see that if μ is a doubling measure then for every node u this (ϵ, μ)-packing $\mathcal{F}$ has the two useful local 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 $\mathcal{F}$. The notion of (ϵ, μ)-packing allows us to state these properties in terms of the underlying doubling measure, and, moreover, to generalize them to arbitrary probability measures. In fact, we will use (ϵ, μ)-packings such that μ is the normalized counting measure $\mu(S) = |S|/n$.

Theorem 3.2. *For any $\delta \in (0, \frac{1}{2})$ any metric has a $(0, \delta)$-triangulation of order $(\frac{1}{\delta})^{O(\alpha)} \log n$, 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 us greedily construct a sequence of nested r-nets $\mathcal{G}_{\lceil \log \Delta \rceil} \subset \ldots \subset \mathcal{G}_1 \subset \mathcal{G}_0$, where $\mathcal{G}_j$ is a 2^j-net for each $j \in [\log \Delta]$. Then for each $i \in [\log n]$ let us define the Y_i-neighbors of u as all nodes in $B_u(12 r_{ui}/\delta)$ that lie in $\mathcal{G}_j$ such that $j = \max(0,\ \lfloor \log(\frac{\delta}{4}\, r_{ui}) \rfloor)$.

The above construction is efficiently computable since so are $(2^{-i}, \mu)$-packings and the nested 2^j-nets.

Let's bound the number of neighbors. Fix 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. By Lemma 1.4 there are at most $(1/\delta)^{O(\alpha)}$ Y_i-neighbors. 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 each node pair uv and any $i \in [\log n]$ we have $|r_{ui} - r_{vi}| \le d_{uv}$.*

Proof: Since $B_v(r_{vi}) \subset B_u(d_{uv} + r_{vi})$, the latter ball contains at least $n/2^i$ nodes, so it follows that $r_{ui} \le d_{uv} + r_{vi}$. Similarly, since $B_{ui} \subset B_v(d_{uv} + r_{ui})$ it follows that $r_{vi} \le d_{uv} + r_{ui}$. □

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 $r_{ui} \le r + d \le r_{(u,i-1)}$. We choose i with respect to u, but by Claim 3.3 this yields some bounds on r_{vj}'s as well; specifically, $r_{(v,i-1)} \ge r$ and $r_{vi} \le r + 2d$.

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 $w = h_B$. If $6r_{vi} \le \delta d$ then $B_v(6r_{vi})$ is contained in both $B_{(u,i-1)}$ and $B_{(v,i-1)}$, hence node w is an X_i-neighbor of u and v, contradiction. Similarly, $B_u(6r_{ui})$ contains some ball $B \in \mathcal{F}_i$, so if $6r_{ui} < \delta d$ then the node $w = h_B$ is an X_i-neighbor of u and v, contradiction. Therefore letting $x = \delta d/6$ we have $x \le r_{ui} \le r + d$ and $x < r_{vi} \le r + 2d$. 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 $r_{ui} \ge x$ it follows that $B \subset B_u(12 r_{ui}/\delta)$ and $j \le \log(6 r_{ui})$; moreover, $j \ge \lfloor \log(\delta r_{ui}/4) \rfloor$ since $r_{ui} \le r + d$. Therefore by definition w is a Y_i-neighbor of u. Similarly, w is a Y_i-neighbor of v, contradiction.[10] Theorem proved.

□

Our $(0,\delta)$-triangulation can be extended to a $(1+\delta)$-approximate distance labeling scheme where each label consists of $[O(\frac{1}{\delta})]^{O(\alpha)}(\log n)(\log n + \log\log \Delta)$ bits, matching the result of Mendel and Har-Peled [43]. Indeed, we assign each node u a unique $\lceil \log n \rceil$-bit identifier $\textsc{id}(u)$ and store each neighbor u of v as a pair ($\textsc{id}(u)$, d_{uv}). We use the upper bound D^+ for the distance estimate, so it suffices to store d_{uv} as a $O(\log \frac{1}{\delta})$-bit mantissa and $O(\log\log \Delta)$-bit exponent. This is because if two numbers x', y' are $(1+\delta)$-approximations of the true values x, y, then the sum $x' + y'$ is $(1+\delta)$-approximation of $x + y$.[11]

Extending a result from [24], Mendel and Har-Peled [43] constructed a family of doubling metrics for which any 1.9-approximate distance labeling scheme needs

$$\Omega(\log n)(\log\log \Delta - \log\log n) \tag{1}$$

bits per label. This is $\Omega(\log n)(\log\log \Delta)$ as long as $\Delta \ge n^{\log^c n}$ for any constant $c > 0$. Their construction works for infinitely many n and for (essentially) a full range of possible values of the aspect ratio Δ. Specifically, it works for some Δ in every interval $[(n/2)^M;\ n^M]$ such that $M \ge 2$ is an integer.

Our next result shows that we can elaborate our distance labeling scheme, getting rid of the $\lceil \log n \rceil$-bit node identifiers and achieving $O_{\alpha,\delta}(\log n)(\log\log \Delta)$-bit labels. This is an improvement whenever $\log\log \Delta = o(\log n)$. Moreover, for any $\Delta \ge n^{\log n}$ and fixed α, δ we match the lower bound (1) up to constant factors.

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

[11] Note that the difference $x' - y'$ is not necessarily a good approximation for $x - y$, so we cannot use the lower bound D^-.

Theorem 3.4. *For any $\delta \in (0, \frac{1}{2})$ 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 will elaborate the construction in the proof Theorem 3.2 using the ideas from the proof of Theorem 2.1. Specifically, we will use the zooming sequences and the host enumeration technique.

Keep the notation from the proof of Theorem 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 $r_{ui}/4$ 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* of u, and denote it f_u. Moreover, for each node u fix some enumeration $\varphi_u(\cdot)$ of all its neighbors; we call it the *host enumeration* of u.

From the proof of Theorem 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 identifiers. 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 will be able to identify, sequentially, all f_{vi} from $i = 0$ to some i_0 such that $f = f_{(v,\, i_0)}$ 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 using $\lceil \log n \rceil$ bits to identify the elements of a zooming sequence of v is again too expensive. Moreover, $f_{(v,i+1)}$ might not be a neighbor of f_{vi}, and w_0 might not be a neighbor of f, so we cannot quite use the host enumeration technique the way it is used to prove Theorem 2.1. Instead, for every node we will define another set of nodes called *virtual neighbors* in such a way 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 only to define ”pointers” between consecutive elements in a zooming sequence: for each i, $f_{(v,i+1)}$ is given only a pointer from f_{vi}, namely as an index in some fixed enumeration of virtual neighbors of f_{vi} (which we call a *virtual enumeration* of f_{vi}). If every node has at most N virtual neighbors, then each such pointer uses only $\lceil \log N \rceil$ bits. The crux of the proof is to define a sufficiently small set of virtual neighbors with the desired properties. We will actually have $N = O_{\alpha,\delta}(\log n)\log(\Delta)$. [12] To translate virtual enumerations into host enumerations, we define a translation function similar to the one in the proof of Theorem 2.1. □

In the remainder of this section we give the full proof of Theorem 3.4. Keep the notation from the proof of Theorem 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 $r_{ui}/4$ 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

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

The elements of T_u will be called the *virtual neighbors* of u. Note that the definition of Z_{uj} is similar to that of Y_{uj} since both are defined in terms of $B_u(r)$ and $\mathcal{G}_l$ for some related values of r and l; essentially, r and 2^l determine the corresponding distance scale. The principal difference is that for the Y-type neighbors this distance scale is $\Theta(r_{uj})$, whereas for the Z-type neighbors it is $\Theta(2^j)$.

[12] Note that $\log N = (\log\log n) + (\log\log\Delta) + O_{\alpha,\delta}(1)$. To avoid the $(\log\log n)$ factor in the theorem statement, we note that due to Lemma 1.2 it is subsumed by $(\log\log\Delta)$.

We will need the following crucial facts about virtual neighbors:

Claim 3.5. *Fix node u and $i \in [\log n]$; let $f = f_{(u,i-1)}$. Then*

(a) if $r_{ui} \leq r_{(u,i-1)}/12$ then the nearest X_i-neighbor of u is an X_i-neighbor of f.

(b) if $x \in [\frac{\delta}{4} r_{ui};\, 6r_{ui}]$, $x \leq r_{(u,i-1)}/2$ 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: Let $r = r_{(u,i-1)}$ and $z = r_{(f,i-1)}$.

For part (a), note that by Lemma 3.1 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 $B_u(6r_{ui})$. Then by Claim 3.3 $z - d_{uf} \geq r - 2d_{uf} \geq r/2 \geq 6r_{ui}$, and consequently $B_u(6r_{ui}) \subset B_u(z - d_{uf}) \subset B_f(z)$. 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 by part (a) v is an X_i-neighbor of f. Moreover, since $d_{uv} \leq 6r_{ui}$ it is easy to see that $w \in Z_{vj}$ for $j = \lceil \log(d_{uv} + x) \rceil$. Since $w \in Z_{vj}$ and $v \in X_f$, it follows that $w \in T_u$.

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

Let us define the labels of nodes. For each node u, let us fix some enumeration $\varphi_u(\cdot)$ of $X_u \cup Y_u$; call it a *host enumeration* of u. Since any ball B_{u0} contains all nodes, the sets X_{u0} coincide for all u, and so do the sets Y_{u0}. Therefore we can guarantee that 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

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

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 such that for each $j \in [|X_u \cup Y_u|]$, the j-th entry of this array is the distance from u to $\varphi_u(j)$, the j-th element of $X_u \cup Y_u$. This distance is encoded as a $O(\log \frac{1}{\delta})$-bit mantissa and $O(\log\log\Delta)$-bit exponent. Moreover, the label of u will contain the maps ζ_{ui}, for each $i \in [\log n]$; each ζ_{ui} is represented by an ordered set of triples (x, y, z) such that $\zeta_{ui}(x, y) = z \neq null$. Finally, the label of u will contain the zooming sequence of u. 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}; recall that by Claim 3.5(c) $f_{(u,i+1)}$ is indeed a virtual neighbor of f_{ui}. This completes the definition of the node labels.

By the proof of Theorem 3.2, the cardinality of each X_{ui}, Y_{ui} and Z_{uj} is upper-bounded by some $K = (1/\delta)^{O(\alpha)}$. 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 Theorem 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 do not 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 similarly if it is close to u. First we need a basic claim about f_{vj}'s:

Claim 3.6. *For any $j \leq i-1$ 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 3.5(c) f is a virtual neighbor of f_{uj}; note that at this point the only description of f available to our algorithm is its index in the virtual enumeration of f_{uj}. Given this index, we can find $\varphi_u(f)$ via the translation map ζ_{uj} and (by Claim 3.6) we can find $\varphi_v(f)$ via the translation map ζ_{vj}.

Now, assuming w_0 is closer to v than to u, we will identify it using $f = f_{(v,i-1)}$. (If w_0 is closer to u, we can identify it similarly using $f_{(u,i-1)}$.) Suppose that w_0 is a virtual neighbor of f. Then we can just check all virtual neighbors of f that are common neighbors of u and v. More precisely, we look at the translation maps $\zeta_{(u,\,i-1)}$ and $\zeta_{(v,\,i-1)}$ and check all entries of the form $(f, \cdot)$; both maps have an entry (f, x) if and only if node $\psi_f(x)$ is both a virtual neighbor of f, and a common neighbor of u and v.

It remains to show that w_0 is a virtual neighbor of f. According to the proof of Theorem 3.2, we can assume that either

(a) $r_{vi} \leq \delta d/6$ and w_0 is the nearest X_i-neighbor of v, or
(b) $r_{vi} > \delta d/6$ and $w_0 \in \mathcal{G}_l$ such that $l = \lfloor \log \delta d \rfloor$.

In case (a) $w_0 \in T_f$ by Claim 3.5a since $r_{(v,i-1)} > d \geq 2\delta d$; in case (b) $w_0 \in T_f$ by Claim 3.5(b) since $x = \delta d$ matches the conditions in the claim. This completes the proof of Theorem 3.4.

4 Low-stretch routing schemes, revisited

First we'll use our result on distance labeling to obtain a *really* simple $(1+\delta)$-stretch routing scheme for doubling graphs, then we merge the techniques from the previous two sections to obtain routing schemes for doubling graphs with super-polynomial aspect ratio. We also discuss extensions to *routing on metrics*.

Theorem 4.1. *In the setting of Theorem 2.1, for any $\delta \in (0,1)$ there exists a $(1+\delta)$-stretch routing scheme with $2^{O(\alpha)}(\phi \log n)$-bit packet headers and routing tables of $(\frac{1}{\delta})^{O(\alpha)}(\phi \log n)(\log \Delta)$ bits, where $\phi = \log(\frac{1}{\delta} \log \Delta)$. Such routing scheme can be efficiently computed.*

Proof. For every node u, let $\textsc{id}(u)$ be its unique $\lceil \log n \rceil$-bit ID. Fix a $3/2$-approximate distance labeling scheme with $2^{O(\alpha)}(\log n)(\log\log \Delta)$-bit labels, which is guaranteed by Theorem 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 $\textsc{id}(u)$. Each packet header consists of L_t and $\textsc{id}(t')$ 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 1.4 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. From the claim and the labeling scheme it follows that $D(L_t, L_v) \leq \frac{3}{2}\,\delta d$. So when node u selects a new intermediate target for a packet with final target t, it selects a neighbor v within distance $\frac{3}{2}\,\delta d$ 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 2.4(c) holds: each packet follows a shortest path to each intermediate target. To reach the i-th intermediate target, $i \geq 1$, the packet traverses a path of length at most $d_{ut}(1 + \frac{3}{2}\,\delta)\delta^{i-1}$. Therefore the total path length is at most $d_{ut}[1 + O(\delta)]$. □

We note that the bounds in Theorem 2.1 are unsatisfactory for metrics with large aspect ratio, and an extension that alleviates the dependency on Δ for weighted graphs that contain near-shortest paths with small hop-counts; this property is, intuitively, a natural property of a "good" network topology. For concreteness we'll state this result for an illustrative special case; so as not to disrupt the flow, the general case and the full proof are deferred to Appendix B.

Theorem 4.2. *Suppose the aspect ratio is at most 2^n and for some $\delta \in (0,1)$ any two nodes in the input graph G are connected by a $(1+\delta)$-stretch path with at most $k \log n$ hops, where $k = (\frac{1}{\delta})^{O(\alpha)}$ and α is the doubling dimension. Then there exists a $(1+\delta)$-stretch routing scheme on G with $O(k \log^3 n)$-bit routing tables and $O(k \log^2 n)$-bit packet headers. Such routing scheme can be efficiently computed.*

Proof Sketch. We will combine the ideas of Theorem 3.4 and Theorem 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 will use the rings, the zooming sequences, and the enumerations as defined in Theorem 3.4. In fact, we'll just use all definitions from the proof Theorem 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 an elaboration of the routing in Theorem 2.1: we use intermediate targets that zoom in towards the true target. If at the current node u the intermediate target is not set, we select a new intermediate target w among the neighbors of u, using the zooming sequence f_t and other data in the routing label of t. To save space in the packet header, this w will be represented not by a global id, but by its number in a virtual enumeration of some f_{ti}. Now suppose an intermediate target w is set, and the packet is at node v. If w is a neighbor of v and, moreover, v can *identify* this w (i.e. find $\varphi_v(w)$), then v forwards the packet using the first-hop pointer to w.

Note that this routing algorithm might fail since it might not be possible to find a 'good' new intermediate target, or identify it at some intermediate node v. However, the algorithm is set up so that this can happen *only* if there is a large gap between d_{vt} and the largest r_{vi} that lies below $\frac{4}{3}d_{vt}$. Verifying this claim is the crux of the proof of the theorem.

If the first routing mode fails, we enter the *second routing mode*, and we stay in this mode till we are done. 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 $w = h_B$ be the node selected from B in Theorem 3.2; without loss of generality say it is a center of B. It is easy to see that the ball $B' = B_{(w,\,i-1)}$ 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

	out-degree	routing table size, bits	packet header size, bits
Chan et al. [14]	$(\frac{\alpha}{\delta})^{O(\alpha)}(\log\Delta)$	$(\frac{\alpha}{\delta})^{O(\alpha)}(\phi\log\Delta)$	$O(\alpha\log\frac{1}{\delta})(\log\Delta)$
Theorem 2.1	$(\frac{1}{\delta})^{O(\alpha)}(\log\Delta)$	$(\frac{1}{\delta})^{O(\alpha)}(\phi\log\Delta)$	same as above
Theorem 4.1	same as above	$(\frac{1}{\delta})^{O(\alpha)}(\phi\log\Delta)(\log n)$	$2^{O(\alpha)}(\phi\log n)$
Theorem 4.2	$(\frac{1}{\delta})^{O(\alpha)}(\log n)$	$(\frac{1}{\delta})^{O(\alpha)}(\phi\log n)(\log\log n)$	$O(\alpha\phi\log n)$
Follow-up work [3]	$(\frac{1}{\delta})^{O(\alpha)}(\log n)$	$(\frac{1}{\delta})^{O(\alpha)}(\log^2 n)$	$\lceil\log n\rceil$
	doubling dimension α, aspect ratio Δ, and $\phi = \log(\frac{1}{\delta}\log\Delta)$		

Table 2: $(1+\delta)$-stretch routing schemes for doubling metrics.

shortest-path tree T_B rooted at w, such that given $\textsc{id}(t)$ it is possible to route from w to the node v_t that stores a path to t. Here it is crucial that we are free to choose the labels for T_B and the mapping v_t from B' to B any way we like. We will choose so that for a given link in the shortest-path 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 w (which is a neighbor of u) is designated as the intermediate target, and the packet is routed to w via the first-hop pointers. From w the packet is routed to v_t via the shortest-path tree T_B. Then node v_t writes the full route to t into the packet header and sends the packet to t. More precisely, v_t will store a $(1+\delta)$-approximate shortest path to t with $k\log n$ hops (the existence of such a path is guaranteed by the theorem statement). Each hop in this path can be encoded by $\lceil\log D_{\text{out}}\rceil \leq \lceil\log n\rceil$ bits, where D_{out} is the out-degree, so the entire path can be stored using at most $k\log^2 n$ 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. □

4.1 Routing schemes on metrics

Finally, we note that all our results on routing schemes on doubling graphs extend to *routing on metrics*. Here we are given a metric (V,d), and we need to construct a routing scheme on some weighted directed graph $G=(V,E)$. The crucial point is that we are free to choose the (unweighted) set of edges E; essentially, it can be seen as an overlay network on V. The edge-weights are determined by the metric: for any edge $uv \in E$, the weight of this edge is d_{uv}. In addition to the maximal size of a routing table and the maximum size of a routing label, the out-degree of E becomes another parameter to be optimized.

Extension to routing on metrics is almost trivial. In fact, in all our proofs we first construct a routing scheme on a low-degree overlay network (which is, by definition, a routing scheme on a metric), and then with some additional work adapt it to the underlying connectivity graph. The quantitative results are summarized in Table 2; we omit the appropriate modifications (simplifications) of the proofs. Note that in this setting Theorem 4.2 does not need any assumptions about low-stretch, low hop-count paths.[13]

FOLLOW-UP WORK: Following the publication of the conference version of this paper, Abraham et al. [3] provided a fine-tuned version of Theorem 4.2, where they completely eliminate the dependence on aspect ratio Δ; see Table 2 for quantitative results.

[13] Recall that in the proof sketch of Theorem 4.2 node v_t stores a low-stretch, low hop-count path from v_t to target t. For routing schemes on metrics, we no longer need such a path; instead, we can create a direct link from v_t to t.

5 Searchable small-world networks

In this section we consider searchable small-world networks on *metrics*. To the best of our knowledge, the most general previous result in this direction is for metrics such that the growth rate of balls (defined as the ratio $|B_u(2r)|/|B_u(r)|$) is both upper- and lower-bounded by constants that are stictly greater than 1; let us call such metrics *UL-constrained*. This result can be easily achieved from Kleinberg's original construction for two-dimensional grids [30]. Here we extend small worlds to doubling metrics.

We will consider routing algorithms where the next hop is chosen among the current node's contacts, by looking only at distances to these contacts and distances from these contacts to the target. Recall from Section 1 that we call such routing algorithms *strongly local*. A very natural routing algorithm is the *greedy* algorithm: select the contact that is closest to the target.

As searchable small-worlds on *metrics* have not been previously studied explicitely, we need to give a formal definition. For simplicity let us focus on the case when the routing algorithm is strongly local.

Definition 5.1. A ***small-world model*** on a metric (V, d) consists of the following two items:

- a distribution over directed graphs on V (from which the graph of contacts is sampled) such that the out-links (contacts) of a given node u are chosen independently for different nodes u;
- a strongly local routing algorithm that, given the target, selects the next hop among the current node's contacts.

For a given graph of contacts, the routing algorithm recursively handles the *queries* of the form (u, t) where u is the initial node and t is the target.

Let us define the *out-degree* of a small-world model as the maximal possible out-degree of its graph of contacts. For a given metric, we would like to balance two conflicting objectives: the out-degree and the length of paths found by the routing algorithm.

We would like the distribution of contacts to have the following informal property: if from the point of view of a given node u two nodes v and w are similar, then these two nodes should have a similar probability of being chosen as contacts of u. Indeed, in our constructions the probability that node v is chosen as a long-range contact of a node u depends only on the rank of d_{uv} among distances from u to all other nodes, and the ratios $\mu(v)/\mu(B_{uj})$, where μ is a doubling measure and $\{B_{uj} : 0 \leq j \leq \lceil \log \Delta \rceil\}$ are balls around u with exponentially increasing radii. Here the doubling measure of v quantifies how dense is the metric in the vicinity of v; intuitively, we need to oversample nodes that lie in very sparse neighborhoods.

Now we can describe our results. Let Δ be the aspect ratio of the metric. While it is relatively straightforward to achieve out-degree $O(\log n)(\log \Delta)$ and $O(\log \Delta)$-hop paths, 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 since we need some long-range contacts for each one of the $(\log \Delta)$ distance scales. However, our second result breaks this barrier. Moreover, in Section 5.2 we argue that for UL-constrained metrics our small worlds essentially coincide with those induced by Kleinberg's *group structures* from [31].[14]

To break the above-mentioned $O(\log \Delta)$ barrier we need to use a non-greedy routing algorithm. Yet, we can still make this algorithm strongly local, so that on each routing step we do not need to use any extra information beyond the current node's list of neighbors. To the best of our knowledge it is the first non-greedy strongly local routing algorithm in the literature.

Let us state the main result of this section. Note that we upper-bound the actual (as opposed to expected) hop counts, so that with high probability our upper bound is valid for all possible queries.

[14] The guarantees in [31] apply to UL-constrained metrics that are subsets of some ℓ_p space, $p \geq 1$. However, the construction itself is well-defined for any metric.

Theorem 5.2. *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 routing 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 $2^{O(\alpha)}(\log n)^2\,(\log \Delta)^{1/2}\,(\log\log \Delta)$ and a strongly local routing algorithm such that w.h.p. all queries complete in $O(\log n)$ hops.

Proof Sketch. To be consistent with the earlier parts of the paper, let us use words 'contact' and 'neighbor' interchangeably. 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; let us call these neighbors the *Y-type neighbors*. It is easy to make sure that the greedy algorithm reduces the distance by at least a factor of two at each step, so any query will take $(\log \Delta)$ steps to complete. However, reducing the distance by a constant factor at each step does not suffice to guarantee $O(\log n)$-hop query paths when the aspect ratio Δ is large.

Let us denote $B_{ui} = B_u(r_{ui})$, where $r_{ui} = r_u(2^{-i})$. In other words, B_{ui} is the smallest ball around node u that contains at least $n/2^i$ nodes, and r_{ui} is the radius of this ball. Let t be the target node, and let us consider the annuli $B_{(t,\,i-1)} \setminus B_{ti}$, indexed by $i \in [\log n]$. Instead of trying to reduce the distance to target t by a constant factor at each step, we will now focus on how quickly the routing algorithm gets us from one such annulus to the next one. Specifically, to guarantee $O(\log n)$-hop query paths, we will need small-world models with the following property:

(*) if the current node u in the routing path lies inside ball $B_{(t,\,i-1)}$ but outside ball B_{ti}, then we get inside ball B_{ti} in at most a constant number of hops.

This property is non-trivial when the radius r_{ti} is much smaller than the distance between u and the target.

To prove part (a) we keep the Y-type neighbors. It turns out that we satisfy property (*) w.h.p. if we throw in another collection of rings of neighbors where the neighbors are distributed with respect to the counting measure; let us call these neighbors the *X-type neighbors*. Specifically, we get from u into the ball B_{ti} using only *two* hops; the one intermediate hop leads from u to some node within distance $d_{ut}/4$ from t.

To prove part (b), however, using all Y-neighbors is not an option since there are too many of them. Instead, we will need to *prune* them. From part (a) it will follow that after we get within distance $d_{ut}/4$ from t, the next hop gets us inside B_{ti}. However, u might not have a neighbor that is sufficiently close to t. To handle this case, we will need to use a non-greedy routing choice, specifically:

(**) if the current node u has no contacts within distance $d_{ut}/4$ from the target node t, then we choose the contact v that is farthest from u subject to the constraint $d_{uv} \leq d_{ut}$.

Intuitively, if we cannot make a sufficiently good progress towards the target, this is because the current node u happens to be in a particularly 'bad' neighborhood. We want the next hop to take us away from this 'bad' neighborhood, and place us into a 'good' neighborhood. This is why we want the next hop to take us to some node v which is far away from node u. Furthermore, we want to *prove* that we necessarily land in a 'good' neighborhood. To prove this we must use the 'badness' of u (since otherwise node v is no better than node u as far as we are concerned). Therefore we do not want to get *too* far from node u, which is expressed by the constraint $d_{uv} \leq d_{ut}$.

To make (**) work, we introduce yet another family of neighbors, which we call the *Z-type neighbors*. Our argument proceeds as follows. If node v is a contact of the current node u, let us say that v is *good* if the ratio d_{uv}/d_{ut} is large enough, yet smaller than 1. We will show that if the current node u is in a 'bad' neighborhood, then any good contact v is in a 'good' one. Moreover, (**) will necessarily find a 'good' contact if u has one. So our job is to make sure that node u has at least one 'good' contact. And indeed with high probability node u will have at least one 'good' contact among the Z-type neighbors □

5.1 Full proof of Theorem 5.2

Let us fill in the details. For simplicity let us assume that in the input metric all distances are distinct. Recall that $B_{ui} = B_u(r_{ui})$, where $r_{ui} = r_u(2^{-i})$ is the radius of the smallest ball around u that contains at least $n/2^i$ nodes. Fix an absolute constant c to be specified later. Recall that α is the doubling dimension; let μ be the $2^{O(\alpha)}$-doubling measure (recall that such measure exists by Theorem 1.3). 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, where c is a sufficiently large constant to make the Chernoff Bounds work out (see Footnote 15 below), and let X_{ui} be the set of selected nodes. Let $X_u = \cup X_{ui}$; these are the *X-type neighbors* of u. Note that w.h.p. $|X_u| \leq O(\log^2 n)$.

Proof of part (a): Let us select the Y-type neighbors of a given node u 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 $Y_u = \cup Y_{uj}$; these are the *Y-type neighbors* of u. Define the set of neighbors of u as $X_u \cup Y_u$. Note that the out-degree of u is within the claimed bound; in particular, we upper-bound $|X_u|$ using Lemma 1.2.

We need to prove that property (*) holds. Suppose t is the target and u is the current node. Let us choose i such that node u lies in the annulus $B_{(t,i-1)} \setminus B_{ti}$. Let us denote $d = d_{ut}$ and $j = \lceil \log(1.25\, d) \rceil$. Note that the set Y_{uj} contains a node w that is within distance $d/4$ from target t.[15] Therefore the greedy routing algorithm will choose such node for the next hop.

If $r_{ti} \geq d/4$ then we are done. Now suppose $r_{ti} < d/4$. By our choice of i we have $r_{(t,\, i-1)} > d$. By Claim 3.3 It follows that $r_{(w,\, i-1)} > \frac{3}{4} d$, so $B_{ti} \subset B_t(d/4) \subset B_{(w,\, i-1)}$. Since ball B_{ti} contains at least a half of the nodes of the ball $B_{(w,\, i-1)}$, it follows that with high probability the set $X_{(w,\, i-1)}$ contains a node in B_{ti}, and we are done. □

Proof of part (b). In the remainder of this subsection we will prove part (b) of the theorem. As we discussed in the proof sketch, we will introduce a new family of contacts (called *Z-type neighbors*), and define the pruned version of the Y-type neighbors.

For a given node u, let us select the contacts as follows. Let us denote $x = \sqrt{\log \Delta}$ and $\rho_j = 2^{(1+1/x)^j}$. Let us consider the annuli $B_u(\rho_j) \setminus B_u(\rho_{j-1})$, indexed by j. For each j such that $\rho_j \leq \Delta$ let us pick a node z_{uj} uniformly at random from the j-th such annulus, provided that it is non-empty; else let z_{uj} be the closest node to u that lies outside $B_u(\rho_j)$, ties broken arbitrarily. Let $Z_u = \cup_j \{z_{uj}\}$; these are the *Z-type neighbors* of u.

For each $i \in [\log n]$ and each (signed) integer j such that

$$|j| \leq (3x+3)(\log\log \Delta) \text{ and } r_{(u,i+1)} < r_{ui} \cdot 2^j < r_{(u,i-1)},$$

let us 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 $Y_{(u,i,j)}$ be the set of selected nodes. Let $Y_u = \cup_{ij} Y_{(u,i,j)}$; these are the *Y-type* neighbors of u. Define the set of neighbors of u as $X_u \cup Y_u \cup Z_u$.

Let us check that the out-degree is small enough. Indeed, there are at most $O(\log^2 n)$ X-type neighbors. Each set $Y_{(u,i,j)}$ contains at most $2^{O(\alpha)}(\log n)$ nodes. Since for these sets there are at most $(\log n)$ valid indices i and at most $O(x \log\log \Delta)$ valid indices j, the number of Y-type neighbors is below the claimed upper bound. Finally, for the Z-type neighbors it suffices to note that $\rho_j \leq \Delta$ implies $j \leq O(x)(\log\log \Delta)$.

[15] More precisely, by Chernoff Bounds for large enough constant c with high probability this happens for all (u,t) pairs simultaneously. In the rest of the proof we will omit these straightforward applications of Chernoff Bounds.

The routing algorithm is simple. Suppose u is the current node and t is the target. If u has a contact within distance $d_{ut}/4$ from t then we greedily choose the contact that is closest to t. Else we do the non-greedy step (**).

This completes the specification of our small-world model; now we need to prove that our routing algorithm satisfies property (*). Suppose t is the target and u is the current node. Let us choose i such that node u lies in the annulus $B_{(t,i-1)} \setminus B_{ti}$. We will show that we get inside the ball B_{ti} in at most three hops.

Indeed, let $d = d_{ut}$ and note that as proved in part (a), if we get within distance $d/4$ from target t then in at most one more hop we are done. Let us consider the hard case: suppose node u does not have a contact in $B_t(d/4)$. Let us choose an integer l such that $r_{ul} \leq d \leq r_{(u,\,l-1)}$. It is easy to see that

$$r_{ul} \cdot 8^{x+1} < 1.25\, d < r_{(u,\,l-1)}/8^{x+1}. \tag{2}$$

Indeed, if the first inequality fails then for $j = \lfloor \log(d/r_{ul}) \rfloor$ some node from $Y_{(u,l,j)}$ lies in $B_t(d/4)$, contradicting the assumption that node u does not have contacts in $B_t(d/4)$. If the second inequality fails, then similar contradiction arises with the set $Y_{(u,\,l-1,\,\cdot)}$.

Now let us 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 the non-greedy step (**) will choose some contact w of u such that

$$d/8^x \leq d_{uw} \leq d. \tag{3}$$

In particular, by (2) and (3) it follows that

$$4r_{ul} < d_{uw} < r_{(u,\,l-1)}/4. \tag{4}$$

Now that we are at w we will be able to make progress towards t. To ensure property (*), the next hop should get us from w to within distance $d_{wt}/4$ from t. Since $d_{wt} > d/4$ by our assumption, it suffices to get inside the ball $B_t(d/16)$. (Note that if the routing 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 $Y_{(w,l,j)}$ for the appropriately chosen j.

Claim 5.3. $d_{uw} - r_{ul} \leq r_{wl} \leq d_{uw} + r_{ul}$.

Proof. The second inequality follows since the ball $B_w(d_{uw} + r_{ul})$ contains the ball B_{ul} and therefore has cardinality at least $n/2^l$. Suppose the first inequality fails. Then the balls B_{wl} and B_{ul} are disjoint; since both balls lie inside $B_u(d_{uw} + r_{wl})$, the latter ball has cardinality at least $n/2^{l-1}$. It follows that $r_{(u,\,l-1)} \leq d_{uw} + r_{wl}$. However, using (4) we have $d_{uw} + r_{wl} \leq 2d_{uw} + r_{ul} < r_{(u,\,l-1)}$, contradiction. □

Combining Claim 5.3 and (4), it follows that $r_{wl}/d_{uw} < (\frac{3}{4}, \frac{5}{4})$. Let us denote $r = d_{wt} + d/16$. Then

$$r \leq 1.07\, d + d_{uw} \leq d_{uw}\,(1.07 \cdot 8^x + 1) < 2^{3x+1}\, r_{wl} \tag{5}$$

In (5) the first inequality follows simply because $d_{wt} \leq d + d_{uw}$, and the second inequality holds by (3).

Let us choose j such that $2^{j-1} < r/r_{wl} \leq 2^j$. Then by (5) we have $j \leq 3x + 2$, and by definition of r we have $B_t(d/16) \subset B_w(r_{wl} \cdot 2^j)$. The radii of these two balls are within a constant factor because

$$\begin{cases} r = \Theta(r_{wl} \cdot 2^j) & \text{by definition of } j \\ r = \Theta(d) & \text{by definition of } r\text{, since } d_{wt} \leq d + d_{uw} \leq 2d. \end{cases}$$

Therefore the set $Y_{(w,l,j)}$ is well-defined, and it follows that with high probability the ball $B_t(d/16)$ contains a node from $Y_{(w,l,j)}$. This completes the proof of part (b) of the theorem.

5.2 Comparison with Kleinberg's small worlds

Let us argue that our small-world models generalize one of the Kleinberg's small worlds. Specifically, we consider the *group structures* from [31] applied to balls in a metric (it was one of the two original applications described in [31]). This small-world model, call it STRUCTURES, can be defined as follows. For any two nodes (u, v), let x_{uv} be the smallest cardinality of a ball containing both u and v. For each node u, define a probability distribution π_u on V (the set of all nodes) by $\pi_u(v) = c_1/x_{uv}$, where c_1 is the suitable normalization factor. Each node u has $\Theta(\log^2 n)$ neighbors chosen independently from distribution π_u. The routing algorithm is greedy.

On UL-constrained metrics our two small-world models essentially coincide with STRUCTURES:

Theorem 5.4. *For UL-constrained metrics, both small-world models in Theorem 5.2 share the following properties with* STRUCTURES*:*

(a) with high probability, any target is found in $O(\log n)$ steps from any starting node.
(b) the routing algorithm is greedy.
(c) each node has $k = \Theta(\log^2 n)$ neighbors.
(d) $\Pr[v$ is a neighbor of $u] = \Theta(\log n)/x_{uv}$, for any nodes (u, v).[16]

Proof. Part (a) is trivial because any UL-constrained metric has a polynomially bounded aspect ratio. For part (b) note that the routing algorithm in Theorem 5.2a is greedy by definition, and in Theorem 5.2b the non-greedy step is taken *only* if there is no neighbor that would reduce the distance to the target by the factor of 4. It is easy to show that if the underlying metric is UL-constrained then the set $X_u \cup Y_u$ will contain such a neighbor, so in Theorem 5.2b the routing algorithm is greedy as well and, moreover, the Z-type neighbors are never used.

Part (c) and (d) follow from the following observations:

(i) On a UL-constrained metric, the aspect ratio is poly-log in n, and the counting measure is doubling.
(ii) For any two nodes (u, v) in a UL-constrained metric, $|B_u(d_{uv})|$ is within a constant factor of x_{uv}.
(iii) In a UL-constrained metric, for any node u and any $i \in \log[n]$ there can be at most a constant number of balls $B_u(2^j)$, $j \in [\log \Delta]$ that are sandwiched between B_{ui} and $B_{(u,i+1)}$, where B_{ui} is the smallest ball around u that contains at least $n/2^i$ nodes.

By (iii), in Theorem 5.2b for every node u and each $i \in [\log n]$ there is at most a constant number of non-empty sets Y_{uij} (and obviously, there is at least one such set). Part (c) follows immediately.

In both parts of Theorem 5.2, for each node u we sample $\Theta(\log n)$ neighbors (namely, the X-type neighbors) uniformly at random from each of the balls $\{B_{ui},\, i \in [\log n]\}$. Here a given node v is selected with probability $\Theta(\log n)/|B_u(d_{uv})|$, which by (ii) is $\Theta(\log n)/x_{uv}$.

Apart from that, we sample $\Theta(\log n)$ neighbors (namely, the Y-type neighbors) from each of the balls $\{B_u(2^j),\, j \in [\log \Delta]\}$. By (ii) we sample them uniformly at random; by (iii) this boosts the probability of selecting a given node by at most a constant factor. So again, a given v is selected as a Y-type neighbor of u with probability $\Theta(\log n)/x_{uv}$. □

5.3 Comparison with the single-link-per-node model

Let us briefly comment on an alternative setting where we are given a graph of local contacts, and we add exactly one long-range contact per node. This has been the original Kleinberg's model [30] (for two-dimensional grids). Recently, following the publication of the conference version of this paper, such setting has been considered for graphs that induce metrics of low grid dimension [16, 19], graphs of bounded treewidth [17], and graphs that exclude a fixed minor [2].

[16] For Theorem 5.2b we ignore Z-type neighbors since it turns out that on UL-constrained metrics they never get used.

We note in passing that our more straightforward result on small worlds (the one that only uses the Y-type neighbors) trivially extends to this setting:

Theorem 5.5. *Consider a graph G such that its shortest paths metric d_G has doubling dimension α. There is a randomized algorithm that assigns to every node exactly one long-range contact so that in the resulting small-world model on d_G the greedy algorithm completes each query in $2^{O(\alpha)}(\log^2 \Delta)$ hops with high probability and in expectation.*

Proof. We will use, implicitly, $(\log \Delta)$ rings of neighbors so that the radii of the rings grow exponentially, and the neighbors are distributed with respect to the doubling measure. Specifically, for each node u we choose u.a.r. an integer $j \in [\log \Delta]$, and then we select the one long-range contact of u from the ball $B = B_u(2^j)$ according to the probability distribution $\mu(\cdot)/\mu(B)$, where μ is a doubling measure on d_G.

Suppose u is the current node and t is the target. Then with probability $p = \left(2^{O(\alpha)} \log \Delta\right)^{-1}$ node u has a long-range contact within distance $d_{ut}/2$ from t. At every step the greedy algorithm is guaranteed some progress via the local contacts. Eventually it will find a suitable long-range contact and halve the distance to target. This will take $(1/p)$ steps in expectation, and, by Chernoff Bounds, $O(1/p)$ steps with high probability. Therefore the query will complete in $(p^{-1} \log \Delta)$ steps in expectation, and in $O(p^{-1} \log \Delta)$ steps with high probability. □

Recall that Theorem 5.2 explored the interesting trade-off between the out-degree and the hop-count. Here, in Theorem 5.5, in order to make progress, a success event at any one node suffices; so if we allow larger out-degree, then the product of hop-count and out-degree stays constant. This seems a good way to capture the above-mentioned tradeoff. Unfortunately, it does not seem to work in general. For instance, if we adapt Theorem 5.2(a) or Theorem 5.2(b) to the current setting then in order to make progress we need success events at two (resp. three) *consecutive* nodes. This results in poor probability of making progress at a given node, and, accordingly, in an unreasonably poor expected hop-count, as compared to a much less sophisticated Theorem 5.5. These considerations suggest that the setting with one long-range contact per node might not quite capture the richer setting of polylog out-degree.

6 Conclusions and open questions

We consider four related node-labeling problems: low-stretch routing schemes, distance labeling, searchable small worlds, and triangulation-based distance estimation. Our results on these problems are unified by a common technique called 'rings of neighbors'; they are further intertwined as shown in Figure 1 (see Section 1). For each of the four problems, we focus on doubling graphs and improve over the existing constructions. In particular, we obtain approximate distance labeling schemes that are optimal up to constant factors for doubling metrics with super-polynomial aspect ratio.[17] We also extend Kleinberg's small world model to doubling metrics, and obtain simpler proofs for the main result in Chan et al. [14] (on routing schemes) and for a result in Mendel and Har-Peled [43] on distance labeling.

Let us suggest several directions in which our results can be extended.

First, for routing schemes on graphs and for searchable small-world networks it is desirable to further alleviate the dependency on the aspect ratio Δ, e.g. by replacing the $(\log \Delta)$ factor by $(\log n)(\log \log \Delta)$ like we did for distance labeling schemes and routing schemes on metrics. A more ambitious task is to obtain poly-log(n) upper bounds that do not depend Δ altogether. After the conference version of this paper has appeared, such results for routing schemes have been obtained by Abraham et al. [3].

[17] An optimal construction for polynomially-bounded aspect ratio has appeared in Talwar [50].

Second, recall that our result on $(0, \delta)$-triangulation achieves order $O_{\alpha,\delta}(\log n)$. However, the lower bound (1) on distance labeling (see Section 3), which is the only lower bound for triangulation that we have, does not preclude triangulations of order $O_{\alpha,\delta}(1)$ for polynomially bounded aspect ratio, and triangulations of order $O_{\alpha,\delta}(\log \log n)$ otherwise. Can we provide doubling metrics with a triangulation-specific lower bound of $\Omega(\log n)$, or, alternatively, construct triangulations of sub-logarithmic order? Intuitively, the latter would be very surprising. Indeed, consider balls around a given node u. Then there are $\Omega(\log n)$ exponentially increasing size scales, and at least as many exponentially increasing distance scales. If the size scales are roughly aligned with the distance scales, then, intuitively, a label of u in any reasonable triangulation should include distances to at least one node in each of these scales.

Third, we would like extend our results on all four problems to *decomposable metrics* [35], a wide class of metrics that includes doubling metrics as well as the shortest-path metrics of graphs excluding a fixed minor, e.g. shortest-path metrics of planar graphs. This direction seems promising since similar extensions (from doubling metrics to decomposable metrics) have been obtained in [35, 1] in the context of metric embeddings. Also, recent results of Abraham et al. [5, 2] construct low-stretch routing schemes, distance labeling schemes, and small-world networks for graphs excluding a fixed minor.

Finally, 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 Slivkins [49] for distributed approaches to metric embeddings and distance estimation, and practically in *Meridian* (Wong et al. [55]), a system for nearest-neighbor and multi-range queries in a peer-to-peer network. While this framework has already lead to significant results, rings of neighbors that we can define theoretically provide a much better coverage than the ones that we know how to construct and maintain in a distributed fashion. Bridging this gap is an interesting open question. Finally, 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 Slivkins [49] for distributed approaches to metric embeddings and distance estimation, and practically in *Meridian* (Wong et al. [55]), a system for nearest-neighbor and multi-range queries in a peer-to-peer network. While this framework has already lead to significant results, rings of neighbors that we can define theoretically provide a much better coverage than the ones that we know how to construct and maintain in a distributed fashion. Bridging this gap is an interesting open question.

Acknowledgments. I am very grateful to the anonymous referees for their careful reading of the manuscript. I would also like to thank Jon Kleinberg and Mark Sandler for useful discussions and help with the initial drafts of the paper.

References

[1] I. Abraham, Y. Bartal, H. T.-H. Chan, K. Dhamdhere, A. Gupta, J. Kleinberg, O. Neiman, and A. Slivkins. Metric Embeddings with Relaxed Guarantees. In *46th Symp. on Foundations of Computer Science (FOCS)*, pages 83–100, 2005.

[2] I. Abraham and C. Gavoille. Object location using path separators. In *25th Annual ACM SIGACT-SIGOPS Symp. on Principles Of Distributed Computing (PODC)*, 2006.

[3] I. Abraham, C. Gavoille, A. Goldberg, and D. Malkhi. Routing in networks with low doubling dimension. In *26th IEEE Intl. Conf. on Distributed Computing Systems (ICDCS)*, 2006.

[4] I. Abraham, C. Gavoille, and D. Malkhi. Routing with improved communication-space trade-off. In *18th Annual Conf. on Distributed Computing (DISC)*, pages 305–319, 2004.

[5] I. Abraham, C. Gavoille, and D. Malkhi. Compact routing for graphs excluding a fixed minor. In *19th Annual Conf. on Distributed Computing (DISC)*, pages 442–456, 2005.

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

[7] I. Abraham and D. Malkhi. Compact routing on Euclidean metrics. In *23rd Annual ACM SIGACT-SIGOPS Symp. on Principles Of Distributed Computing (PODC)*, pages 141–149, 2004.

[8] I. Abraham and D. Malkhi. Name independent routing for growth bounded networks. In *17th ACM Symp. on Parallel Algorithms and Architectures (SPAA)*, pages 49–55, 2005.

[9] I. Abraham, D. Malkhi, and O. Dobzinski. LAND: Stretch $(1+\epsilon)$ locality-aware networks for DHTs. In *15th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 550–559, 2004.

[10] P. Assouad. Plongements lipschitziens dans $\mathbf{R}^n$. *Bull. Soc. Math. France*, 111(4):429–448, 1983.

[11] B. Awerbuch, A. Bar-Noy, N. Linial, and D. Peleg. Improved routing strategies with succinct tables. *J. of Algorithms*, 11(3):307–341, 1990.

[12] B. Awerbuch, A. Goldberg, M. Luby, and S. Plotkin. Network decomposition and locality in distributed computation. In *30th Symp. on Foundations of Computer Science (FOCS)*, pages 364–369, 1989.

[13] B. Awerbuch and D. Peleg. Sparse partitions. In *31st Symp. on Foundations of Computer Science (FOCS)*, pages 503–513, 1990.

[14] H. T.-H. Chan, A. Gupta, B. M. Maggs, and S. Zhou. On hierarchical routing in bounded-growth metrics. In *16th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 762–771, 2005. Full and updated version available as a Carnegie Mellon University ETR CMU-PDL-04-106.

[15] P. S. Dodds, R. Muhamad, and D. J. Watts. An experimental study of search in global social networks. *Science*, 301:827–829, 2003.

[16] P. Duchon, N. Hanusse, E. Lebhar, and N. Schabanel. Could any graph be turned into a small world? In *19th Annual Conf. on Distributed Computing (DISC)*, pages 511–513, 2005. Full version to appear in *Theoretical Computer Science*, special issue on complex networks, 2006.

[17] P. Fraigniaud. A new perspective on the small-world phenomenon: greedy routing in tree-decomposed graphs. In *13th Annual European Symp. on Algorithms (ESA)*, pages 791–802, 2005.

[18] P. Fraigniaud, C. Gavoille, and C. Paul. Eclecticism shrinks even small worlds. In *23rd Annual ACM SIGACT-SIGOPS Symp. on Principles Of Distributed Computing (PODC)*, pages 169–178, 2004.

[19] P. Fraigniaud, E. Lebhar, and Z. Lotker. A doubling dimension threshold $\Theta(\log\log n)$ for augmented graph navigability. In *14th Annual European Symp. on Algorithms (ESA)*, 2006.

[20] 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 Transactions on Networking*, 9:525–540, October 2001. Preliminary version in IEEE INFOCOM 1999.

[21] C. Gavoille and M. Gengler. Space-efficiency for routing schemes of stretch factor three. *Journal of Parallel and Distributed Computing*, 61(5):679–687, 2001.

[22] C. Gavoille, M. Katz, N. A. Katz, C. Paul, and D. Peleg. Approximate distance labeling schemes. In *9th Annual European Symp. on Algorithms (ESA)*, pages 476–487, 2001.

[23] C. Gavoille and D. Peleg. Compact and localized distributed data structures. *J. of Distributed Computing*, 16:111–120, 2003.

[24] C. Gavoille, D. Peleg, S. Perennes, and R. Raz. Distance labeling in graphs. *J. of Algorithms*, 53(1):85–112, 2004. (Preliminary version in *12th ACM-SIAM SODA*, 2001).

[25] A. Gupta, R. Krauthgamer, and J. R. Lee. Bounded geometries, fractals, and low–distortion embeddings. In *44th Symp. on Foundations of Computer Science (FOCS)*, pages 534–543, 2003.

[26] J. Guyton and M. Schwartz. Locating nearby copies of replicated Internet servers. In *ACM SIGCOMM*, 1995.

[27] K. Hildrum, J. Kubiatowicz, and S. Rao. Object location in realistic networks. In *16th ACM Symp. on Parallel Algorithms and Architectures (SPAA)*, pages 25–35, 2004.

[28] S. M. Hotz. *Routing information organization to support scalable interdomain routing with heterogeneous path requirements*. PhD thesis, University of Southern California, 1994.

[29] J. Kleinberg. Navigation in a small world. *Nature*, 406:485, 2000.

[30] J. Kleinberg. The small-world phenomenon: an algorithmic perspective. In *32nd ACM Symp. on Theory of Computing (STOC)*, pages 163–170, 2000.

[31] J. Kleinberg. Small-world phenomena and the dynamics of information. In *15th Advances in Neural Information Processing Systems (NIPS)*, pages 431–438, 2001.

[32] J. Kleinberg. Complex networks and decentralized search algorithms. In *International Congress of Mathematicians*, 2006.

[33] J. Kleinberg, A. Slivkins, and T. Wexler. Triangulation and Embedding Using Small Sets of Beacons. In *45th Symp. on Foundations of Computer Science (FOCS)*, pages 444–453, 2004.

[34] C. Kommareddy, N. Shankar, and B. Bhattacharjee. Finding close friends on the Internet. In *12th IEEE Intl. Conf. on Network Protocols (ICNP)*, 2001.

[35] R. Krauthgamer, J. Lee, M. Mendel, and A. Naor. Measured descent: A new embedding method for finite metrics. *Geometric and Functional Analysis (GAFA)*, 15(4):839–858, 2005. Preliminary version in *45th IEEE FOCS*, 2004.

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

[37] E. Lebhar and N. Schabanel. Close to optimal decentralized routing in long-range contact networks. In *31st Intl. Colloquium on Automata, Languages and Programming (ICALP)*, pages 894–905, 2004.

[38] J. Luukkainen and E. Saksman. Every complete doubling metric space carries a doubling measure. *Proc. Amer. Math. Soc.*, 126(2):531534, 1998.

[39] G. Manku, M. Bawa, and P. Raghavan. Symphony: Distributed hashing in a small world. In *4th USENIX Symp. on Internet Technologies and Systems (USITS)*, 2003.

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

[41] C. Martel and V. Nguyen. Analyzing Kleinberg's (and other) small-world models. In *23rd Annual ACM SIGACT-SIGOPS Symp. on Principles Of Distributed Computing (PODC)*, pages 179–188, 2004.

[42] C. Martel and V. Nguyen. Analyzing and characterizing small-world graphs. In *16th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 311–320, 2005.

[43] M. Mendel and S. Har-Peled. Fast construction of nets in low dimensional metrics, and their applications. In *21st ACM Symp. on Computational Geometry (SoCG)*, pages 150–158, 2005.

[44] S. Milgram. The small world problem. *Psychology Today*, 2:60–67, 1967.

[45] D. Peleg. Proximity-preserving labeling schemes and their applications. In *The 25th Intl. Workshop on Graph-Theoretical Concepts in Computer Science (LNCS 1665)*, pages 30–41, 1999.

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

[47] D. Peleg and E. Upfal. A trade-off between space and effciency for routing tables. *J. of the ACM*, 36(3):510–530, 1989. (Preliminary version in *20th ACM STOC*, 1988).

[48] C. G. Plaxton, R. Rajaraman, and A. W. Richa. Accessing nearby copies of replicated objects in a distributed environment. *Theory Comput. Syst.*, 32(3):241–280, 1999. Preliminary version in *9th ACM SPAA*, 1997.

[49] A. Slivkins. Distributed Approaches to Triangulation and Embedding. In *16th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 640–649, Jan. 2005.

[50] K. Talwar. Bypassing the embedding: approximation schemes and compact representations for growth restricted metrics. In *36th ACM Symp. on Theory of Computing (STOC)*, pages 281–290, 2004.

[51] M. Thorup and U. Zwick. Compact routing schemes. In *13th ACM Symp. on Parallel Algorithms and Architectures (SPAA)*, pages 1–10, 2001.

[52] M. Thorup and U. Zwick. Approximate distance oracles. *J. of the ACM*, 52(1):1–24, 2005. (Preliminary version in *33rd ACM STOC*, 2001).

[53] A. L. Volberg and S. V. Konyagin. On measures with the doubling condition. *Izv. Acad. Nauk SSSR*, 51:666–675, 1987. In Russian; English transl. Math. USSR Izv. 30 (1988), 629-638.

[54] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–42, 1998.

[55] B. Wong, A. Slivkins, and E. G. Sirer. Meridian: A Lightweight Network Location Service without Virtual Coordinates. In *ACM SIGCOMM*, 2005.

[56] J.-M. Wu. Hausdorff dimension and doubling measures on metric spaces. *Proc. Amer. Math. Soc.*, 126(5):14531459, 1998.

Appendix A: Existence of (ϵ, μ)-packings

We prove Lemma 3.1 on the existence of (ϵ, μ)-packings which is implicit (but never articulated) in Section 5 of [49]. We state it in a slightly stronger form that will be used in the proof of Theorem 4.2.

Lemma A.1. *Consider a finite 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 there exists a ball $B_v(r) \in \mathcal{F}$ such that $d_{uv} + r \leq 6r_u(\epsilon)$. Moreover, such $\mathcal{F}$ can be efficiently computed.*

Proof. Let $r_u = r_u(\epsilon)$. For a given node u, say a ball $B_v(r)$ is *u-zooming* if it is a subset of $B_u(3r_u)$, has measure at least $\epsilon/16^\alpha$, and $B_v(4r)$ has measure at most ϵ. We claim that for every node u either there exists a u-zooming ball, or there exists a node $b_u \in B_u(2r_u)$ of measure at least ϵ.

Suppose neither of the two exists. Let $r = r_u$. By the doubling property of the metric (see Lemma 1.1), $B_u(r)$ can be covered by 16^α balls of radius $r/8$. At least one of these balls, say $B_v(r/8)$, has measure at least $\epsilon/16^\alpha$; since without loss of generality $B_v(r/8)$ overlaps with $B_u(r)$, it follows that $d_{uv} \leq \frac{9}{8}r$ and $B_v(r/2) \subset B_u(2r)$. Since there is no u-zooming ball, in particular the ball $B_v(r/8)$ is not u-zooming, so $B_v(r/2)$ has measure at least ϵ.

The argument applied to $B_u(r)$ can now be applied to $B_v(r/2)$ and so forth. Iterating this argument i times, we come up with a node v such that $d_{uv} \leq \frac{9}{8}r(2 - 2^{-i})$ and $B_v(r/2^i)$ has cardinality at least ϵ. For large enough i, namely for i such that $r/2^i < 1$, this ball consists of only one node, which therefore has measure at least ϵ. Contradiction; claim proved.

In accordance with the above claim, for every given node u we define B_u to be a u-zooming ball if such ball exists, or else we define $B_u = \{b_u\}$ where b_u is a node in $B_u(2r_u)$ that has measure at least ϵ. Note that a suitable B_u can be efficiently computed by simply checking each ball whether it is u-zooming, and then checking each node in $B_u(2r_u)$.

Let $\mathcal{F}$ be a maximal collection of disjoint balls B_u. Note that such $\mathcal{F}$ can be efficiently computed by consecutively going through all balls B_u, and including a given B_u in $\mathcal{F}$ if it is disjoint with other balls that are already in $\mathcal{F}$. We will show that $\mathcal{F}$ is the desired (ϵ, μ)-packing. It suffices to prove the following claim: for each node v some ball $B_u \in \mathcal{F}$ lies within $B_v(6r_v)$.

Suppose that for a given v the claim is false. Since by definition of a v-zooming ball $B_v \subset B_v(3r_v)$, it follows that $B_v \not\in \mathcal{F}$. Since $\mathcal{F}$ is maximal, B_v overlaps with some ball $B_u \in \mathcal{F}$. If $B_u = \{b_u\}$ then it trivially lies in $B_v(3r_v)$, contradiction. So B_u is a u-zooming ball; say w is its center, and r is its radius. By definition of a u-zooming ball, $B_w(4r)$ has measure at most ϵ. If $4r \geq d_{vw} + r_v$, then ball $B_w(4r)$ contains ball $B_v(r_v)$; as the latter ball has measure at least ϵ, the two balls coincide, and thus B_u lies in $B_v(r_v)$, contradiction. Therefore $4r < d_{vw} + r_v$.

Recall that ball B_u overlaps with ball B_v; let x be a node that lies in both balls. Since $B_v \subset B_v(3r_v)$, applying triangle inequality to the triple (v, x, w) we get $d_{vw} \leq 3r_v + r$. Plugging this into the previous inequality, we obtain $3r < 4r_v$. It follows that $r + d_{vw} < 6r_v$. Consequently, ball $B_u = B_w(r)$ lies in the ball $B_v(6r_v)$, contradiction. Claim proved. □

The above proof actually extends to complete infinite metrics, but we do not need it here.

Appendix B: Proof of Theorem 4.2 on routing schemes

We will prove Theorem 4.2 in the following more general form:

Theorem B.1. *Suppose for some $\delta \in (0,1)$ any two nodes in the input graph G are connected by a $(1+\delta)$-stretch path with at most N_δ hops. Let α be the doubling dimension, let Δ be the aspect ratio, and let D_{out} be the out-degree of G. Then there exists a $(1+\delta)$-stretch routing scheme on G with*

- *$O(\alpha\phi\log n) + N_\delta(\log D_{\mathit{out}})$-bit packet headers and*
- *$(\frac{1}{\delta})^{O(\alpha)}(\phi + N_\delta)(\log n)(\log D_{\mathit{out}})$-bit routing tables,*

where $\phi = \log(\frac{1}{\delta}\log\Delta)$. Such routing scheme can be efficiently computed.

We will combine the ideas of Theorem 3.4 and Theorem 2.1 with some new tricks. We will use (i) the basic rings of neighbors, (ii) zooming sequences and intermediate targets, (iii) the first-hop pointers, and (iv) host/virtual enumerations. Our basic setup is from the proof of Theorem 3.4. For simplicity let's assume $\delta \leq 1/8$ and let $\delta' = \delta/(1-\delta)$.

NOTATION. We borrow a lot of definitions from the previous proofs:

- From Theorem 3.2, we borrow, firstly, radii r_{ui} and balls B_{ui}; secondly, $(2^{-i},\mu)$-packings $\mathcal{F}_i$ and sets X_{ui} of X_i-neighbors; and thirdly, 2^j-nets $\mathcal{G}_j$ and sets Y_{ui} of Y_i-neighbors.
- From Theorem 3.4, we borrow the zooming sequences $f_u = \{f_{ui} : i \in [\log n]\}$, the sets T_u of virtual neighbors; host enumerations $\varphi_u(\cdot)$, virtual enumerations $\psi_u(\cdot)$, and translation functions ζ_{ui}. For convenience, we set $\psi_u(v) = \mathit{null}$ whenever v is not a virtual neighbor of u.
- From Theorem 2.1 we borrow the first-hop pointers.

We use $(2^{-i},\mu)$-packings $\mathcal{F}_i$ in a somewhat stronger form provided by Lemma A.1; for each $B \in \mathcal{F}_i$, let $h = h_B$ be a node and $r = r_B$ be a radius such that $B = B_h(r)$ and $6r_u(2^{-i}) \geq d_{uh} + r$. We need to fix h because B can have multiple centers, i.e. nodes v such that $B = B_v(r)$ for some r, whereas Lemma A.1 guarantees this inequality only for one of them. We redefine the set X_{ui} of X_i-neighbors of u as follows as the set of all nodes $h = h_B$ such that $B \in F_i$ and $r_{(u,i-1)} \geq d_{uh} + r_B$.

We introduce some new notation. For each node t, each $i \in [\log n]$ and each $j \in [\log\Delta]$, we define:

- ID(t) as a unique global $\lceil\log n\rceil$-bit identifier for t;
- x_{ti} as the nearest X_i-neighbor of t;
- y_{tj} as the nearest Y_j-neighbor of t;
- J_{ti} as the set of all integers between $\lfloor\log(\frac{\delta}{4}r_{ti})\rfloor$ and $\lceil\log(6r_{ti})\rceil$;
- S_{ti} as the set of all y_{tj} such that $j \in J_{ti}$.

All nodes x_{ui} and all nodes in all sets S_{ui} are called *friends* of u.

DATA STRUCTURES. Routing labels and routing tables will contain distances between some pairs of nodes. All these distances as stored as a $O(\log\frac{1}{\delta})$-bit mantissa and $\log\log\Delta$-bit exponent. It will be easy to see that this many bits suffice for our purposes; we omit the details and treat the stored distances as exact distances.

The *routing label* of target t contains ID(t) and the information about the zooming sequence and the friends of t, specifically:

- sets J_{ti}, for all i.
- the host enumeration of t for f_{t0}, x_{t0} and all nodes in S_{t0}.
- for each $i \geq 1$, the virtual enumeration of $f_{(t,i-1)}$ for f_{ti}, x_{ti} and all nodes in S_{ti}.
- the distances from t to all f_{ti}, all x_{ti} and all nodes in S_{ti}.

In the routing label, the info about all nodes f_{ti} and x_{ti} is stored as an array indexed by i; similarly, the info about all nodes $y_{tj} \in S_{ti}$ is stored as an array indexed by j. The global IDs are not used.

The *routing table* of each node u includes:

- its label, radii r_{ui} for all i, and distances to all its neighbors (but not to its virtual neighbors),

- translation maps ζ_{ui}, for all $i \in [\log n]$.
- the first-hop pointer from u to each neighbor of u, which we can store using only $\lceil \log D_{\text{out}} \rceil$ bits.

Node u does not know the global IDs of its neighbors; they are indexed according to φ_u.

USING THE DATA STRUCTURES. Suppose t is the target and u is the current node. Say node w is a (u, i, j)-*landmark* if the following three conditions hold:

(c1) w is a neighbor of u and a virtual neighbor of $f_{(t,i-1)}$.
(c2) if $j = \infty$ then $w = x_{ti} \in X_{ui}$; else $j \in J_{ui}$ and $w = y_{tj} \in Y_{ui}$.
(c3) for all $l \leq i - 1$ node f_{tl} is a neighbor of u;

Say node w is (u, i, j)-*good* if conditions (c1)-(c3) hold and, moreover,

(c4) $d_{wt} \leq \delta' d_{uw}$ and $6 r_{ui} \leq \delta' d_{uw}$ and $j \geq \lfloor \log \frac{\delta}{1+\delta}\, d_{uw} \rfloor$.
(c5) $r_{ui} < 2\beta d_{uw} \leq r_{(u,i-1)}$ for some β such that $1 - \delta' \leq \beta < 1/(1-\delta)$.

Say a node is u-*good* if it is (u, i, j)-good for some pair (i, j). Note that by condition (c2) a (u, i, j)-landmark is unique if it exists, whereas there could be multiple u-good nodes.

Here is the meaning behind these definitions. A current node u in the routing can select some u-good node w as an intermediate target; the definition is tailored so that, on one hand, a u-good node is a good intermediate target, and on the other hand, we could show that such nodes exist. Then the packet will be routed along some initial segment of a shortest uw-path. In particular, each node v in this segment will know where to forward the packet; essentially, it will be due to the fact that w is a (v, i, j)-landmark.

First we show that (u, i, j)-landmarks and u-good nodes exist, then we show how to identify them. The following claim is an elaboration of the arguments in the proof of Theorem 3.4.

Claim B.2. *Fix any nodes u and t, and let $d = d_{ut}$.*
(a) If $r_{ul} \geq \frac{4}{3} d_{ut}$ for some l then f_{tl} is a Y_l-neighbor of u.
(b) if $\delta d/6 \leq r_{ui} < 2d \leq r_{(u,i-1)}$ for some i, then there exists a u-good node.

Proof: **(a)** Let $d = d_{ut}$. Note that $w = f_{tl} \in \mathcal{G}_j$, $j = \lfloor \log(r_{tl}/4) \rfloor$, and by Claim 3.3 we have $|r_{ul} - r_{tl}| \leq d$. By definition of Y_l-neighbors, we need to check two things: that $d_{uw} \leq 12 r_{ul}/12$ and that $l \in J_{ul}$. Firstly,

$$d_{wt} \leq r_{tl}/4 \leq (r_{ul} + d)/4 < r_{ul}/2, \text{ so } d_{uw} \leq d + d_{wt} < 1.5 r_{ul}.$$

Secondly, $j \in J_{ul}$ follows because $r_{tl} \geq r_{ul} - d \geq r_{ul}(1 - \frac{3}{4}) \geq \delta r_{ul}$.

(b) We will produce a (u, i, j)-landmark w such that $d_{wt} \leq \delta d$. For such w by triangle inequality we have

$$d(1-\delta) \leq d - d_{wt} \leq d_{uw} \leq d + d_{wt} \leq d(1+\delta),$$

so it is easy to see that conditions (c4) and (c5) hold and w is u-good.

If $r_{ti} \leq \delta d/6$ then let $w = x_{ti}$; else let $w = y_{tj}$, $j = \lfloor \log \delta d \rfloor$. In either case, $d_{wt} \leq \delta d$. We claim that w is a (u, i, j)-landmark. Since condition (c3) holds by part (a), we just need to check (c1) and (c2).

Let $x = \delta d$ and $f = f_{(t,\, i-1)}$. There are two cases. Firstly, suppose $r_{ti} \leq x/6$ and $w = x_{ti}$. By definition of X_i-neighbors for some radius r we have $B_w(r) \in \mathcal{F}_i$ and $d_{wt} + r \leq 6 r_{ti} \leq x$. Therefore

$$d_{uw} + r \leq d + d_{wt} + r \leq d + x < 2d \leq r_{(u,\, i-1)},$$

so $w \in X_{ui}$. Since $r_{(t,\, i-1)} \geq r_{(u,\, i-1)} - d \geq d > 12 r_{ti}$, by Claim 3.5a w is a virtual neighbor of f.

Now suppose $r_{ti} > x/6$. Note that $r_{ti} \leq r_{ui} + d < 3d$, so $x \in [\frac{\delta}{4}; r_{ti} 6 r_{ti}]$. Then $w = y_{tj} \in \mathcal{G}_j \cap B_t(z)$ satisfies all conditions in Claim 3.5b, hence is a virtual neighbor of f. Finally, u is a Y_i-neighbor of u since $12 r_{ti}/\delta > 2d > d + d_{wt} \geq d_{uw}$ and $j = \lfloor \log x \rfloor \geq \lfloor \log \delta r_{ti}/4 \rfloor$. □

Claim B.3. *Given the routing table of u and the routing label of t, one can efficiently:*
(a) check whether a u-good node w exists; if so, find $\varphi_u(w)$ and (i, j) such that w is (u, i, j)-good.

(b) check whether the (u,i,j)-landmark w exists, for given (i,j), and find $\varphi_u(w)$ if it does.

Proof: Consider the following algorithm. First, read $\varphi_u(f_{t0})$ from the routing table of u. Then consecutively for each i from 1 to $\lceil \log n \rceil$, let $f = f_{(t,\, i-1)}$, do the following:

1. Note that by construction condition (c3) holds and we know $\varphi_u(f)$.
2. for $w = x_{ui}$ and then consecutively for each $w = y_{uj}$, $j \in J_{ul}$ in the order of decreasing j:
   a. check $\zeta_{ui}\left(\varphi_u(f), \psi_f(w)\right)$. If it is not *null* then it is equal to $\varphi_u(w)$, and condition (c1) holds.
   b. check condition (c2). If it holds, then node w is (u,i,j)-identifiable.
   c. if (c1) and (c2) hold, we can check (c4) and (c5). If they hold, too, then node w is u-good.
3. Check $\zeta_{ui}\left(\varphi_u(f), \psi_f(f_{ti})\right)$. If it is *null* then exit. If it is not *null* then it is equal to $\varphi_u(f_{ti})$.

For part (a) we exit if in step 2b we find a (u,i,j)-identifiable node; for part (b) we exit if in step 2c we find a u-good node. it is easy to see that if a (u,i,j)-identifiable (resp. u-good) node exists, then our algorithm finds and identifies it. □

FIRST ROUTING MODE. The routing will have two *modes*, $\mathcal{M}_1$ and $\mathcal{M}_2$. Routing starts in $\mathcal{M}_1$, then may switch to $\mathcal{M}_2$; if it does, it does not go back to $\mathcal{M}_1$. In what follows, the target node is denoted by t.

The first routing mode is an elaboration of the routing algorithm in the proof of Theorem 2.1. In this mode the packet is routed to an *intermediate target* w, until it reaches w or t, or switches to $\mathcal{M}_2$, or a new intermediate target is chosen. If the current intermediate target w has been chosen at node u, then the packet header contains the routing label of t, the distance $D_{\text{est}} = d_{uw}$, and the *intermediate target id*, which is a pair (i,j) such that w is (u,i,j)-good.

Suppose node u receives a packet. First u checks whether it is the target: if $\text{ID}(t) = \text{ID}(u)$ then we are done. If u is not the target, there are two cases, depending on whether the intermediate target id is *null*.

- If the intermediate target id is *null*, u checks whether a u-good node w exists; if so, u finds $\varphi_u(w)$ and a pair (i,j) such that w is (u,i,j)-good (see Claim B.3a). If u-good nodes do not exist, the routing switches to $\mathcal{M}_2$. Else, u chooses w as the next intermediate target, sets $D_{\text{est}} = d_{uw}$, and sets the intermediate target id to (i,j).

- If the intermediate target id is (i,j), then u checks whether the (u,i,j)-landmark node w exists (see Claim B.3b), finds $\varphi_u(w)$ it if it does, or switches to $\mathcal{M}_2$ if it doesn't.

Suppose the first-hop pointer from u to w denotes edge uv, for some node v. If $d_{uw} - d_{uv} \leq 2\delta' D_{\text{est}}$, or if v is w itself, then u sets the intermediate target id to *null*. Finally, u forwards the packet to v. This completes the description of the first routing mode. For convenience assume that initially the sender receives the packet (from itself) such the intermediate target id is *null*.

We claim that the routing in $\mathcal{M}_1$ is sufficiently nice, namely that the intermediate targets zoom in towards t, and the packet follows shortest paths from one intermediate target to another. We will need a simple application of triangle inequality: for any nodes u, w and t such that $d_{wt} \leq \delta' d$ we have

$$(1-\delta')\, d_{uw} \leq d_{uw} - d_{wt} \leq d_{ut} \leq d_{uw} + d_{wt} \leq (1+\delta')\, d_{uw}.$$

Claim B.4. *Let $u_0, u_1, \ldots, u_{k-1}$ be the nodes where the new intermediate target id has been set; let u_k be the last node that the packet has reached in $\mathcal{M}_1$. Then for a fixed $i < k$ we have:*

(a) the indermediate target w chosen at u_i is at least $\frac{3}{4}\frac{1}{\delta}$ times closer to t then u_i.

(b) u_i is at least $\frac{1}{4}\frac{1}{\delta}$ times closer to t then u_{i-1}.

(c) the packet trajectory from u_i to u_{i+1} is a segment of a shortest (u_i, w_i)-path .

Proof: (a) Let $u = u_i$. Then w is u-good, so $d_{wt} \leq \delta' d_{uw}$ and

$$d_{ut} \geq (1-\delta')d_{wt} \geq (1-\delta')d_{wt}/\delta' = (1-2\delta)d_{wt}/\delta \geq 3d_{wt}/4\delta.$$

(b) Let $v = u_{i+1}$ and suppose $v \neq w$. Let x be the node visited by the packet right before v. Then by definition of $\mathcal{M}_1$ node v lies on a shortest xw-path, and at node x we had $d_{vw} = d_{xw} - d_{xv} \leq 2\delta' D_{\text{est}}$, where $D_{\text{est}} = d_{uw}$. Therefore,

$$d_{vt} \leq d_{vw} + d_{wt} \leq 3\delta' d_{uw} \leq 3\delta' d_{ut}/(1-\delta') = 3\delta d_{ut}/(1-2\delta) \leq 4\delta d_{ut}.$$

(c) The proof is similar to that of Claim 2.4, but somewhat more complicated since u_{i+1} is not necessarily equal to w. Let $u = u_i$ and $v = u_{u+1}$. Let $\rho(x)$ be the path traversed by the packet from node x to v; let $\rho_L(x)$ be the metric length of this path. We need to show that $\rho_L(u) = d_{uv} = d_{uw} - d_{vw}$.

We claim that for every node $x \in \rho(u)$ we have $\rho_L(x) = d_{xv} = d_{xw} - d_{vw}$. We will use induction on $\rho(x)$. Consider an edge $xy \in \rho(u)$ and assume $\rho_L(y) = d_{yv} = d_{yw} - d_{vw}$. By definition of $\mathcal{M}_1$ node y lies on a shortest xw-path, so $d_{xy} + d_{yw} = d_{xw}$. It follows that

$$d_{xv} + d_{vw} \geq d_{xw} = d_{xy} + d_{yw} = d_{xy} + d_{yv} + d_{vw} \geq d_{xv} + d_{vw},$$

so $\rho_L(x) = d_{xy} + \rho_L(y) = d_{xy} + d_{yv} = d_{xv} = d_{xw} - d_{vw}$. □

SWITCHING BETWEEN THE MODES. It is crucial that the routing switches from $\mathcal{M}_1$ to $\mathcal{M}_2$ only if for the current node a certain condition (Lemma B.5) holds. We will see later that under this condition $\mathcal{M}_2$ work efficiently. The forthcoming Lemma B.5 is really the crux of the proof of Theorem 4.2.

Lemma B.5. *Suppose the routing switches to $\mathcal{M}_2$ at node v. Then $6r_{vi}/\delta < \frac{4}{3}d_{vt} \leq r_{(v,i-1)}$ for some i.*
Proof: Suppose such i does not exist. Let u be the last node that receives the packet in $\mathcal{M}_1$ with null intermediate target id. If $u = v$ then for i such that $r_{ui} < \frac{3}{2}d_{ut} \leq r_{(u,i-1)}$ we must have $6r_{ui} \geq \frac{4}{3}\delta d_{ut}$, so by Claim B.2b there exists a u-good node, contradiction. Therefore $u \neq v$. It follows that:

- the routing did not switch to $\mathcal{M}_2$ at u, so u has set the intermediate target id to a pair (i,j) such that there exists a (u,i,j)-good node w.
- node v received the packet with a non-null intermediate target id (equal to (i,j)), so it must be the case that $d_{xw} - d_{xv} > 2\delta' d_{uw}$, where x is the node visited by the packet immediately before v.

Since the routing switched to $\mathcal{M}_2$ at v, by the specification of $\mathcal{M}_1$ there is no (v,i,j)-landmark node. For the sake of contradiction, We will show that node w is a (v,i,j)-landmark; this will complete the proof of the Lemma.

We need to check conditions (c1-c3) in the definition of a (v,i,j)-landmark. For condition (c3), we claim that for each $l \leq i-1$ we have $f_{tl} \in Y_{vj}$. Indeed, since w is (u,i,j)-good, it follows that $d_{wt} \leq \delta' d_{uw}$ and $r_{ul} \geq 2d_{uw}(1-\delta')$. By Claim B.4c node v lies on a shortest uw-path, so $d_{uv} + d_{vw} = d_{uw}$. Moreover, $r_{vl} \geq r_{ul} - d_{uv}$ by Claim 3.3. Putting this all together and letting $\beta = \frac{4}{3}$, we have:

$$\begin{aligned} r_{ul} &\geq 2d_{uw}(1-\delta') \geq \beta d_{uw}(1+\delta') \geq \beta(d_{uw} + d_{wt}) = \beta(d_{uv} + d_{vw} + d_{wt}) \\ r_{vl} &\geq r_{ul} - d_{uv} \geq r_{ul} - \beta d_{uv} \geq \beta(d_{vw} + d_{wt}) \geq \beta d_{vt}, \end{aligned}$$

so the claim follows by Claim B.2a.

Since w is (u,i,j)-good, it is a virtual neighbor of $f_{(t,\,i-1)}$. Therefore it remains to check condition (c2). To this end, we claim that $w \in X_{vi}$ if $j = \infty$, and $w \in Y_{vi}$ otherwise.

If $j = \infty$ then by definition of (u,i,j)-landmarks $w = x_{ti} \in X_{ui}$, so by definition of X_i-neighbors for some r we have $B_w(r) \in \mathcal{F}_i$ and $r_{(u,i-1)} \geq d_{uw} + r$. It follows that

$$r_{(v,\,i-1)} \geq r_{(u,\,i-1)} - d_{uv} \geq d_{uw} + r - d_{uv} = d_{vw} + r,$$

so w is a X_i-neighbor of v, too.

If $j < \infty$ then by definition of (u,i,j)-landmarks it must be the case that $w = y_{uj} \in \mathcal{G}_j$. We need to show that $w \in Y_{vi}$, i.e. that (a) $d_{vw} \leq 12 r_{vi}/\delta$ and (b) $j \geq \lfloor \delta r_{vi}/4 \rfloor$.

Recall that $\delta \leq 1/8$. Since $\frac{4}{3} d_{vt} \leq r_{(v,i-1)}$ and we assumed that the i in the statement of the Lemma does not exist, it must be the case that $6 r_{vi} \geq \frac{4}{3}\delta d_{vt}$. Therefore:

$$\begin{aligned} d_{vt} &\geq d_{vw} - d_{wt} > 2\delta' d_{uw} - \delta' d_{uw} = \delta' d_{uw} \geq d_{wt}, \\ d_{vw} &\leq d_{vt} + d_{wt} \leq 2 d_{vt} \leq 2(3/4)(6/\delta) r_{vi} = 9 r_{vi}/\delta. \end{aligned}$$

This proves part (a). For part (b) recall that $j \geq \lfloor \log \frac{\delta}{1+\delta}\, d_{uw} \rfloor$ and $r_{ui} \leq 2 d_{uw}/(1-\delta)$ since w is (u,i,j)-good. In particular, it suffices to show that $4 d_{uw} \geq (1+\delta) r_{vi}$. Indeed,

$$r_{vi} \leq d_{uv} + r_{ui} \leq d_{uw} + 2 d_{uw}/(1-\delta) \leq 4 d_{uw}/(1+\delta),$$

claim proved. This completes the proof of the Lemma. □

SECOND ROUTING MODE. Suppose routing switches to $\mathcal{M}_2$ at node u; let $d = d_{ut}$. By Lemma B.5 for some i it is the case that $6 r_{ui}/\delta < \frac{4}{3} d \leq r_{(u,i-1)}$. By Lemma 3.1 there exists a ball $B \in \mathcal{F}_i$ of cardinality at least $n/2^{i+O(\alpha)}$ such that $B \subset B_u(6 r_{ui})$. Let $w = h_B$ be the node selected from B in Theorem 3.2; recall that it is a center of B. It is easy to see that the ball $B' = B_{(w,\,i-1)}$ contains target t. Indeed, $d_{uw} \leq 6 r_{ui} \leq \frac{4}{3}\delta d \leq d/6$ since $\delta \leq \frac{1}{8}$, and by Claim 3.3

$$r_{(w,i-1)} \geq r_{(u,i-1)} - d_{uw} \geq 4d/3 - d/6 \geq d + d_{uw} \geq d_{wt}.$$

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-path tree T_B rooted at h. We label the edges of T_B so that given $\textsc{id}(t)$, $t \in B'$ 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 node v with a range R_v such that if a packet is at u, and edge $uv \in T_B$, and $\textsc{id}(t)$ is within this range, then the packet is forwarded to v.

It is crucial that we are free to choose the ranges R_v to edges of T_B and the mapping v_t from B' to B any way we want. 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 $\textsc{id}(t) \in R_l$.

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-path 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, where D_{out} is the maximal degree of the underlying connectivity graph, so the entire path can be stored using at most $N_\delta \lceil \log D_{\text{out}} \rceil$ bits. Since a given node can lie in only one ball $B \in F_i$, it has to store at most $2^{O(\alpha)}$ paths for each i, for a total of at most $2^{O(\alpha)} \log n$ paths. This completes the second routing mode.

Claim B.6. *If the routing switches to $\mathcal{M}_2$ at node u, then from u to t it has stretch $1 + O(\delta)$.*

PROOF OF CORRECTNESS. The space requirements of both routing modes are summarized in Table 3. We need to show that our routing scheme has stretch $1+O(\delta)$. If the packet reaches the target without switching

	routing table size, bits	packet header size, bits
mode $\mathcal{M}_1$	$(\frac{1}{\delta})^{O(\alpha)}(\phi \log n)(\log D_{\text{out}})$	$O(\alpha\phi \log n)$
mode $\mathcal{M}_2$	$2^{O(\alpha)}(N_\delta \log n)(\log D_{\text{out}})$	$N_\delta \lceil \log D_{\text{out}} \rceil$
total	$(\frac{1}{\delta})^{O(\alpha)}(\phi + N_\delta)(\log n)(\log D_{\text{out}})$	$O(\alpha\phi \log n) + N_\delta \lceil \log D_{\text{out}} \rceil$

Table 3: Space requirements; let $\phi = \log(\frac{1}{\delta} \log \Delta)$.

to $\mathcal{M}_2$, this follows from Claim B.4. Now suppose it switches to $\mathcal{M}_2$ at node w in the middle of a path to some intermediate target v. Let u be the node that set v as the intermediate target and let $d = d_{ut}$. Let ρ_{xy} be the distance traversed by the packet on its path from node x to node y.

By Claim B.6 $\rho_{wt}/d_{wt} \leq 1 + O(\delta)$. By Claim B.4ab $v \in B_t(6\delta d)$. By Claim B.4c, node w lies on some shortest path from u to v, and the packet followed this path from u to w. Putting this together, we get

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

Suppose the packet originated at node s. If $s = u$ then we are done. If $s \neq u$ then by Claim B.4 $\rho_{su} \leq (1 + O(\delta))\, d_{st}$ and by Claim B.4ab $d \leq \delta d_{st}$. Therefore,

$$\rho_{st} = \rho_{su} + \rho_{ut} \leq (1 + O(\delta))\, (d_{st} + d) \leq (1 + O(\delta))\, d_{st},$$

as claimed. This completes the proof of Theorem B.1.