On Hierarchical Routing in Doubling Metrics

Hubert T-H. Chan* Anupam Gupta* Bruce M. Maggs*† Shuheng Zhou†

Abstract

We study the problem of routing in doubling metrics, and show how to perform hierarchical routing in such metrics with small stretch and compact routing tables (i.e., with small amount of routing information stored at each vertex). We say that a metric (X, d) has *doubling dimension* $\dim(X)$ at most α if every set of diameter D can be covered by 2^α sets of diameter $D/2$. (A *doubling metric* is one whose doubling dimension $\dim(X)$ is a constant.) We show how to perform $(1 + \tau)$-stretch routing on metrics for any $0 < \tau \leq 1$ with routing tables of size at most $(\alpha/\tau)^{O(\alpha)} \log^2 \Delta$ bits with only $(\alpha/\tau)^{O(\alpha)} \log \Delta$ *entries*, where Δ is the diameter of the graph; hence the number of routing table entries is just $\tau^{-O(1)} \log \Delta$ for doubling metrics. These results extend and improve on those of Talwar (2004).

We also give better constructions of sparse *spanners* for doubling metrics than those obtained from the routing tables above; for $\tau > 0$, we give algorithms to construct $(1 + \tau)$-stretch spanners for a metric (X, d) with maximum degree at most $(2 + 1/\tau)^{O(\dim(X))}$, matching the results of Das et al. for Euclidean metrics.

1 Introduction

The *doubling dimension* of a metric space (X, d) is the least value α such that each ball of radius R can be covered by at most 2^α balls of radius $R/2$ [13]. For any $\alpha \in \mathbb{Z}$, the space $\mathbb{R}^\alpha$ under any of the ℓ_p norms has doubling dimension $\Theta(\alpha)$, and hence this doubling dimension extends the standard notion of geometric dimension; moreover, it can be seen as a way to parameterize the inherent "complexity" of metrics.

In this paper, we study the problem of designing routing algorithms for networks whose structure is parameterized by the doubling dimension $\dim(X) = \alpha$; we show that one can route along paths with stretch $(1 + \tau)$ with small routing tables—with only $O((\alpha/\tau)^{O(\alpha)} \log \Delta)$ *entries*, where Δ is the diameter of the network. Each entry stores at most $O(\log n + \log \Delta)$ bits, and hence for doubling metrics—where α is a constant—and any $\tau \leq 1$, we have $(1 + \tau)$-stretch routing with only $O(\log^2 \Delta)$ bits of routing information at each node.

The idea of placing restrictions on the growth rate of networks to bound their "intrinsic complexity" is by no means novel; it has been around for a long time (see, e.g., [18]), and has recently been used in several contexts in the literature on object location in peer-to-peer networks [23, 17, 16]. While these papers used definitions and restrictions that differ slightly from each other, we note that our results hold in those models as well. Our results extend those of Talwar [25], whose routing schemes for metrics with $\dim(X) = \alpha$ require local routing information of $\approx O(\log^\alpha \Delta)$ bits. Formally, we have the following main result.

THEOREM 1.1. *Given any network G inducing a metric (X, d) with $\dim(X) = \alpha$ and any $\tau > 0$, there is a routing scheme on G that achieves $(1 + \tau)$-stretch and where each node stores only $(\frac{\alpha}{\tau})^{O(\alpha)} \log^2 \Delta$ bits of routing information.*

The proof of the theorem proceeds along familiar lines; we construct a set of hierarchical decompositions (HDs) of the metric (X, d), where each HD consists of a set of successively finer partitions of X with geometrically decreasing diameters. Each node in X maintains a table containing next hops to a small subset of clusters in these partitions; to route a packet from s to t, we use the routing table for s to pick some "small cluster" C in s' table that contains t and send the packet to some node x in C; a similar process repeats at node $x \in C$ until the packet reaches t. The idea is to create routing tables which ensure that the distance from x to t is much smaller than that from s to t, and hence the detour taken in going from s to t is only $\tau d(s, t)$. (Details of routing schemes appear in Section 4.)

While this framework is well-known, the standard ways to construct HDs are top-down methods which iteratively refine partitions. These methods create long-range dependencies which require us to build $O(\log n)$ HDs in general; in order to use the locality of the doubling metrics and get away with $\tilde{O}(\alpha)$ HDs, we develop a bottom-up approach that avoids these dependencies when building HDs. The analy-

*Computer Science Department, Carnegie Mellon University, Pittsburgh PA 15213. {hubert, anupamg, bmm}@cs.cmu.edu. Bruce Maggs is supported in part by NSF Award CNF–0435382, NSF Award CNF–0433540, NSF ITR Award ANI–0331653, NSF ITR Award CCR–0205523, and US ARO Award DAAD19-02-1-0389.

†Electrical and Computer Engineering Department, Carnegie Mellon University, Pittsburgh PA 15213. szhou@ece.cmu.edu. We thank the members and companies of the PDL Consortium (including EMC, Engenio, Hewlett-Packard, HGST, Hitachi, IBM, Intel, Microsoft, Network Appliance, Oracle, Panasas, Seagate, Sun, and Veritas) for their interest, insights, feedback, and support. This material is based on research sponsored in part by the Army Research Office, under agreement number DAAD19–02–1–0389.

sis of this process uses the Lovász Local Lemma (much as in [19, 13]); details are given in Section 3.

Apart from the above result on low-stretch routing, the proof of Theorem 1.1 can be used to infer the existence of linear-sized *spanners* for doubling metrics, i.e., subgraphs with only $O_{\tau,\alpha}(n)$ edges that maintain distances to be within a factor of $(1+\tau)$. We further give simpler and tighter constructions of spanners, extending similar results of Das et al. [7] for Euclidean metrics.

THEOREM 1.2. *Every metric (X, d) has a $(1+\tau)$-spanner H where the degree of each vertex is at most $(2 + 1/\tau)^{O(\dim(X))}$; hence H has a linear number of edges for any constant τ and $\dim(X)$.*

1.1 Related Work Distributed packet routing protocols have been widely studied in the theoretical computer science community; see, e.g., [9, 10, 3, 21, 6, 22], or the survey by Gavoille [11] on some of the issues and techniques. Note that these results, however, are usually for general networks, or for networks with some topological structure. By placing restrictions on the doubling dimension, we are able to give results which degrade gracefully as the "complexity" of the metric increases. For example, it is known that any universal routing algorithm with stretch less than 3 requires *some* node to store at least $\Omega(n)$ routing information [12]; however, these graphs generate metrics with large $\dim(X)$. Our results thus allow one to circumvent these lower bounds for metrics of "lower dimension".

Packet routing in low dimensional networks has been previously studied in Talwar [25], that gives algorithms that require $O(\alpha(\frac{6}{\tau\alpha})^\alpha(\log^{\alpha+2}\Delta))$ bits of information to be stored per node in order to achieve $(1+\tau)$-stretch routing—for constant stretch τ and doubling dimension α. The resulting dependence of $O(\log^{2+\alpha}\Delta)$ should be contrasted with the dependence of $O(\log^2\Delta)$ bits of information in our schemes. We should point out that his algorithms are based on graph decomposition ideas with a top-down approach and do not require the LLL to construct routing tables.

One of the papers that influence this work is that of Kleinrock and Kamoun [18]. They describe a general hierarchical clustering model on which our routing schemes are based. They show that routing schemes based on a hierarchical clustering model do not cause much increase in the *average path length* for networks that satisfy the following two assumptions: (a) the diameter of any cluster S chosen is bounded above by $O(|S|^\nu)$ for some constant $\nu \in [0,1]$, and (b) the average distance between nodes in the network is $\Theta(n^\nu)$. In contrast, we give bounds on the path stretch on a *per node-pair* level using slightly different assumptions on the network geometry.

Other papers on object location in peer-to-peer networks [23, 17, 16] have also used restrictions similar to [18] on the growth rate of metrics; in particular, they consider metrics where increasing the radius of any ball by a factor of 2 causes the number of points in it to increase by at most some constant factor 2^β. (Plaxton et al. [23] also consider the *lower bound* on the growth.) Here the parameter β can be considered to be another notion of "dimension" for a metric space. It can be shown that $\dim(X) \leq 4\beta$ [13, Prop. 1.2]; hence our results hold for such metrics as well. Our scheme is also similar in spirit to a data-tracking scheme of Rajaraman et al. [24], who use approximations by tree distributions to obtain bounds on the stretch incurred.

Finally, sparse spanners have been studied widely, having found applications in network algorithms (see, e.g., [22]), since they allow us to store information about the metric compactly. Our work extends the results of Arya et al. [2] and Das et al. [7] who have show the existence of $(1+\tau)$-spanners for $\mathbb{R}^\alpha$ with $O_{\tau,\alpha}(n)$ edges. Independent of our work, Har-Peled and Mendel [14] have also obtained, among many other results, constructions of sparse spanners for doubling metrics; they also give linear-time procedures to find these constructions.

2 Definitions and Notation

Let the input metric be (X, d); this paper deals with finite metrics with at least 2 points. We use standard terminology from the theory of metric spaces; many definitions can be found in [8] and [15]. Given $x \in X$ and $r \geq 0$, we let $\mathbf{B}(x, r)$ denote $\{x' \in X \mid d(x, x') \leq r\}$, i.e., the ball of radius r around x. Given a subset $S \subseteq X$, the distance of $x \in X$ to the set S is $d(x, S) = \min\{d(x, x') \mid x' \in S\}$.

The *doubling constant* λ_X of a metric space (X, d) is the smallest value λ such that every ball in X can be covered by λ balls of half the radius. The *doubling dimension* of X is then defined as $\dim(X) = \log_2 \lambda_X$; we use the letter α to denote $\dim(X)$. A metric is called *doubling* when its doubling dimension is a constant. A subset $Y \subseteq X$ is an *r-net* of X if (1) for every $x, y \in Y, d(x, y) \geq r$ and (2) $X \subseteq \cup_{y\in Y}\mathbf{B}(y, r)$. Such nets always exist for any $r > 0$, and can be found using a greedy algorithm.

PROPOSITION 2.1. (SEE, E.G.,[13]) *If all pairwise distances in a set $Y \subseteq X$ are at least r (e.g., when Y is an r-net of X), then for any point $x \in X$ and radius t, we have that $|\mathbf{B}(x, t) \cap Y| \leq \lambda_X^{\lceil \log_2 \frac{2t}{r} \rceil}$.*

A *cluster* C in the metric (X, d) is just a subset of points of the set X. The diameter of the cluster C is the largest distance between points of the cluster. Each cluster is associated with a *center* $x \in X$ (*which may not lie in* C) and the *radius* of the cluster C is the smallest value r such that the cluster C is contained in $\mathbf{B}(x, r)$.

DEFINITION 2.1. *Given $r > 0$, an* r-**ball partition** Π *of (X, d) is a partition of X into clusters $C_1, C_2, \ldots$, with each cluster C_i having a radius at most r.*

By scaling, let us assume that the smallest inter-point distance in X is exactly 1. Let Δ denote the diameter of the metric (X, d), and hence Δ is also the aspect ratio of the metric. Define $\rho = 256\alpha + 1$ and $h = \lceil \log_\rho \Delta \rceil$. Let us define $\eta_i = 1 + \rho + \rho^2 + \ldots + \rho^i < \rho^{i+1}/(\rho - 1)$; note that $\eta_i = \rho\,\eta_{i-1} + 1$. Let us fix a $\rho^i/2$-net and denote with N_i for the metric (X, d), for every $0 \le i \le h + 1$.

2.1 Hierarchical Decompositions (HDs)

We now give a formal definition of a *hierarchical decomposition* (HD) which is used throughout this paper and is the basic object of our study. As noted below, such a decomposition can be naturally associated with a decomposition tree that is used for our hierarchical routing schemes.

DEFINITION 2.2. *A* ρ-hierarchical decomposition $\mathbf{\Pi}$ *(*ρ*-*HD*) of the metric* (X, d) *is a sequence of partitions* $\Pi_0, \ldots, \Pi_h$ *with* $h = \lceil \log_\rho \Delta \rceil$ *such that:*

1. *The partition* Π_h *has one cluster* X*, the entire set.*
2. **(geometrically decreasing diameters)** *The partition* Π_i *is an* η_i*-ball partition. Since inter-point distances are at least* 1*, it implies that* $\Pi_0 = \{\{x\} : x \in X\}$*; in other words, each cluster in* Π_0 *is a singleton vertex.*
3. **(hierarchical)** Π_i *is a refinement of* Π_{i+1} *and each cluster in* Π_i *is contained within some cluster of* Π_{i+1}.

Given such a ρ-HD $\mathbf{\Pi} = (\Pi_i)_{i=0}^h$, the partition Π_i is called the *level-*i partition of $\mathbf{\Pi}$ and clusters in Π_i are the *level-*i clusters. Note that these clusters have a radius η_i and hence diameter $\le 2\eta_i$. Furthermore, define the *degree* $\deg(\mathbf{\Pi})$ to be the maximum number of level-i clusters contained in any level-$(i+1)$ cluster in Π_{i+1}, for all $0 \le i \le h - 1$.

2.1.1 Hierarchical Decompositions and HSTs

A hierarchical decomposition is a *laminar family* of sets, where given any two sets, they are either disjoint or one contains the other. It is well known that such a family $\mathcal{F}$ of sets over X can be associated with a natural decomposition tree whose vertices are sets in $\mathcal{F}$ and whose leaves are all the smallest sets in the family (which are elements of X, in this case). We can use this to associate a so-called hierarchically well-separated tree (also called an HST [4]) $T_{\mathbf{\Pi}}$ with a hierarchical decomposition $\mathbf{\Pi}$; since each edge in $T_{\mathbf{\Pi}}$ connects some $C \in \Pi_i$ and $C' \in \Pi_{i-1}$ with $C' \subseteq C$, we associate a *length* η_i with edge (C, C'). Given such a tree $T_{\mathbf{\Pi}}$, we can (and indeed do) talk about its level-i clusters with no ambiguity; these are the same level-i clusters in the associated Π_i. Note that the degree of vertices in this tree $T_{\mathbf{\Pi}}$ is bounded by $\deg(\mathbf{\Pi}) + 1$.

2.2 Padded Probabilistic Ball-Partitions

Recall that an r-ball partition Π of (X, d) is a partition of X into a set of clusters $C \subseteq X$, each contained in a ball $\mathbf{B}(v, r)$ for some $v \in X$. $\mathbf{B}(x, t)$ is *cut* in the partition Π if there is no cluster $C \in \Pi$ such that $\mathbf{B}(x, t) \subseteq C$. In general, $\mathbf{B}(x, t)$ is *cut* by a set $S \subseteq X$ if both $S \cap \mathbf{B}(x, t)$ and $\mathbf{B}(x, t) \setminus S$ are non-empty.

Let $\mathcal{P}$ be a collection of all possible partitions of X, and hence $\Pi \in \mathcal{P}$. Given a partition $\Pi \in \mathcal{P}$ and $x \in X$, let $C_\Pi(x)$ be the cluster of Π containing x.

DEFINITION 2.3. ([13]) *An* (r, ε)-padded probabilistic ball-partition *of a metric* (X, d) *is a probability distribution* μ *over* $\mathcal{P}$ *satisfying:*

1. **(bounded radius)** *Each* Π *in the support of* μ *is an* r*-ball partition.*
2. **(padding)** $\forall x \in X$, $\Pr_\mu\left[d(x, X \setminus C_\Pi(x)) \ge \varepsilon r\right] \ge \frac{1}{2}$.

(This is called a padded probabilistic decomposition in [13].) Each cluster C in every partition Π in the support of a probabilistic ball-partition μ has radius at most r; and for any $x \in X$, a random r-ball partition Π drawn from the distribution μ does not cut $\mathbf{B}(x, \varepsilon r)$ (and hence $\mathbf{B}(x, \varepsilon r)$ is contained in cluster $C_\Pi(x) \in \Pi$) with probability $\ge 1/2$.

3 Padded Probabilistic Hierarchical Decompositions

In this section, we define a (ρ, ε)-padded probabilistic hierarchical decomposition (PPHD) of the metric (X, d), on which the routing algorithm is based. A PPHD is a probability distribution over HDs that has a "probabilistic padding" property similar to that in Definition 2.3. For any pair of nodes s, t in X and any ball containing both s and t with a diameter of $\approx d(s, t)$, the PPHD ensures that this ball is contained in a single cluster of radius only slightly ($\approx \alpha$ factor) larger than $d(s, t)$ at a suitable level with probability $\ge \frac{1}{2}$. Thus the shortest s-t path is contained entirely in this cluster of radius not much more than $d(s, t)$. This is the general intuition for PPHDs and the starting point for the routing algorithm.

For our applications, we refine PPHDs so that they consist of only $m = O(\alpha \log \alpha)$ of HDs. We first give an existence proof, using the Lovász Local Lemma (LLL), to show that such decompositions exist in Section 3.1. We then outline a randomized polynomial-time algorithm to find the decompositions using Beck's techniques [5] in Section 3.2.

The existence proof for the PPHDs has the following outline. We first give a randomized algorithm to form a single random hierarchical decomposition $\mathbf{\Pi}$, which proves the existence of PPHDs, albeit with support over an exponential number of HDs. To reduce the size to something that depends only on α, we have to use the locality property of the metric space and the LLL. One significant complication in the proof is that we cannot use the standard top-down decomposition schemes to construct PPHDs, since they have long-range correlations that preclude the application of the LLL. Our solution to this problem is to build the decomposition trees in a bottom-up fashion and to make sure that the coarser partitions respect the cluster boundaries made in the finer partitions.

3.1 Existence of PPHDs Motivated by the routing application, we are interested in finding the following structure, which we call a (ρ, ε)-*padded probabilistic hierarchical decomposition*. This is a probability distribution μ over ρ-hierarchical decompositions (as defined in Definition 2.2) so that given $\mathbf{B}(x, \varepsilon r)$ with $r \approx \rho^i$, if we choose a random ρ-HD $\mathbf{\Pi}$ from μ and examine the partition Π_i in it, $\mathbf{B}(x, r)$ is cut in this partition Π_i with probability at most $\frac{1}{2}$.

DEFINITION 3.1. (**PPHD**) *A* (ρ, ε)-padded probabilistic hierarchical decomposition *(referred to as a* (ρ, ε)*-PPHD) is a distribution* μ *over* ρ*-hierarchical decompositions, such that for any point* $x \in X$ *and any value* r *s.t.* $\rho^{i-1} \le r \le \rho^i$,

$$\Pr_{\mathbf{\Pi} \in \mu}[\mathbf{B}(x, \varepsilon r) \text{ is cut in } \Pi_i] \le \tfrac{1}{2},$$

where the random ρ*-hierarchical decomposition chosen is* $\mathbf{\Pi} = (\Pi_i)_{i=0}^h$. *The* degree *of the PPHD* μ *is defined to be* $\deg(\mu) = \max_{\mathbf{\Pi} \in \mu} \deg(\mathbf{\Pi})$.

Note that the definition of a PPHD extends both the idea of a padded probabilistic ball-partition and that of HDs—we ask for a distribution over entire HDs, instead of over ball-partitions at a certain scale r. However, having picked a random ρ-HD $\mathbf{\Pi} = (\Pi_i)_{i=0}^h$ from this distribution, we demand that balls of radius $\approx \varepsilon\rho^i$ be cut with small probability only in partition Π_i that is "at the correct distance scale". Our main theorem of this section is the following:

THEOREM 3.1. *Given a metric* (X, d)*, there exists a* (ρ, ε)*-PPHD* μ *for* (X, d) *with* $\rho = O(\alpha)$ *and* $\varepsilon = O(1/\alpha)$*. The degree* $\deg(\mu)$ *of the PPHD is at most* $\alpha^{O(\alpha)}$*. Furthermore, there exists a distribution* μ_m *whose support is over only* $m = O(\alpha \log \alpha)$ *HDs.*

Since any hierarchical decomposition $\mathbf{\Pi}$ can be associated with a tree $T_{\mathbf{\Pi}}$ (as mentioned in Section 2.1), the above theorem can be viewed as guaranteeing a set of m trees such that the level-i clusters in half of these trees do not cut a given ball of radius $\approx \varepsilon\rho^i$. This proves the existence of an appropriate *tree cover*.

DEFINITION 3.2. *A stretch-k Steiner tree cover for* (X, d) *is a set of trees* $\mathcal{T} = \{T_1, \ldots, T_m\}$ *(with each tree* T_i *possibly containing Steiner points* $\notin X$*, and edges having lengths), where for every* $x, x' \in X$*, there exists a tree* $T_i \in \mathcal{T}$ *for* (X, d) *such that the (unique shortest) path in* T_i *between* x *and* x' *has length at most* $k\, d(x, x')$.

LEMMA 3.1. *Given a metric* (X, d) *with* $\dim(X) = \alpha$*, there exists a stretch-*$O(\rho/\varepsilon)$ *Steiner tree cover consisting of* $O(\alpha \log \alpha)$ *trees, where each tree has degree at most* $\alpha^{O(\alpha)}$.

We omit the simple proof of the above lemma and the description of how the Steiner points can be removed from the trees without altering distances and degrees. We prove Theorem 3.1 in the rest of this section. We first prove (in Section 3.1.1) that one can obtain the result where the PPHD μ has support over many HDs. We then use the Lovász Local Lemma (in Section 3.1.2) to show that a PPHD distribution μ_m with support over only a small number of HDs exists.

3.1.1 Padded Probabilistic Hierarchical Partitions If we do not care about the number of HDs in the support of a PPHD, the existence result of Theorem 3.1 has been proved earlier [25] with better guarantees; the proof basically follows from the padded decompositions given in [13]. However, we now give another proof that introduces ideas that are ultimately useful in obtaining a PPHD distribution whose support is over a small number of HDs.

THEOREM 3.2. *Given a metric* (X, d)*, there exists a* (ρ, ε)*-PPHD* μ *for* (X, d) *with* $\rho = O(\alpha)$ *and* $\varepsilon = O(1/\alpha)$*, and with degree* $\deg(\mu) = \alpha^{O(\alpha)}$*. Furthermore, one can sample from* μ *in polynomial time.*

Proof. We define a randomized process that builds a random hierarchical decomposition tree in a bottom-up fashion, instead of the usual top-down way. To build a HD $\mathbf{\Pi}$, we start with ($\Pi_0 = \{\{x\} : x \in X\}$) and perform an inductive step. At any step, we are given a partial structure $(\Pi_i, \ldots, \Pi_0)$ where for each $j \le i$, the clusters in Π_{j-1} (which is an η_{j-1}-ball partition) are contained within the clusters of Π_j. We then build a new partition Π_{i+1}, with all clusters of Π_i being contained within clusters of Π_{i+1}. We have to ensure that clusters of Π_{i+1} are contained in balls of radius at most η_{i+1} and that any ball of radius εr for $\rho^i \le r \le \rho^{i+1}$ is cut in Π_{i+1} with probability at most $\frac{1}{2}$. This way, we end up with a valid random HD $\mathbf{\Pi}$. The claimed probability distribution μ is the one naturally generated by this algorithm. To create the clusters of Π_{i+1}, we use a decomposition procedure whose property is summarized in the following lemma.

LEMMA 3.2. *Given a metric* (X, d) *with a* Γ*-ball partition* Π' *of* X *into clusters lying in balls of radius at most* $\Gamma \ge 1$*, and a value* $\Lambda \ge 8\Gamma$*, there is a randomized algorithm to create a* $(\Lambda + \Gamma)$*-ball partition* Π'' *of* X*, where each cluster of* Π' *is contained in some cluster of* Π''*, and for any* $x \in X$ *and radius* $0 \le r \le \Lambda$,

$$\Pr[\mathbf{B}(x, r) \text{ is cut in } \Pi''] \le \frac{O(r + \Gamma)}{\Lambda}\alpha.$$

Proof. Note that we can assume that $\Gamma < \Lambda/c\alpha$ and $\Lambda \ge \alpha$, since otherwise the lemma is trivially true. Using the algorithmCUT-CLUSTERS given in Figure 3.1, we create a partition of Y (and hence of X); all distances are measured according to the original distance function d in X.

Let us define $\mathcal{B}_x = \mathbf{B}(x, r)$. Note that if $\mathcal{B}_x$ is cut in Π'' due to some value of L from $v \in N$ (for the first time), then L falls into the interval $[d(v, x) - r - \Gamma, d(v, x) + r + \Gamma]$.

0. Let $Y \leftarrow X$, $p \leftarrow \frac{c\alpha\Gamma}{\Lambda}$ for constant c to be fixed later, N be a $\Lambda/2$-net of X.
1. Pick an arbitrary "root" vertex $v \in N$ not picked before
2. Set the initial value of the "radius" $L \leftarrow \Lambda/2$
3. Flip a coin with bias p
4. If the coin comes up heads, goto Step 11
5. If the coin comes up tails, increment L by Γ
6. If $L > \Lambda(1 - 1/4\alpha)$
7. choose a value $\hat{L}$ from $[0, \Lambda/(4\alpha)]$ u.a.r.
8. round down $\hat{L}$ to the nearest multiple of Γ
9. set $L \leftarrow \Lambda(1 - 1/4\alpha) + \hat{L}$
10. Else goto Step 3
11. Form a new cluster C' in Π'' containing all clusters in $\Pi' \cap Y$ with centers lie in $\mathbf{B}(v, L)$
12. Remove the vertices in C' from Y
13. (Remark: C' has radius at most $\Lambda + \Gamma$)
14. If $Y \neq \emptyset$ goto Step 1
15. End

Figure 3.1: **Algorithm** CUT-CLUSTERS

Indeed, if $\mathcal{B}_x$ is cut in Π'', there are at least two clusters $C'_1, C'_2 \in \Pi'$ such that they both cut $\mathcal{B}_x$, and $\mathbf{B}(v, L)$ contains one of their centers but not both. Since both clusters intersect $\mathcal{B}_x$, their centers c'_1 and c'_2 are at distance at most $r + \Gamma$ from x. If $L < d(v, x) - r - \Gamma$, the triangle inequality implies that $\mathbf{B}(v, L)$ cannot contain either center. Similarly, if $L > d(v, x) + r + \Gamma$, $\mathbf{B}(v, L)$ contains both of them. Hence the value of L must fall into the interval indicated above.

If a cut in Step 11-12 is made due to the appearance of a heads in Step 4, we call such a cut a *normal cut*; else we call it a *forced cut*. We now bound the probability that the ball $\mathcal{B}_x = \mathbf{B}(x, r)$ is cut due to either type.

Normal cuts. Consider the first instant in time when the parameter L for some root $v \in N$ reaches a value such that the cut obtained by taking all $\Pi' \cap Y$ clusters with centers in $B(v, L)$ would cut $\mathcal{B}_x$. (If there is no such time, then $\mathcal{B}_x$ is never cut by a normal cut.) In this case, L must also be in the range $d(v, x) \pm (r + \Gamma)$, and increases with time. Now either (i) we make a normal cut before L goes outside this range; or (ii) we make a forced cut; or (iii) L goes outside the range and we make no cut in this range. In any case, the fate of $\mathcal{B}_x$ is decided; $\mathcal{B}_x$ is either cut or contained in a new cluster with center v. We now upper-bound the probability that event (i) happens. There are at most $2(r+\Gamma)/\Gamma$ coin flips made (with bias p) when the value of L is in the correct range of width at most $2(r + \Gamma)$ and one of these flips must come up heads for the cut to be made. The trivial union bound now shows this probability to be at most $\frac{2(r+\Gamma)}{\Gamma}\, p = \frac{2c(r+\Gamma)}{\Lambda}\alpha$.

Forced cuts. Let us look at some root $v \in N$ and bound the probability that a forced cut is made with cutting radius L from v in some range $\mathcal{R}_x = d(v, x) \pm (r + \Gamma)$. Since the cut is forced and the value of L is greater than $\Lambda(1 - 1/4\alpha) \geq 3\Lambda/4$, we must have flipped a sequence of at least $\Lambda/4\Gamma$ successive tails; the probability of this event is at most

$$(1 - p)^{(\Lambda/4\Gamma)} \leq e^{-p\Lambda/4\Gamma} = e^{-\frac{c}{4}\alpha}. \tag{3.1}$$

Now, we choose $\hat{L}$ to be a multiple of Γ uniformly in a range of width at most $\Lambda/4\alpha$, and hence the probability that L falls into a range of length $2(r + \Gamma)$ is at most $2(r + \Gamma)/(\Lambda/4\alpha)$. Multiplying this by (3.1), we obtain a bound of $e^{-\frac{c}{4}\alpha} \times \frac{8(r+\Gamma)}{\Lambda}\,\alpha$ on the probability that a forced cut is made around v with L in the range $\mathcal{R}_x$ such that the cluster C' with center v in Π'' may cut $\mathcal{B}_x$. Finally, for any $x \in X$, $\mathcal{B}_x$ can only be cut by clusters from roots $v \in N$ that are at distance at most $(r+\Gamma)+\Lambda \leq 3\Lambda$ from x; by Prop. 2.1, there are at most $|\mathbf{B}(x, 3\Lambda) \cap N| = (\frac{6\Lambda}{\Lambda/2})^{\alpha} \leq (12)^{\alpha}$ of such roots. Now we choose c to be large enough; the probability of B_x being cut by a forced due to any such root is at most $12^{\alpha} \times e^{-\frac{c}{4}\alpha} \times \frac{8(r+\Gamma)}{\Lambda}\,\alpha \leq \frac{O(r+\Gamma)}{\Lambda}\alpha$ by the union bound.■

We now use the above lemma to prove Theorem 3.2. Using $\Pi' = \Pi_i$, $\Gamma = \eta_i < \rho^i(\rho/(\rho-1))$, and $\Lambda = \eta_{i+1} - \Gamma = \rho^{i+1}$, and using $N = N_{i+1}$ (which is a $\rho^{i+1}/2 = \Lambda/2$ net), we create a $(\Gamma + \Lambda = \eta_{i+1})$-ball partition such that for all x and all $r \leq \rho^{i+1}$ and $\varepsilon = O(1/\alpha)$, we have

$$\Pr[\mathbf{B}(x, \varepsilon r) \text{ cut}] \leq \frac{O(\varepsilon r+\Gamma)}{\Lambda}\,\alpha \leq \frac{O(\rho^i)}{\rho^{i+1}}\,\alpha \leq \frac{1}{10} < \frac{1}{2}, \tag{3.2}$$

for ρ/α and c being large enough constants. The probability distribution μ over all decompositions $\mathbf{\Pi}$ thus generated satisfy the requirements of a PPHD as given in Definition 3.1. Finally, we bound the degree $\deg(\mu)$ of the PPHD μ; note that each level-i cluster is centered at some $v \in N_i$, hence the number of level-i clusters contained in some level-$(i+1)$ cluster is $(2\eta_{i+1}/(\rho^i/2))^{O(\alpha)} = \alpha^{O(\alpha)}$ by Prop. 2.1. ■

Few Hierarchical Decompositions. The above proof immediately gives us a PPHD μ_M with a support on only $M = O(\log n + \log\log\Delta)$ HDs. By sampling from the distribution μ for M times, we get the HDs $\mathbf{\Pi}^{(1)}, \ldots, \mathbf{\Pi}^{(M)}$, and let the PPHD μ_M be the uniform distribution on these HDs. By (3.2), for each $j \in [1 \ldots M]$, point $x \in X$ and radius $r \leq \rho^i$, $\mathbf{B}(x, \varepsilon r)$ is not cut in the partition $\Pi_i^{(j)}$ with probability $1/10$; hence a Chernoff bound implies that this ball is cut in the level-i partitions of more than $M/2$ of the HDs with probability less than $1/(n \log\Delta)^{O(1)}$. Now taking the trivial union bound over all possible values of the center $x \in X$, and all the $\log\Delta$ values of r which are powers of 2 shows that the μ_M is a $(\rho, \varepsilon/2)$-PPHD **whp**.

3.1.2 Even Fewer Hierarchical Decompositions

While the proof of Theorem 3.2 and the discussion above do not produce a PPHD with small support (of size $O(\alpha \log\alpha)$), we have seen all the essential ideas required to prove the existence of such a distribution μ_m and hence to complete the proof of Theorem 3.1. To prove this result, we use the locality of the construction, in conjunction with the Lovász Local Lemma (LLL). This locality property is the very reason why we built the hierarchical decomposition bottom-up; it ensures that if any particular ball is not cut at some low level i (the "local decisions"), it is not cut at levels higher than i (i.e., the "non-local decisions"). Also, we choose the decomposition procedure of Theorem 3.2 in preference to others (e.g., those in [13] and [25]) since they choose a single random radius for all clusters in one particular partition Π of X, which causes correlations across the entire metric space. (The LLL has been used in similar contexts in [13, 19].)

Proof of Theorem 3.1: To show that there is a distribution μ_m over only $m = O(\alpha \log \alpha)$ trees, we use an idea similar to that in the previous section, augmented with some ideas from [13]. Instead of building one hierarchical decomposition $\mathbf{\Pi}$ bottom-up, we build m hierarchical decompositions $\mathbf{\Pi}^{(1)}, \ldots, \mathbf{\Pi}^{(m)}$ simultaneously (also from the bottom up).

As before, the proof proceeds inductively; we assume that we are given level-i partitions $\Pi_i^{(1)}, \ldots, \Pi_i^{(m)}$, where $\Pi_i^{(j)}$ is the level-i partition belonging to $\mathbf{\Pi}^{(j)}$. We then show that we can build level-$(i+1)$ partitions $\Pi_{i+1}^{(1)}, \ldots, \Pi_{i+1}^{(m)}$ where each $\Pi_i^{(j)}$ is a refinement of the corresponding $\Pi_{i+1}^{(j)}$, and any given ball $\mathbf{B}(x, \varepsilon r)$ with $\rho^i \leq r \leq \rho^{i+1}$ is cut in at most $m/2$ of these level-$(i+1)$ partitions. We start off this process with each $\Pi_0^{(j)} = \{\{x\} : x \in X\}$ being the partition consisting of all singleton points in X. Let $J = \{1, \ldots, m\}$. Given m level-i partitions $(\Pi_i^{(j)})_{j \in J}$, we create m level-$(i+1)$ partitions $(\Pi_{i+1}^{(j)})_{j \in J}$ using the procedure in Lemma 3.2 independently on each of the m decompositions; parameters are set as in the proof of Theorem 3.2, with $\Lambda = \rho^{i+1}$, $\Gamma = \eta_i$, and $\varepsilon = 1/O(\alpha)$. This extends the m hierarchical decompositions to the $(i+1)^{st}$ level; it remains to show that the probability of balls being cut is small.

To describe the events of interest, let us take $\beta = \varepsilon \rho^{i+1}$ and define Z to be a β-net of X. For each $z \in Z$, define $\mathcal{B}_z$ to be $\mathbf{B}(z, 2\beta)$, and $\mathcal{E}_z^{i+1}$ to be event that $\mathcal{B}_z$ is cut in more than $m/2$ of the partitions $(\Pi_{i+1}^{(j)})_{j=1}^m$, which we refer to as a "bad" event (used in Section 3.2). We prove the claim using the Lovász Local Lemma.

CLAIM 3.3. *Given any* $(\Pi_i^{(j)})_{j=1}^m$, $\Pr[\bigwedge_{z \in Z} \overline{\mathcal{E}_z^{i+1}}] > 0$.

LEMMA 3.3. **(Lovász Local Lemma)** *Given a set of events* $\{\mathcal{E}_z^{i+1}\}_{z \in Z}$*, suppose that each event is mutually independent of all but at most* B *other events. Further suppose that, for each event* $\mathcal{E}_z^{i+1}$, $\Pr[\mathcal{E}_z^{i+1}] \leq p$. *Then if* $\mathrm{e}p(B+1) < 1$, $\Pr[\bigwedge_{z \in Z} \overline{\mathcal{E}_z^{i+1}}] > 0$.

Proof of Claim 3.3: First, let us calculate the probability of $\mathcal{E}_z^{i+1}$: by changing the constant in ε, we can make the probability that a ball $\mathcal{B}_z$ is cut in one level-$(i+1)$ partition to be at most $1/8$. Let us denote by A_z^j the event that $\mathcal{B}_z$ is cut in partition $\Pi_{i+1}^{(j)}$. The expected number of partitions in which the ball is cut is at most $m/8$. Since the partitions are constructed independently, the probability for the event $\mathcal{E}_z^{i+1}$ that $\mathcal{B}_z$ is cut in $m/2$ partitions (which is at least four times the expectation) is at most $\exp(-9m/40)$; this can be established using a standard Chernoff bound. This, in turn, is at most $(0.8)^m$, which we define to be p.

Next we show that an event $\mathcal{E}_z^{i+1}$ is mutually independent of all events $\mathcal{E}_{z'}^{i+1}$ such that $d(z, z') > 4\eta_{i+1}$. For each partition $\Pi_{i+1}^{(j)}$, each root $v \in N_{i+1}$ determines its radius by conducting a random experiment independent of any other roots' experiments. These random experiments, and only these, determine whether events such as A_z^j occur. In turn, whether event $\mathcal{E}_z^{i+1}$ occurs is determined only by events $A_z^1, \ldots, A_z^m$. For a particular j, for each z, all of the cuts that could affect $\mathcal{B}_z$ in the algorithm CUT-CLUSTERS are made from roots $v \in N_{i+1}$ at distance at most $2\beta + \Gamma + \Lambda = 2\beta + \eta_{i+1} < 2\eta_{i+1}$ from z. Whether event A_z^j occurs is determined by the experiments corresponding to these roots alone. If $d(z, z') > 4\eta_{i+1}$, then there is no intersection between the experiments for z and the experiments for z'. Since $\mathcal{E}_z^{i+1}$ is determined by $A_z^1, \ldots, A_z^m$, $\mathcal{E}_z^{i+1}$ is mutually independent of the set of all $\mathcal{E}_{z'}^{i+1}$ such that $d(z, z') > 4\eta_{i+1}$.

We apply the LLL now. Note that the number of $z' \in Z$ within distance $4\eta_{i+1}$ of $\mathcal{E}_z^{i+1}$ for $z \in Z$ is at most $|\mathbf{B}(z, 4\eta_{i+1}) \cap Z| \leq \left(\frac{8\eta_{i+1}}{\beta}\right)^{\alpha} \leq O(\alpha)^{\alpha}$. We define this quantity to be B; $\mathrm{e}p(B+1)$ is at most 1 for $m = O(\alpha \log \alpha)$ and Claim 3.3 follows. ■

Having proved the claim, let us now show that with nonzero probability, each $\mathbf{B}(x, r)$ for $x \in X$ and $\rho^i \leq r \leq \rho^{i+1}$ is not cut in at least $m/2$ of the level-$(i+1)$ partitions $(\Pi_{i+1}^{(j)})_{j \in J}$. Let us call this event SC_{i+1}. The claim shows that with nonzero probability, each ball $\mathcal{B}_z$ with $z \in Z$ is not cut in at least $m/2$ of the partitions $(\Pi_{i+1}^{(j)})_{j \in J}$. Since each $x \in X$ is at distance at most β to some $z_x \in Z$, the triangle inequality implies that $\mathbf{B}(x, \varepsilon r) \subseteq \mathbf{B}(x, \beta)$ is not cut if $\mathbf{B}(z_x, 2\beta)$ is not cut, which holds in at least half of the partitions. Hence SC_{i+1} also holds with nonzero probability.

Finally, we prove that we can choose a random set of HD's $(\mathbf{\Pi}^{(j)})_{j \in J}$ such that SC_{i+1} occurs for each $1 \leq i+1 \leq h$ *simultaneously* with nonzero probability. The key to the proof is that we have assumed an arbitrary (worst-case) set of partitions $(\Pi_i^{(j)})_{j=1}^m$ at level i in proving a nonzero lower bound on $\Pr[SC_{i+1}]$. Hence, we can ignore any dependence among the events SC_{i+1} for $1 \leq i+1 \leq h$, and simply multiply their nonzero probabilities together to obtain a nonzero lower bound on the probability that they all occur simultaneously. ■

3.2 An Algorithm for Finding the Decompositions

The above procedure can be made algorithmic using an approach based on Beck's algorithmic version of the LLL (see, e.g., [1, 5]). The decomposition satisfies all properties of the one that is shown to exist using LLL in Theorem 3.1, although with some changes in constant parameter values. As in the proof of Theorem 3.1, we build $m = O(\alpha \log \alpha)$ HDs level by level in a bottom-up fashion.

On any particular level $i+1$, we begin by choosing m partitions at random. After making the random choices, we examine the partitions and identify all of the bad events that have occurred. We then group together bad events that may depend on each other, as well as "good" events that may depend on the bad events. Each group forms a connected component in the LLL dependency graph. We show that, with high probability, all connected components have size $O(\log \nu)$, where $\nu = |Z|$ is the size of the $\varepsilon\rho^{i+1}$-net of X.

Once the groups have been identified, we need to eliminate the bad events. Hence, for each group, we "undo" all of the random choices concerning that group, while not modifying any choices that do not affect the group. New choices must be made for each group so that no bad event occurs. Because the group size is small (the number of centers $v \in N_{i+1}$ concerning the group that we choose random radius for is also $O(\log \nu)$), we can find new settings for these choices using exhaustive search in polynomial time.

One interesting complication in this proof is that the set of clusters containing a group have different shapes in the m different partitions. In each partition, we cut out a "hole", and redo the choices within the hole. The boundary of the hole is formed from the boundaries of the clusters that may influence the bad events (and the good events) in the group. In forming the boundary, additional good events may be added to the hole. As a consequence, it is possible that a good event inside a hole in one partition may appear inside a different hole in another partition. Hence, when we perform exhaustive search, these holes must be considered together. However, our method of bounding the size of each connected component already takes into account any merging of holes on account of shared good events, so that we never have to redo the choices for a group of size more than $O(\log \nu)$.

Another issue is that the subset of centers in a hole that belong to N_{i+1}, the $\rho^{i+1}/2$-net that covers the entire metric, may not by themselves cover the hole. (Portions of the hole may be covered by centers outside the hole.) So for each of the m partitions, we may have to add additional net points inside the hole to obtain a complete cover for it. We show that the size of net points in the hole increases by only a constant factor and remains $O(\log \nu)$, and the degree of the hierarchical decomposition trees is at most $\alpha^{O(\alpha)}$ as before.

4 The $(1+\tau)$-Stretch Routing Schemes

Given a (ρ, ε)-PPHD μ_m with a support on m HDs, we can now define, for every $0 < \tau \leq 1$, a $(1+\tau)$-stretch routing scheme which uses routing tables of size at most $m(\alpha/\tau)^{O(\alpha)} \log^2 \Delta$ bits at every node.

We consider routing schemes in two models. In a basic model, we assume that there is no underlying routing fabric and each node can only send packets to its direct neighbors. In a second model, we can build an overlay hierarchical routing scheme upon an underlying routing fabric like IP that can send packets to any specific node in the network. We specify the routing algorithm in the basic model, but also indicate how one can circumvent certain steps of this algorithm when an underlying routing mechanism is given.

Let us recall some of the notation defined earlier. Let $(\mathbf{\Pi}^{(j)})_{j=1}^{m}$ be the m hierarchical decompositions on which μ_m has positive support, and the level-i partition corresponding to $\mathbf{\Pi}^{(j)}$ be called $\Pi_i^{(j)}$. Recall that we can associate each hierarchical decomposition $\mathbf{\Pi}^{(j)}$ with a tree T_j (as outlined in Section 2.1). Note that each of these trees has a $\deg(\mu_m)$ bounded by $\alpha^{O(\alpha)}$ and a height of at most $h = \lceil \log_\rho \Delta \rceil$. Recall that each internal vertex of the tree T_j at level i corresponds to a cluster of $\Pi_i^{(j)}$ and leaves of $T_j, \forall j \in J$, correspond to vertices in X, where $J = \{1, \ldots, m\}$. Let each internal vertex v of each tree T_j *label* its children by numbers between 1 and $\deg(\mu_m)$; v does not label anything with the number 0, but uses it to refer to its parent. Note that this allows us to represent any path in a tree T_j by a sequence of at most $2h = O(\log_\rho \Delta)$ labels.

Lemma 3.1 already shows that the m trees thus created form a small $O(\rho/\varepsilon) = O(\alpha^2)$-stretch Steiner tree cover, which can be used for routing purposes (as in Section 4.3). However, since such a large stretch is not always acceptable, we improve on this scheme in the following subsections to get better routing bounds.

4.1 The Addressing Scheme Given a tree T_j and a vertex $x \in X$, we assign x a *local address* $\mathsf{addr}_j(x)$, which consists of $h = \lceil \log_\rho \Delta \rceil$ blocks, one for each level of the tree T_j. Each block has a fixed length. The i^{th} *block* of the $\mathsf{addr}_j(x)$ corresponds to partition $\Pi_i^{(j)}$ and contains the label assigned to the cluster C_x containing x in $\Pi_i^{(j)}$ by C_x's parent in T_j. Since any such label is just a number between 1 and $\deg(\mu_m)$, where $\deg(\mu_m) = \alpha^{O(\alpha)}$, we need $O(\alpha \log \alpha)$ bits per block. In fact, one can extend this addressing scheme to any cluster C in T_j. If C is a level-i cluster, the k^{th}-block of $\mathsf{addr}_j(C)$ contains $*$'s for $k < i$; $\mathsf{addr}_j(X)$ for the root cluster of T_j contains all $*$'s matching all vertices in X.

The *global address* $\mathsf{addr}(x)$ of point $x \in X$ is the concatenation $\langle \mathsf{addr}_1(x), \cdots, \mathsf{addr}_m(x) \rangle$ of its local addresses $\mathsf{addr}_j(x)$ for $j \in J$. Since each cluster C belongs to only one tree T_j, we define $\mathsf{addr}_{j'}(C)$ to be a sequence of #'s of the correct length (where # are dummy symbols matching nothing), and hence define a global address of C as well. (This is only for simplicity; in actual implementations, cluster addresses for T_j can be given by the tuple $\langle \mathsf{addr}_j(C), j \rangle$.)

Since there are $O(\alpha \log \alpha)$ bits per block, h blocks per local address, and m local addresses per global address, substitution of the appropriate values gives the address length A to be at most $m \times h \times \lceil \log(\deg(\mu_m)) \rceil = O(\alpha \log \alpha) \times \lceil \log_\rho \Delta \rceil \times O(\alpha \log \alpha) = O(\alpha^2 \log \alpha \log \Delta)$ bits.

4.2 The Routing Table For each point $x \in X$, we maintain a routing table Route_x that contains the following information for each T_j, $1 \le j \le m$:

1. For each ancestor of x in T_j that corresponds to a cluster C containing x, we maintain a table entry for C.
2. Moreover, for each such C, we maintain an entry for each descendant of C in T_j reachable within ℓ hops in tree T_j. Here $\ell = \Theta(\log_\rho 1/\varepsilon\tau)$, with the constants chosen such that $\eta_{i-\ell} \le \frac{\varepsilon\tau}{4}\rho^{i-1}$.

In the routing table Route_x for x, each of the above entries thus corresponds to some level-i' cluster C' in T_j. Let $\mathsf{close}_x(C')$ be the closest point in C' to x. (We assume, w.l.o.g., that ties are broken in some consistent way, so that any node y on a shortest path from x to $\mathsf{close}_x(C')$ has the value $\mathsf{close}_y(C') = \mathsf{close}_x(C')$; in fact, this consistency is the only property we use.) For this C', Route_x stores **(a)** the global address $\mathsf{addr}(C')$ by which the table is indexed, **(b)** the identity of a "next hop" neighbor y of x that stays on a shortest path from x to the closest point $\mathsf{close}_x(C')$ in C', and **(c)** an extra bit $\mathsf{ValidPath}_x(C')$: if the cluster ℓ levels above C' in T_j is the cluster C, then $\mathsf{ValidPath}_x(C')$ is set to be `true` if $\mathbf{B}(x, \varepsilon\rho^{i'+\ell})$ is entirely contained within cluster C and $d(x, \mathsf{close}_x(C')) \le \varepsilon\rho^{i'+\ell}$, and is set to be `false` otherwise. Of course, if we reach the root of T_j while trying to go up ℓ levels, then the bit is set to be `true`. Note that if there is an underlying routing fabric like IP, we can store the IP-address of some node in C' (say, the closest one) instead of **(b)** and **(c)** above.

LEMMA 4.1. *The number of entries in the routing table* Route_x *of any* $x \in X$ *is at most* $\log \Delta \times (\alpha/\tau)^{O(\alpha)}$.

Proof. Let us estimate the number of entries in Route_x for any $x \in X$. There are m trees. For each tree T_j, for all $j \in J$, there are $h = \lceil \log_\rho \Delta \rceil$ ancestors of x and the degree of the tree is bounded by $\deg(\mu_m) = \alpha^{O(\alpha)}$. Recall that ρ and $1/\varepsilon$ are both $O(\alpha)$, and hence $\ell = O(\log(\alpha/\tau))$. Plugging these values in, we get that the number of entries for x across m trees is at most $m \times h \times (\deg(\mu_m))^\ell = O(\alpha \log \alpha) \times O(\log_\alpha \Delta) \times \alpha^{O(\alpha\ell)} = \log \Delta \times (\alpha/\tau)^{O(\alpha)}$. Each entry is indexed by one global address (of at most $A = O(\alpha^2 \log \alpha \log \Delta)$ bits), and contains the identity of the next hop (which uses $O(\log \text{degree-of-}x) = O(\log n)$ bits) and one additional $\mathsf{ValidPath}$ bit. ∎

The forwarding algorithm makes use of two functions, $\mathsf{NextHop}_x$ and $\mathsf{PrefMatch}_x$. For a point x and a level-i' cluster C' in T_j, the function $\mathsf{NextHop}_x(\mathsf{addr}(C'))$ returns the next hop on the path from x to $\mathsf{close}_x(C')$ provided that the next hop does not leave the cluster C at level $i' + \ell$ that contains C', and null otherwise. (As we shall see, the packet forwarding algorithm is guaranteed never to encounter a null next hop.) Given points x and t in X, the function $\mathsf{PrefMatch}_x(t)$ returns an $\mathsf{addr}(C')$ in Route_x such that in some T_j, t belongs to the level-i cluster C', $\mathsf{ValidPath}_x(C')$ is `true`, and the value i is *the smallest* across all trees. Note that both of these functions can be computed efficiently by node x. Furthermore, it is possible to support the functions with data structures of size comparable to that of Route_x.

Note that once the points in X have been assigned addresses (for which we have described only an off-line algorithm), the routing tables can be built up in a completely distributed fashion. In particular, a distributed breadth-first-search algorithm can be applied to determine whether a ball of a certain radius is cut in a particular decomposition, and a distributed implementation of the Bellman-Ford algorithm can be used to establish the next-hop entries for destinations for which the shortest paths lie within a certain cluster.

4.3 The Forwarding Algorithm The idea behind the forwarding algorithm is to start a packet off from its origin s towards an *intermediate* cluster C containing its destination t; the packet header thus consists of two pieces of information $\langle \mathsf{addr}(t), \mathsf{addr}(C) \rangle$, where t is the destination node for the packet and C is the *intermediate* cluster containing t. Initially, the cluster can be chosen (degenerately) to be the root cluster of (say) tree T_1.

Upon reaching a node x in the intermediate cluster C, a new and smaller intermediate cluster C', also containing t, must be chosen, possibly from a different tree; the packet header must be updated with $\mathsf{addr}(C')$ that remains the same until reaching C'. Suppose that the new cluster C' containing t is at level i'. After selecting this cluster, the packet is sent off towards C' with the new header, following a shortest path that stays within the cluster $\hat{C}$ at level $i' + \ell$ that contains both x and C'. This process is repeated until ultimately the packet reaches the cluster containing only the destination t. The algorithm is presented in Figure 4.2.

THEOREM 4.1. *The forwarding algorithm has a stretch of at most* $(1 + \tau)$*, where* $\tau \le 1$.

Proof. We first show that the algorithm is indeed valid; each of the steps can be executed and the packet eventually reaches t. Suppose that the packet has just reached a node x in an intermediate cluster C containing t (with $\mathsf{addr}(C)$ in its header); thus x needs to execute Step 3 to find a new cluster C' containing t. Clearly, $\mathsf{PrefMatch}_x(t)$ can return the root cluster C_{root} of any T_j, since it contains t. We show, however, that the cluster C' returned by $\mathsf{PrefMatch}_x(t)$ has a small diameter and nodes along a valid shortest path from x to C' will forward the packet correctly until it reaches C'.

1. Let packet header be $\langle \mathsf{addr}(t), \mathsf{addr}(C) \rangle$.
2. If C contains x, the current node, then
3.   find $\mathsf{addr}(C') \leftarrow \mathsf{PrefMatch}_x(t)$
4.   let $y \leftarrow \mathsf{NextHop}_x(\mathsf{addr}(C'))$
5.   forward packet with new header
6. $\langle \mathsf{addr}(t), \mathsf{addr}(C') \rangle$ to y.
7. Else (now $x \notin C$)
8.   let $y \leftarrow \mathsf{NextHop}_x(\mathsf{addr}(C))$
9.   forward packet with unchanged header
10. $\langle \mathsf{addr}(t), \mathsf{addr}(C) \rangle$ to y.
11. End

Figure 4.2: The Forwarding Algorithm at Node x

LEMMA 4.2. *If the packet is at node x with distance to the target t being $d(x,t) \leq \varepsilon\rho^i$, Step 3 must return some $\mathsf{addr}(C')$ such that cluster $C' \ni t$ is at level at most $(i-\ell)$ in some $T_{j'}$ with $\mathsf{ValidPath}_x(C')$ being $\mathtt{true}$. Furthermore, all vertex v on all shortest paths from x to $\mathsf{close}_x(C') = \mathsf{close}_v(C')$ has a non-null $\mathsf{NextHop}_v(\mathsf{addr}(C'))$.*

Proof. The (ρ,ε)-PPHD ensures that there exists at least one tree T_j such that $\mathbf{B}(x,\varepsilon\rho^i)$ is not cut in the level-i partition $\Pi_i^{(j)}$; let $\hat{C}_{cont} \in \Pi_i^{(j)}$ be the level-i cluster in T_j that contains $\mathbf{B}(x,\varepsilon\rho^i)$. Let $C_t \in \Pi_{i-\ell}^{(j)}$ be the level-$(i-\ell)$ cluster in T_j containing t. The $\mathsf{ValidPath}_x(C_t)$ bit must be $\mathtt{true}$ since $\mathbf{B}(x,\varepsilon\rho^i) \subseteq \hat{C}_{cont}$ in $\Pi_i^{(j)}$ and $d(x, \mathsf{close}_x(C_t)) \leq d(x,t) \leq \varepsilon\rho^i$; thus $\mathsf{PrefMatch}_x$ can (and may indeed) just return $\mathsf{addr}(C_t)$ given no "better" choices. However, $\mathsf{PrefMatch}_x$ always finds a cluster C' in some $T_{j'}$, at the *lowest* level across all trees, such that $t \in C'$, and $\mathsf{ValidPath}_x(C')$ is $\mathtt{true}$ in Route_x. Let the level of C' be i'; the value i' is at most $(i-\ell)$. Now Let $\hat{C} \in \Pi_{i'+\ell}^{(j')}$ be the cluster ℓ levels above $C' \in \Pi_{i'}^{(j')}$ in $T_{j'}$ that contains both x and C'. (Such $\hat{C}$ must exist at level $i'+\ell$ for $\mathsf{addr}(C')$ to be in Route_x.) We know that $\mathbf{B}(x,\varepsilon\rho^{i'+\ell}) \subseteq \hat{C}$ and $d(x,\mathsf{close}_x(C')) \leq \varepsilon\rho^{i'+\ell}$ since $\mathsf{ValidPath}_x(C')$ is $\mathtt{true}$ in Route_x. Thus all shortest paths from x to $\mathsf{close}_x(C')$ are entirely contained in $\hat{C}$. Hence, the $\mathsf{NextHop}_v(\mathsf{addr}(C'))$ pointer at any node v on one of these paths must be non-null since all shortest paths from v to $\mathsf{close}_v(C') = \mathsf{close}_x(C')$ are all contained in $\hat{C}$, the cluster ℓ levels above C' in T'_j. ∎

It remains to bound the path stretch. Consider the case when a packet is sent from s to t. Let C' be a cluster at level $i-\ell$ returned by Step 3 of the forwarding algorithm. Note that if the level $i \leq \ell$, then $C' = \{t\}$ and we send the packet directly to t with $\tau = 0$. Using these short distances as the base case, we now do induction on the distance from s to t.

If C' is a non-trivial cluster containing t, then we go on a shortest path from s to some vertex $v = \mathsf{close}_s(C') \in C'$. Since $t \in C'$, $d(s,v) \leq d(s,t)$. Because the diameter of C' is at most $2\eta_{i-\ell}$, $d(v,t) \leq 2\eta_{i-\ell} < \varepsilon\rho^{i-1} < d(s,t)$. (The last inequality holds because if $\varepsilon\rho^{i-1} \geq d(s,t)$, then $\mathsf{PrefMatch}_s$ would have returned a cluster at a level lower than that of C' by Lemma 4.2.) Hence, we can apply the induction hypothesis to find a path from v to t of length at most $(1+\tau)d(v,t) \leq (1+\tau)2\eta_{i-\ell}$. The path from s to t as derived from Route_s is of length at most $d(s,v) + (1+\tau)d(v,t) < d(s,t) + (1+\tau)2\eta_{i-\ell}$. The stretch of the path from s is t is then $1 + (1+\tau)2\eta_{i-\ell}/d(s,t)$. This quantity is at most $1+\tau$ since $\tau \leq 1$ and we have chosen constants so that $\eta_{i-\ell} \leq \tau\varepsilon\rho^{i-1}/4$. ∎

4.4 Reducing table size to $O_{\alpha,\tau}(\log \Delta)$ Here are some ideas to further reduce the storage. We choose a *portal* $\mathsf{portal}_j(C) \in C$ for each cluster C in T_j, so that $x \in X$ is a representative for at most one cluster from each T_j (T_j must have no internal degree 2 nodes, which can be done). Set $\ell = \Theta(\log_\rho m/\varepsilon\tau)$, a bit bigger than before. Each node x keeps a smaller routing table Route_x; instead of keeping entries for "ℓ-levels down" clusters for *all* ancestors in each T_j, it keeps entries only for *its ℓ nearest* ancestors. Furthermore, if x is the portal for cluster $C \in T_j$, entries for the ℓ nearest ancestors of C are also kept at x, but in a new routing table Route_C. The total size of all $\leq (m+1)$ routing tables at x is only $O_{\alpha,\tau}(\log \Delta)$ bits now. We can now imagine "routing to a cluster $C \in T_j$" by sending the packet to $\mathsf{portal}(C)$, and each node x runs possibly $m+1$ copies of the routing algorithm: one for itself and one for each C with $\mathsf{portal}(C) = x$.

Suppose one routes a packet from s to t, we change the forwarding algorithm at node s as follows. If the $\mathsf{lca}(s,t)$ has level at most ℓ, the routing proceeds as before. Otherwise, the longest prefix matches at s will return clusters ℓ levels above s, in which case s forwards the packet in round-robin fashion to its parent clusters (one in each T_j) to try and find one whose ancestor ℓ levels up contains t in its subtree. If this fails, s tries the clusters two levels up in each tree, and so on. This doubling search procedure ensures that the extra distance traveled before we see the lca of s and t in some tree is only $\tau d(s,t)$, and hence the stretch remains $(1+\tau)$.

5 Constant-Degree Spanners for Doubling Metrics

Given a metric (V,d) with doubling dimension α and $\tau > 0$, this section shows how to construct a $(1+\tau)$-spanner whose maximum degree is bounded by $(2+\frac{1}{\tau})^{O(\alpha)}$. Our construction consists of two phases. In the first phase, we construct a spanner $(V,\hat{E})$ from a nested sequence of nets $\{Y_i\}$; we include an edge if the end points are from the same net and "reasonably close" to each other. We then show that the edges in this spanner can be directed such that the out-degree of each vertex is bounded, and hence the spanner is sparse. We then have a second phase, in which we modify these edges in $\hat{E}$ to obtain another spanner, but now with bounded degree. Our main theorem, whose proof we sketch in Section 5.1, is the following:

THEOREM 5.1. *Given a metric (V, d) with doubling dimension α, there exists a $(1+\tau)$-spanner such that the degree of every vertex is at most $(2+\frac{1}{\tau})^{O(\alpha)}$.*

5.1 Constructing a sparse $(1+\tau)$-spanner We first describe the construction of a sparse $(1+\tau)$-spanner. Without loss of generality, we assume $\tau \leq \frac{1}{2}$. For $\tau > \frac{1}{2}$, we still run the whole procedure with $\tau' = \frac{1}{2}$. All the bounds would still hold because $4^{O(\alpha)} = (2+\frac{1}{\tau})^{O(\alpha)}$. Without loss of generality, we assume that the distance between any two distinct vertices is at least 1. Otherwise, we can re-scale the metric. Given $\tau > 0$, let $\gamma := 4 + \frac{32}{\tau}$ and $p := \lceil \log_2 \gamma \rceil$.

Our construction requires a hierarchical sequence of nets, which is defined as follows. Define $Y_{-p} := V$. For $i > -p$, let Y_i be a 2^i-net of Y_{i-1}. (Note that since the inter-vertex distance is at least 1, $Y_i = V$ for $-p \leq i < 0$.) For each net Y_i in the sequence, we include the edges whose end points are in the net and are close together. In particular, define for $i \geq -p$, $E_i = \{(u,v) \in Y_i \times Y_i \,|\, \gamma \cdot 2^{i-1} < d(u,v) \leq \gamma \cdot 2^i\}$. Let $\hat{E} = \cup_i E_i$, and $(V, \hat{E})$ is the spanner obtained from the construction. The following lemma shows that $(V, \hat{E})$ preserves distances in the metric and is sparse:

LEMMA 5.1. *The graph $(V, \hat{E})$ is a $(1+\tau)$-spanner for (V, d). Furthermore, the edges of $\hat{E}$ can be directed such that each vertex has out-degree bounded by $(2+\frac{1}{\tau})^{O(\alpha)}$.*

While we omit the proof, let us indicate how to direct the edges. For each $v \in V$, define $i^*(v) := \max\{i \,|\, v \in Y_i\}$. For each edge $(u,v) \in \hat{E}$, direct it from u to v if $i^*(u) < i^*(v)$; if $i^*(u) = i^*(v)$, direct the edge arbitrarily.

Bounded-degree spanners: We now modify $\hat{E}$ to get another spanner $(G, \tilde{E})$ with the same number of edges, but with bounded degree in the following way. Let l be the smallest positive integer such that $\frac{1}{2^{l-1}} \leq \tau$. Then $l = O(\log \frac{1}{\tau})$. For each vertex $u \in V$, and for $-p \leq i \leq i^*(u)$, define $M_i(u)$ to be the set of vertices w such that $w \in N_i(u)$ and (w,u) is directed into u. Define $I = \{i \,|\, \exists v \in M_i(u)\}$. Suppose the elements of I are listed in increasing order $i_1 < i_2 < \cdots$; for brevity, we write $M_j^u := M_{i_j}(u)$.

We now keep all the arcs directed *out of* u. Moreover, for $1 \leq j \leq l$, we keep the arcs directed from M_j^u *into* u. For $j > l$, we pick an arbitrary vertex $w \in M_{j-l}^u$ and replace every arc from M_j^u into u by an arc from M_j^u into w. Let $(V, \tilde{E})$ be the resulting undirected graph. Since every edge in $\hat{E}$ is either kept or replaced by another edge (which might be already in $\hat{E}$), $|\tilde{E}| \leq |\hat{E}|$. The following lemma, whose proof is omitted, gives the claimed result:

LEMMA 5.2. *Every vertex in $(V, \tilde{E})$ has degree bounded by $(2+\frac{1}{\tau})^{O(\alpha)}$. Furthermore, if $\tilde{d}$ is the metric induced by $(V, \tilde{E})$, then $\tilde{d} \leq (1+4\tau)\hat{d}$.*

References

[1] Noga Alon and Joel Spencer. *The Probabilistic Method*. Wiley Interscience, New York, 1992.

[2] S. Arya, G. Das, D. M. Mount, J. S. Salowe, and M. H. M. Smid. Euclidean spanners: short, thin, and lanky. *STOC*, pp. 489–498, 1995.

[3] B. Awerbuch and D. Peleg. Routing with polynomial communication-space trade-off. *SIAM J. Discrete Math.*, 5(2):151–162, 1992.

[4] Y. Bartal. Probabilistic approximations of metric spaces and its algorithmic applications. In *FOCS*, pp. 184–193, 1996.

[5] J. Beck. An algorithmic approach to the Lovász local lemma. I. *Random Struct. Alg.*, 2(4):343–365, 1991.

[6] L. J. Cowen. Compact routing with minimum stretch. *J. Algorithms*, 38(1):170–183, 2001.

[7] G. Das, G. Narasimhan, and J. Salowe. A new way to weigh malnourished Euclidean graphs. In *SODA*, pp. 215–222, 1995.

[8] M. M. Deza and M. Laurent. *Geometry of cuts and metrics*. Springer-Verlag, Berlin, 1997.

[9] G. N. Frederickson and R. Janardan. Designing networks with compact routing tables. *Algorithmica*, 3:171–190, 1988.

[10] G. N. Frederickson and R. Janardan. Efficient message routing in planar networks. *SICOMP*, 18(4):843–857, 1989.

[11] Cyril Gavoille. Routing in distributed networks: Overview and open problems. *ACM SIGACT News*, 32(1):36–52, 2001.

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

[13] Anupam Gupta, Robert Krauthgamer, and James R. Lee. Bounded geometries, fractals, and low–distortion embeddings. In *FOCS*, pp. 534–543, 2003.

[14] S. Har-Peled and M. Mendel. Fast constructions of nets in low dimensional metrics, and their applications. manuscript, 2004.

[15] J. Heinonen. *Lectures on analysis on metric spaces*. Springer-Verlag, New York, 2001.

[16] K. Hildrum, J. D. Kubiatowicz, S. Rao, and B. Y. Zhao. Distributed object location in a dynamic network. In *SPAA*, pp. 41–52, 2002.

[17] David R. Karger and Matthias Ruhl. Finding nearest neighbors in growth-restricted metrics. In *STOC*, pp. 63–66, 2002.

[18] Leonard Kleinrock and Farouk Kamoun. Hierarchical routing for large networks. Performance evaluation and optimization. *Comput. Networks*, 1(3):155–174, 1976/77.

[19] Robert Krauthgamer and James R. Lee. The intrinsic dimensionality of graphs. In *STOC*, pp. 438–447, 2003.

[20] M. Molloy and B. Reed. *Graph colouring and the probabilistic method*. Springer-Verlag, Berlin, 2002.

[21] D. Peleg and E. Upfal. A trade-off between space and efficiency for routing tables. *JACM*, 36(3):510–530, 1989.

[22] D. Peleg. *Distributed computing*. SIAM, Phila., PA, 2000.

[23] 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.

[24] R. Rajaraman, A. W. Richa, B. Vöcking, and G. Vuppuluri. A data tracking scheme for general networks. In *SPAA*, pp. 247–254, 2001.

[25] Kunal Talwar. Bypassing the embedding: Algorithms for low-dimensional metrics. In *STOC*, pp. 281–290, 2004.