On Hierarchical Routing in Doubling Metrics

Anupam Gupta, Bruce M. Maggs, Shuheng Zhou

CMU-PDL-04-106
December 2004

Parallel Data Laboratory
Carnegie Mellon University
Pittsburgh, PA 15213-3890

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 a 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.) For a connected graph G, whose shortest path distances* d_G *induce the doubling metric* (X,d_G)*, we show how to perform* $(1+\tau)$*-stretch routing on G for any* $0<\tau\leq 1$ *with routing tables of size at most* $(\alpha/\tau)^{O(\alpha)}\log\Delta\log\delta$ *bits with only* $(\alpha/\tau)^{O(\alpha)}\log\Delta$ entries*, where* Δ *is the diameter of G and* δ *is the maximum degree of G. 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 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. 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.

Keywords: doubling metrics, bounded growth, hierarchical routing, decomposition

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 $R/2$ [12]. 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 G. Each entry stores at most $O(\log\delta)$ bits, where δ is the maximum degree of G, and hence for doubling metrics—where α is a constant—and any $\tau \leq 1$, we have $(1+\tau)$-stretch routing with only $O(\log\Delta \log\delta)$ 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., [16]), and has recently been used in several contexts in the literature on object location in peer-to-peer networks [21, 15, 14]. 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 [23], 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, whose shortest path distances d_G induce the doubling metric (X,d_G) 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 $(\frac{\alpha}{\tau})^{O(\alpha)} \log\Delta\log\delta$ bits of routing information, where Δ is the diameter of G and δ is the maximum degree of G.*

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 and 5.)

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 analysis of this process uses the Lovász Local Lemma (much as in [17, 12]); details are given in Section 3.

1.1 Related Work Distributed packet routing protocols have been widely studied in the theoretical computer science community; see, e.g., [8, 9, 2, 19, 6, 20], or the survey by Gavoille [10] 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 [11]; 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 [23], 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 $O(\log\Delta\log\delta)$ 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 [16]. 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 [21, 15, 14] have also used restrictions similar to [16] 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. [21] 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 $\dim(X) \leq 4\beta$ [12, 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. [22], who use approximations by tree distributions to obtain bounds on the stretch incurred.

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 [7] and [13]. 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.,[12]) *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 $h = \left\lceil \log_{\rho} \Delta \right\rceil$. 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 $0 \leq i \leq h+1$.

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* $h = \left\lceil \log_\rho \Delta \right\rceil$ *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 $\leq 2\eta_i$. 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 $0 \leq i \leq h-1$.

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 [3]) $T_{\mathbf{\Pi}}$ 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. ([12]) *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)** $\forall x \in X$, $\Pr_\mu \left[d(x, X \setminus C_\Pi(x)) \geq \varepsilon r \right] \geq \frac{1}{2}$.

(This is called a padded probabilistic decomposition in [12].) 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 $\geq 1/2$.

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 $\geq \frac{1}{2}$.

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 [4] 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 [23] with better guarantees; the proof basically follows from the padded decompositions given in [12]. 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.

---

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

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 $\Lambda \geq \alpha$, 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 $[d(v,x) - r - \Gamma, d(v,x) + r + \Gamma]$. 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 $\mathbf{\Pi}^{(1)},\dots,\mathbf{\Pi}^{(M)}$, and let the PPHD μ_M be the uniform distribution on these HDs. By (3.2), for each $j \in [1\dots M]$, point $x \in X$ and radius $r \le \rho^i$, $\mathbf{B}(x,\varepsilon r)$ 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 [12] and [23]) 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 [12, 17].)

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 [12]. Instead of building one hierarchical decomposition $\mathbf{\Pi}$ bottom-up, we build m hierarchical decompositions $\mathbf{\Pi}^{(1)},\dots,\mathbf{\Pi}^{(m)}$ simultaneously (also from the bottom up).

As before, the proof proceeds inductively; we assume that we are given level-i partitions $\Pi_i^{(1)},\dots,\Pi_i^{(m)}$, 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 $\Pi_{i+1}^{(1)},\dots,\Pi_{i+1}^{(m)}$ 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 $\rho^i \le r \le \rho^{i+1}$ 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 $J = \{1,\dots,m\}$. 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* $\mathcal{E}_z^{i+1}$, $\Pr[\mathcal{E}_z^{i+1}] \le p$. *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; $ep(B+1)$ 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, 4]). 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 $0 < \tau \le 1$, a $(1+\tau)$-stretch routing scheme which uses routing tables of size at most $m(\alpha/\tau)^{O(\alpha)} \log \Delta \log \delta$ 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 $h = \left\lceil \log_\rho \Delta \right\rceil$. 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 $J = \{1, \dots, m\}$. 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 $h = \left\lceil \log_\rho \Delta \right\rceil$ 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 $m \times h \times \lceil \log(\deg(\mu_m)) \rceil = O(\alpha \log \alpha) \times \left\lceil \log_\rho \Delta \right\rceil \times O(\alpha \log \alpha) = O(\alpha^2 \log \alpha \log \Delta)$ 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 $\eta_{i-\ell} \leq \frac{\varepsilon\tau}{4}\rho^{i-1}$.

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 $d(x, \mathsf{close}_x(C')) \leq \varepsilon\rho^{i'+\ell}$, 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, which we do not store in Route_x since we can deduce it from $\mathsf{addr}(x)$ based on the clustering structure); each entry indeed contains the identity of the next hop (which uses $O(\log\delta)$ bits, where δ is the maximum degree of G), a path length field (to be specified in Section 5.1), 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.

---

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 $\langle \mathsf{addr}(t), \mathsf{addr}(C') \rangle$ to y.
6. Else (now $x \notin C$)
7. let $y \leftarrow \mathsf{NextHop}_x(\mathsf{addr}(C))$
8. forward packet with unchanged header $\langle \mathsf{addr}(t), \mathsf{addr}(C) \rangle$ to y.
9. End

Figure 4.2: The Forwarding Algorithm at Node x

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

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'.

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* $(i-\ell)$ *or lower in some* $T_{j'}$ *with* $\mathtt{ValidPath}_x(C')$ *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 $\mathtt{ValidPath}_x(C_t)$ 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 $\mathtt{ValidPath}_x(C')$ 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 $\mathtt{ValidPath}_x(C')$ 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'. ■

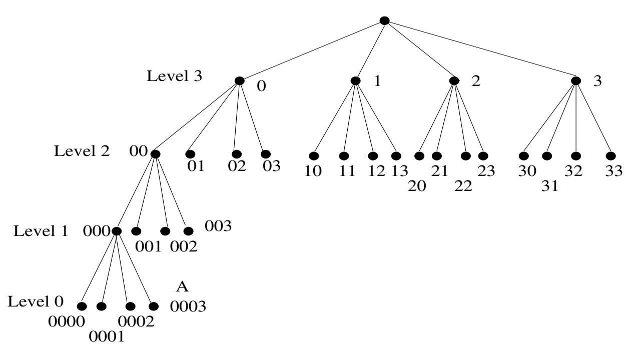
It remains to bound the path stretch. Consider the case when a packet is sent from s to t. Let C' be a cluster at level $i-\ell$ returned by Step 3 of the forwarding algorithm. Note that if the level $i \leq \ell$, then $C' = \{t\}$ and we send the packet directly to t with $\tau = 0$. Using these short 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 $\eta_{i-\ell} \leq \tau\varepsilon\rho^{i-1}/4$. ■

5 Routing Table Construction and Path Characteristics for $(1+\varepsilon)$-Stretch Routing

The hierarchical routing scheme we are going to describe in this section is a completion of what is lacking in Section 4; hence we focus primarily the process of building up routing tables using a distributed implementation of Bellman-Ford algorithm for the base model that we introduce in Section 5.1. For overlay routing, we store the IP address of an intermediate node to reach each destination in the routing tables and the process of routing table updates are similar to that of prefix routing, e.g., in [14]. Although the Forwarding algorithm remains the same as that in Section 4.3, we will elaborate in more details on its behavior in Section 5.2 when it is coupled with the new routing algorithm.

Our routing scheme is similar in spirit to that of Closest Entry Routing (CER) scheme described in Kleinrock and Kamoun (KK) [16]. They define a hierarchical routing scheme by first specifying an "optimal" underlying hierarchical clustering structure that they impose on the network nodes, where the optimization objective is to minimize the routing table length; each level-k cluster is defined recursively as a set of level-$(k-1)$ clusters, with the level-0 clusters being individual nodes. This leads naturally to a tree representation as shown in Figure 5.3 (a), where internal tree nodes represent clusters; Figure 5.3 (b) shows that the destination addresses in the routing table of node A corresponds to clusters at different levels of the decomposition tree, hence reflecting the structure of the hierarchical clustering of network nodes. In KK, two nodes share common routing table entries for all the clusters that contain both of them. KK assumes that all clusters at the same level have the same number of sub-clusters within them, and each cluster is a connected component. The KK hierarchical routing procedure leads a message down a tree path, fixing more prefix digits at each step, much as prefix routing, traversing smaller and smaller clusters that contain the destination node until it reaches the destination itself.



(a) A tree representation

Level 3	0***	1***	2***	3***
Level 2	00**	01**	02**	03**
Level 1	000*	001*	002*	003*
Level 0	0000	0001	0002	0003

(b) Routing table entries in node A

Figure 5.3: A 4-level hierarchical clustering structure of network nodes

The reduction of routing table size generally leads to an increase in network path length. In order to derive bounds on the increase in the average path length, they further assume that a shortest-path between two nodes in a cluster lies within the cluster. They also prescribe an upper-bound of d_k on the (strong) diameter of a k^{th} level cluster, with d_k decreasing as k decreases. They show that routing schemes based on the hierarchical clustering model cause essentially no increase in the *average network path length* for a family of large distributed networks. Specifically, the networks they consider are all connected graphs upon which it is possible to fit a hierarchical clustering whose outcome satisfies the assumptions above. In addition, (a) the resulting clusters at any level satisfy the following: 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 $\Theta(N^{\nu})$, where N is the size of such a network.

In contrast, our hierarchical routing schemes give bounds on the path stretch on a *per node-pair* level on certain networks that are connected graphs G, where the natural metric (X,d) induced by shortest path distances between any pair of nodes in G is a doubling metric. In addition, the main improvement our work over that of KK is: while the KK routing scheme is based on assumptions regarding the existence of a "good" partition of the network, the method itself does not provide an algorithm for computing such a partition; we are able to prove the existence of a (ρ,ε)-PPHD with a support on m Hierarchical Decompositions and actually find them by following the Clustering algorithm and its constructive algorithm described in Section 3. Note that while we guarantee a degree bound for the decomposition trees across all levels, we do not require they are exactly the same.

It would be ideal if once we construct such a set of network partitions, we can run the hierarchical routing algorithm specified in KK at each individual decomposition tree. However, it is not possible to directly apply KK's routing scheme or their proof techniques for three reasons. First, while KK assumes that each cluster subnetwork is fully connected, this is not satisfied in our decomposition. Second, the shortest paths between two nodes in a cluster are not guaranteed to stay within the cluster. Finally, although the maximal distance in G between vertices of C_k, for all $0 \le k \le h$, is bounded within the diameter of C_k, $2\eta_k$, which is geometrically decreasing as k decreases, it is a weak diameter bound and not necessarily satisfied by the distance induced by the subgraph corresponding to each cluster C_k.

We thus adopt as many definitions and notations as possible from KK in this section while inventing some new techniques for addressing the above issues in the design and specification of a modified hierarchical routing scheme given a (ρ,ε)-PPHD μ_m with a support on m HDs and in the analysis of the characteristics of paths as induced by the routing tables thus created. The important property of a (ρ,ε)-PPHD that we will use in defining our routing scheme is that, for $\rho^{i-1} < r \le \rho^i$, there is at least one tree T_j such that $\mathbf{B}(s,\varepsilon r)$ is contained in a level i cluster C_i in the level-i partition $\Pi_i^{(j)}$; since a ball is a connected component, all shortest paths from s to vertices within $\mathbf{B}(s,\varepsilon r)$ must be contained within C_i in the level-i partition $\Pi_i^{(j)}$.

5.1 Routing Table Update Rules In this section, we focus on the process of building up routing tables once the nodes in the network have been assigned addresses that reflect their positions in each of the m decomposition trees. During this process, routing information is aggregated and exchanged between special nodes in different clusters at each level. We refer to such special nodes as exchange nodes (for routing) or entry points (for packet forwarding) of their corresponding clusters. The algorithm for selecting exchange nodes for each cluster is an independent issue that we do not address in this paper. Similar to the CER hierarchical routing scheme described in KK, no routing information describing the internal behavior of a cluster is propagated outside a cluster; hence a cluster is regarded from outside as a single node whose distance to itself is zero.

We use a modified version of the distributed Bellman-Ford algorithm as in Fig 5.4 to perform routing updates: especially, to establish the next-hop entries and update estimated path lengths for destination clusters in the routing tables for the basic model. For routing updates, we are going to focus on entries for one specific decomposition tree T_j that corresponds to $\mathbf{\Pi}^{(j)} = (\Pi_i^{(j)})_{i=0}^h$.

Let s and t be two neighboring nodes (that they are connected by a channel (s,t)) which belong to the same k^{th} level cluster $C_k \in \Pi_k^{(j)}$ and not to any lower level cluster in T_j, where $k \in \{1,2,\dots,h\}$. Let $C_{k-1}(s), C_{k-1}(t) \in \Pi_{k-1}^{(j)}$ respectively denote the $k-1^{st}$ level clusters to which s and t each belong in tree T_j. Let $C_k(s,t)$ denote the level-k cluster that contains both s and t; note that $C_{k-1}(s), C_{k-1}(t) \subseteq C_k(s,t)$ in T_j since T_j represents a laminar decomposition. We use $\mathsf{lca}^j(s,t)$ to denote the lowest common ancestor of s and t in a particular tree T_j; hence $\mathsf{lca}^j(s,t) = C_k(s,t) \in \Pi_k^{(j)}$. For a pair of nodes s, t, $\mathsf{lca}^j(s,t)$ can be determined by inspecting the common prefixes of local addresses, $\mathsf{addr}_j(s)$ and $\mathsf{addr}_j(t)$.

Recall that in node s, for any cluster $C_i(s)$ in T_j that contains s at level i, for all $i = 0,\dots,h$, routing table entries are kept for all clusters that are descendants of $C_i(s) \in \Pi_i^{(j)}$ within ℓ levels down a decomposition tree for $T_j, \forall j$. Thus each entry in the routing table Route_s for T_j corresponds to some level-(i') cluster $C' \in \Pi_{i'}^{(j)}$ in T_j, where $i' = 0,1,\dots,h-1$; that entry is also denoted as C' and indexed by the global address $\mathsf{addr}(C')$ of its associated cluster C', and contains the following fields in Route_s: **(a)** a next hop $\mathsf{NextHop}_s(\mathsf{addr}(C'))$ to reach C' from s, **(b)** a path length field $\mathsf{HF}(s,C')$ that is the current path length at node s for reaching cluster C' through $\mathsf{NextHop}_s(\mathsf{addr}(C'))$, and **(c)** a $\mathsf{ValidPath}_s(C')$ bit. Initially, the path length fields for all the entries in Route_s for tree T_j are set to ∞ except for the self entries as shown in the Initialization Procedure in Fig 5.4.

We use $C_i(s,C') = C_i(s) \in \Pi_i^{(j)}$ to denote the level-i common ancestor of s and $C' \in \Pi_{i'}^{(j)}$ such that $i \geq i'+1$ and $C_i(s) \supseteq C'$. Note that $C_h(s,C') = C_h(s)$ contains $C' \in \Pi_{i'}^{(j)}$, for all $i' \leq h-1$, since $C_h(s)$ contains the entire network. Similarly, we use $\mathsf{lca}^j(s,C')$ to denote the lowest common ancestor of s and $C' \in \Pi_{i'}^{(j)}$ in tree T_j, where $C' \subseteq \mathsf{lca}^j(s,C') \subseteq C_i(s,C')$ for all i such that $C' \subseteq C_i(s)$. For node s and cluster C', the $\mathsf{lca}^j(s,C')$ can be determined by inspecting the common prefixes of local addresses $\mathsf{addr}_j(s)$ and $\mathsf{addr}_j(C')$.

As a consequence of the routing table specification, routing table entries at node s and t at all levels below $k-\ell$ in T_j refer to different cluster destinations; whereas all the other entries from level $k-\ell$ up to h refer to the same cluster destinations in T_j. The objective of the updating procedure is to compare the estimated lengths of the paths from s or t to any common destination and to update the routing tables to reflect the shorter paths. Whenever s receives a route update from t, for each common destination cluster C', its corresponding entry is potentially updated with a new next hop $\mathsf{NextHop}_s(\mathsf{addr}(C'))$, the path length $\mathsf{HF}(s,C')$ through the new $\mathsf{NextHop}_s(\mathsf{addr}(C'))$ as in Step 2-4, and the $\mathsf{ValidPath}_s(C')$ bit as in Step 5-9 of the Route Update Procedure in Fig 5.4.

We have a slightly different way of setting the $\mathsf{ValidPath}_s(C')$ bit from that specified in Section 4.2 to maximize the chance of setting it `true`. However, as before, once the $\mathsf{ValidPath}_s(C')$ bit is set to be `true`, a shortest path from s to C' is indeed guaranteed by following the next hop in Route_s for an entry C' and that in Route_v of each subsequent nodes v along the path from s to an entry point of C'.

Let a common destination entry for T_j in Route_s and Route_t correspond to a level-(i') cluster $C' \in \Pi_{i'}^{(j)}$, where $i' \geq k-\ell$. We denote the level of $\mathsf{lca}^j(s,C')$ in T_j as l_0; The following inequalities, $i'+1 \leq l_0 \leq i'+\ell$, must be satisfied for C' to be an entry in Route_s. The $\mathsf{ValidPath}_s(C')$ bit is set to be `true` so long as for "any" of the common ancestor $C_i(s,C')$ of s and C' at level i, for all $l_0 \leq i \leq i'+\ell$, both $\mathsf{HF}(s,C') \leq \varepsilon\rho^i$ and $\mathbf{B}(s,\varepsilon\rho^i) \subseteq C_i(s,C')$ are true. It is set to be `false` otherwise. Note that when $i' \geq h-\ell$, both $\mathsf{HF}(s,C') \leq \Delta$ and $\mathbf{B}(s,\Delta) \subseteq C_h(s,C')$ are always true since $C_h(s,C')$ is the entire network; hence we set $\mathsf{ValidPath}_s(C')$ bit `true` for all C' at level $h-\ell$ and above in Step 5 of the Initialization Procedure.

The reason we set $\mathsf{ValidPath}_s(C')$ bit this way is the following. Recall that by constructing the m decomposition trees, each node s "knows" if $\mathbf{B}(s,\varepsilon\rho^i)$ is contained $C_i(s) \in \Pi_i^{(j)}$ in tree T_j; naturally, if $\mathbf{B}(s,\varepsilon\rho^i) \subseteq C_i(s) \in \Pi_i^{(j)}$, then $\mathbf{B}(s,\varepsilon\rho^i) \subseteq C_l(s) \in \Pi_l^{(j)}$ is true for all $l \geq i$. However, if $\mathbf{B}(s,\varepsilon\rho^i) \not\subseteq C_i(s)$, we do not assume that we know information such as "whether a ball $\mathbf{B}(s,r)$ of a radius $\varepsilon\rho^i > r > \varepsilon\rho^{(i-1)}$ is

Initialization Procedure: initialize Route_s for tree T_j at node s

1. For $i = 0, 1, \ldots, h$
2. $\quad$ $\mathsf{HF}(s, C_i(s)) = 0$, and $\mathsf{ValidPath}_s(C_i(s)) = \mathtt{true}$
3. For all other entries $C' \not\ni s$, let i' = level of C' in tree T_j
4. $\quad$ $\mathsf{HF}(s, C') = \infty$
5. $\quad$ If $i' \geq h - \ell$, then $\mathsf{ValidPath}_s(C') = \mathtt{true}$
6. End

Route Update Procedure: upon receiving a route update from t such that $\mathsf{lca}^j(s,t) = C_k$

1. For each common entry $C' \in \Pi_{i'}^{(j)}$, which represents a level-(i') cluster in T_j, where $i' \geq k - \ell$
2. $\quad$ If $\mathsf{HF}(s, C') > d(x,t) + \mathsf{HF}(t, C')$, then
3. $\quad\quad$ $\mathsf{HF}(s, C') \leftarrow d(x,t) + \mathsf{HF}(t, C')$
4. $\quad\quad$ nexthop field of $C' \leftarrow t$
5. $\quad\quad$ If $i' < h - \ell$, then
6. $\quad\quad\quad$ Let l_0 = level of $\mathsf{lca}^j(s, C')$ in T_j and m satisfies $\varepsilon\rho^{m-1} \leq \mathsf{HF}(s, C') \leq \varepsilon\rho^m$
7. $\quad\quad\quad$ for all levels $i : \max\{l_0, m\} \leq i \leq i' + \ell$
8. $\quad\quad\quad\quad$ If $\mathbf{B}(s, \varepsilon\rho^m) \subseteq \mathbf{B}(s, \varepsilon\rho^i) \subseteq C_i(s)$ in T_j, then
9. $\quad\quad\quad\quad\quad$ $\mathsf{ValidPath}_s(C') = \mathtt{true}$
10. $\quad\quad\quad\quad\quad$ Goto 1
11. End

Figure 5.4: DISTRIBUTED BELLMAN-FORD Algorithm for T_j at Node s

contained in $C_i(s)$ or not", since that is not the type of information that our constructive algorithm provides by default; note that if $r \leq \varepsilon\rho^{(i-1)}$, we will just check if $\mathbf{B}(s, \varepsilon\rho^{(i-1)}) \subseteq C_{i-1}(s)$ to decide if $\mathbf{B}(s,r) \subseteq C_i(s)$. Our routing algorithm thus makes minimal assumptions about the information that is available at each node about balls around it being contained at a certain level or not.

Another specification in terms of routing that is different from that of Section 4.2 is the following. Assume we route a packet from s toward C'. Instead of assuming the packet should always enter a cluster C' through the closest point $x = \mathsf{close}_s(C')$ in C' to s, we only require that the packet enters C' through a closest *entry* point $e_0 \in C'$. Correspondingly, for node s and a level-(i') cluster $C' \in \Pi_{i'}^{(j)}$ in T_j, the function $\mathsf{NextHop}_s(\mathsf{addr}(C'))$ returns the next hop on the path from s to e_0 provided that the next hop does not leave the cluster C at level $(i' + \ell)$ that contains C', and null otherwise. Recall an entry point $e_0 \in C'$ advertises routes for C' it belongs to. Note also e_0 does not need to be the closest one to s in C' in order to achieve $(1+\tau)$-stretch routing. (This is also true for overlay routing.) As a basic routing scheme, we keep a next hop $\mathsf{NextHop}_s(\mathsf{addr}(C'))$ in Route_s toward a closest entry point $e_0 \in C'$ for the sake of routing table consistency that we will elaborate shortly.

For overlay routing, we keep the IP address of an arbitrary entry point e_0 to C' (instead of a next hop $\mathsf{NextHop}_s(\mathsf{addr}(C'))$ toward e_0), since IP routing will deliver a packet from s to e_0 directly given the IP address of e_0 without having to rely on hop-by-hop forwarding as in the basic model that we focus in this section.

DEFINITION 5.1. *We call a path an internal path in cluster C if all the nodes in that path belong to C.*

Similar to KK, we define the *equilibrium* condition as the situation when no changes occur in the topology of network and the contents of $\mathsf{HF}(s, C')$ in the routing table reach "minimal" constant values after a certain number of updates.

CLAIM 5.2. *The distributed Bellman-Ford algorithm guarantees that in equilibrium condition, $\mathit{HF}(s,C')$ will be the length of the shortest path from s to a closest entry point e_0 of C' when* $\mathsf{ValidPath}_s(C')$ *is* `true`*, i.e.,* $\mathit{HF}(s,C') = d(s,e_0)$ *in* Route_s.

Proof. Let the level of $C' \in \Pi_{i'}^{(j)}$ in tree T_j be $i' < h-\ell$ and let the level of $\mathsf{lca}^j(s,C')$ be l_0. We only set $\mathsf{ValidPath}_s(C')$ `true` in the routing algorithm when for "any" of the level-i cluster $C_i(s) \in \Pi_i^{(j)}$, where $l_0 \le i \le i'+\ell$, both $\mathsf{HF}(s,C') \le \varepsilon\rho^i$ and $\mathbf{B}(s,\varepsilon\rho^i) \subseteq C_i(s) \in \Pi_i^{(j)}$ hold. Denote the lowest such level r, where $r \in [l_0, i'+\ell]$. All shortest paths from s to some entry point $x' \in C'$ of distance $d(s,x') \le \mathsf{HF}(s,C') \le \varepsilon\rho^r$ are thus internal to $C_r(s) \in \Pi_r^{(j)}$ in T_j, since such paths are contained in $\mathbf{B}(s,\varepsilon\rho^r)$, which is a connected component entirely contained in $C_r(s) \in \Pi_i^{(j)}$. Note that some $x' \in C'$ must have advertised itself as an entry point to C' for such paths to be established within $\{C_r(s) - C'\}$ and for $C' \in \Pi_{i'}^{(j)}$ to appear in Route_s. Thus $C' \subseteq C_r(s)$ since $x' \in \{C' \cap C_r(s)\} \neq \emptyset$ and $r > l_0$; we thus denote $C_r(s)$ as $C_r(s,C')$ from this point on.

In addition, every node $v \in C_r(s,C')$, including those along the shortest paths from s to x' inside $\mathbf{B}(s,\varepsilon\rho^r)$, contains a routing table entry to C', since it is a descendant of $C_r(s,C')$ within ℓ levels down the decomposition tree T_j. Propagation and subsequent updating of routing information among nodes of $C_r(s,C')$ is equivalent to finding minimum path internal to $C_r(s,C')$ from any node $v \in \{C_r(s,C') - C'\}$ to an entry point of C' that is closest to node v; for s, the closest entry point to C' is e_0.

Improvements are made sequentially at each update over the distance $\mathsf{HF}(u,C')$ from u to C' among nodes within $\mathbf{B}(s,\varepsilon\rho^r)$, until it reaches a minimal constant value if no changes occur in the topology of the network; hence all $u \in \mathbf{B}(s,\varepsilon\rho^r)$ "knows" how to route to C' with a path of bounded length. Given multiple entry points to C', the distributed Bellman-Ford algorithm guarantees that we find a shortest path not only to some entry point x' of C', but also to the closest, e_0 of C', from s in equilibrium condition, i.e., $\mathsf{HF}(s,C') = d(s,e_0)$. The entire path stays within $\mathbf{B}(s,\varepsilon\rho^r) \subseteq C_r(s,C')$, where r is specified as above.

Note that when $i' \ge h-\ell$, both $\mathsf{HF}(s,C') \le \Delta$ and $\mathbf{B}(s,\Delta) \subseteq C_h(s,C')$ are always true since $C_h(s,C')$ is the entire network; hence we set $\mathsf{ValidPath}_s(C')$ `true` for all C' at level $h-\ell$ and above. The same argument as above applies to this case. ■

The reason we require a closest entry point to C' is primarily for route convergence purpose when our protocol serves as an underlying routing scheme. For overlay routing, we allow an entry point to be any exchange node or simply a random node within the cluster, which is commonly assumed in peer-to-peer networks. Note that an exchange node of a given cluster is a node of that cluster which is connected to one or more nodes external to that cluster as defined in KK. We will use exchange node and entry point interchangeably unless we specify otherwise. The $(1+\tau)$-stretch property we are going to prove for hierarchical routing paths does not require the entry point for a cluster C' to be the closest to s either – a point that we will not elaborate on from now on.

FACT 5.3. *If a shortest path from s to e_0, an entry point to a level-(i') cluster $C' \in \Pi_{i'}^{(j)}$, is internal to $C_i(s) \in \Pi_i^{(j)}$ in tree T_j, where $i > i'$, then cluster $C' \ni e_0$ must be a sub-cluster that is entirely contained in $C_i(s)$ in T_j, i.e., $C' \subseteq C_i(s)$.*

Proof. First observe $e_0 \in \{C_i(s) \cap C'\}$, since shortest path from s to e_0 is internal to $C_i(s) \in \Pi_i^{(j)}$ in T_j. Since T_j represents a laminar decomposition, where a lower level cluster is always entirely contained in a higher level cluster, $e_0 \in C_i$ is sufficient to guarantee that $\{C' \ni t\} \subseteq C_i$. ■

5.2 Path Characteristics We forward packets according to the Forwarding algorithm in Figure 4.2. Let s and t be two arbitrary nodes. For destination t, let $C'(t)$ be the cluster whose $\mathsf{addr}(C'(t))$ is returned by the function $\mathsf{PrefMatch}_s(t)$ at Step 3 of the Forwarding algorithm. We assume, w.l.o.g., $C'(t) \in \Pi_{i'}^{(j)}$, i.e., $C'(t)$

is in the level-(i') partition $\Pi_{i'}^{(j)}$, where $i' \leq h - \ell$, in tree T_j. Recall that $h = \lceil \log_\rho \Delta \rceil$. Let $l_0 \leq h$ be the level of $\mathsf{lca}^j(s, C'(t)) \in \Pi_{l_0}^{(j)}$ in T_j.

We say $C'(t) \ni t$ is the cluster that has the longest *valid* prefix matching with t in Route_s, since the level of $C'(t)$ is the lowest across all trees among clusters C' in Route_s such that $C' \ni t$ and $\mathsf{ValidPath}_s(C')$ is `true`. Before we proceed, we first give more definitions, some of which are adapted from KK.

h_{st}^c: Length of the estimated minimum path from node s to node t as derived from the routing information at node s. (The superscript c stands for clustered routing.)

Exchange node e_z: a node of a cluster C that is connected to one or more nodes external to C.

$A_i(t)$: Subset of all exchange nodes (entry points) that connect a level-i cluster $C_i(t) \in \Pi_i^{(j)}$ in tree T_j, for all $j = 1, \ldots, m$, with any other level-i cluster within the same ancestor $C_n(t) \in \Pi_n^{(j)}$ in the same tree T_j, for all $n \leq i + \ell$. From the above definitions, all entry points e_z of $C'(t) \in \Pi_{i'}^{(j)}$ that connect $C'(t)$ to any other level-(i') cluster that stays within $C_{i'+\ell}(t) \in \Pi_{i'+\ell}^{(j)}$ in tree T_j hence belong to $A_{i'}(t)$.

Let $e_0 \in A_{i'}(t) \cap C'(t)$ be the closest entry point for s to reach $C'(t) \in \Pi_{i'}^{(j)}$ in T_j.

$\hat{C}_k(s,t)$: For $k \leq h-1$, $\hat{C}_k(s,t) \in \Pi_k^{(j)}$ is the level-k cluster in T_j, where $l_0 \leq k \leq i' + \ell$, that is the lowest-level common cluster of s and t such that $\mathbf{B}(s, \varepsilon\rho^k) \subseteq \hat{C}_k(s,t)$ and $\mathbf{B}(s, \varepsilon\rho^k)$ contains a shortest path from s to $C'(t) \in \Pi_{i'}^{(j)}$ in T_j, where $i' \leq h - \ell$; such $\hat{C}_k(s,t) \in \Pi_k^{(j)}$ always exists since we know $\mathbf{B}(s, \varepsilon\rho^r) \subseteq C_r(s,t)$ and $\mathsf{HF}(s, C'(t)) \leq \varepsilon\rho^r$ must both hold for some $l_0 \leq r \leq i' + \ell$, given that $\mathsf{ValidPath}_s(C'(t))$ is `true` in Route_s, due to the specification of the distributed Bellman-Ford algorithm. Let k be the lowest such level r. Note that $C'(t) \subseteq \hat{C}_k(s,t)$ since T_j represents a laminar decomposition and k is at least l_0. For $k = h$, $\hat{C}_k(s,t) = C_h(s,t) \in \Pi_h^{(j)}$, is the root cluster X of T_j that corresponds to the entire network G. In this case, $\hat{C}_k(s,t) = C_h(s,t)$ always contains all shortest paths from s to $C'(t) \in \Pi_{i'}^{(j)}$ in T_j, where $i' = h - \ell$, given that G is a connected graph.

$h_{se_z}^i(t)$: Length of the shortest path from node s to an exchange node $e_z \in A_{i'}(t) \cap C'(t)$ as contained in $\hat{C}_k(s,t)$ defined above. The superscript i stands for an internal path within $\hat{C}_k(s,t)$. At equilibrium, $h_{se_0}^i(t) = \mathsf{HF}(s, C'(t)) = d(s, e_0)$ since the shortest path from s to e_0 is internal to $\hat{C}_k(s,t)$, and by Claim 5.2, $\mathsf{HF}(s, C'(t)) = d(s, e_0)$ in Route_s given that $\mathsf{ValidPath}_s(C'(t))$ is `true` in Route_s and e_0 is the closest entry point to $C'(t)$ for node s. Recall that $\mathsf{HF}(s, C'(t))$ is the current path length filed in Route_s for node s to reach $C'(t) \in \Pi_{i'}^{(j)}$ via its current $\mathsf{NextHop}_s(\mathsf{addr}(C'(t)))$. Note when the shortest path from s to e_z is not internal to $\hat{C}_k(s,t)$, we denote it with $h_{se_z}^i = \infty$.

In order to reach t, function $\mathsf{PrefMatch}_s(t)$ is called by the Forwarding algorithm at node s, which looks across Route_s for all trees and picks a tree T_j that contains $C'(t)$ with a closest entry point $e_0 \in A_{i'}(t) \cap C'(t)$. Node s then stores $\langle \mathsf{addr}(t), \mathsf{addr}(C'(t)) \rangle$ in the packet header and sends the packet to $\mathsf{NextHop}_s(\mathsf{addr}(C'(t)))$; the packet header remains the same while intermediate nodes v forward the packet along a shortest path from s to e_0, that is contained in the common cluster $\hat{C}_k(s,t)$ of s and t in T_j, until it reaches e_0.

The key observation we have regarding a path h_{st}^c from s to t is the following. The path may not be contained within the lowest common ancestor $\mathsf{lca}^j(s,t) \in \Pi_{l_0}^{(j)}$ of s and t in a particular tree T_j. However, the segment from s to $C'(t)$, is contained within $\hat{C}_k(s,t)$ in T_j, where $l_0 \leq k \leq i' + \ell$, when following a shortest path from s to e_0, which is the closest entry point to $C'(t)$. Recall $\hat{C}_k(s,t)$ is a common cluster of s and $C'(t)$ at a level higher than that of $\mathsf{lca}^j(s,t)$. Conceptually, we route packets from s to t within $\hat{C}_k(s,t) \in \Pi_k^{(j)}$ to avoid being stuck in $\mathsf{lca}^j(s,t) \in \Pi_{l_0}^{(j)}$, which may not contain any path (e.g., when $\mathsf{lca}^j(s,t)$ in T_j is disconnected) or contains only very long paths from s to t. The shortest path from s to e_0 is thus an internal path relative to $\hat{C}_k(s,t)$, which we denote with $h_{se_0}^i$.

Finally, We define a constant $\phi = \frac{4}{\rho^\ell \varepsilon}$ that we will use throughout this section. It is easy to verify that $2\eta_{i-\ell} \leq \frac{2}{\rho^{\ell-1}(\rho-1)\varepsilon}\varepsilon\rho^i < \phi\varepsilon\rho^i$. Recall that $\ell = \Theta(\log_\rho 1/\varepsilon\tau)$ and $\rho = \Theta(\frac{1}{\varepsilon})$, where we choose suitable constants so that $\rho^2 \leq \frac{1-\phi}{\phi}$ is satisfied. The rest of this section is dedicated to the proof of the main theorem of this section, before which we first prove two lemmas regarding the level of $C'(t)$ and $\hat{C}_k(s,t)$ given $d(s,t)$. Note that we always have $k \leq h$ and $i' \leq h-\ell$. We will ignore the case when $k = h$ until the end of this section.

LEMMA 5.1. *Let $d(s,t) \leq (1-\phi)\varepsilon\rho^i$, where $1 \leq i \leq h$. The cluster $C'(t) \in \Pi_{i'}^{(j)}$ in T_j that has the longest valid prefix matching with t with $\mathsf{ValidPath}_s(C'(t)) = \mathtt{true}$, is at a level $i' \leq \max(0, i-\ell)$; the common cluster $\hat{C}_k(s,t) \in \Pi_k^{(j)}$ as defined above that contains the shortest path from s to $C'(t)$ is at level $k \leq i$.*

Proof. We first prove the lemma when $i \leq \ell$ with the following claim.

Case $i \leq \ell$.

CLAIM 5.4. *Let $\varepsilon\rho^{i-1} < d(s,t) \leq \varepsilon\rho^i$ for $1 \leq i \leq \ell$. Then $C'(t) = C_0(t)$ is t itself; the lowest common cluster $\hat{C}_k(s,t)$ such that $\mathbf{B}(s,\varepsilon\rho^k) \subseteq \hat{C}_k(s,t)$ and $\mathbf{B}(s,\varepsilon\rho^k)$ contains the shortest path from s to $C_0(t)$, i.e., t itself, is at level $k = i$.*

Proof. Node s has a routing table entry for all t such that $d(s,t) \leq \varepsilon\rho^\ell$, since $\mathbf{B}(s,d(s,t)) \subseteq \mathbf{B}(s,\varepsilon\rho^\ell)$ is fully contained in some level-ℓ cluster $C_\ell(s) \in \Pi_\ell^{(q)}$ in some tree T_j, and $C'(t)$ is $C_0(t) \in \Pi_0^{(q)}$.

The properties of the (ρ,ε)-PPHD ensure that there is at least one tree T_j such that $\mathbf{B}(s,\varepsilon\rho^i) \subseteq C_i(s) \in \Pi_i^{(j)}$ in T_j. Since $d(s,t) \leq \varepsilon\rho^i$, we know that $t \in \mathbf{B}(s,\varepsilon\rho^i)$ and $C_0(t) \subseteq C_i(s)$ in T_j. The lowest common cluster $\hat{C}_k(s,t)$ such that $\mathbf{B}(s,\varepsilon\rho^k) \subseteq \hat{C}_k(s,t)$ and $\mathbf{B}(s,\varepsilon\rho^k)$ contains the shortest path from s to $C'(t) = C_0(t)$, i.e., t itself, is $C_i(s) \in \Pi_i^{(j)}$ in tree T_j and $k = i$. ■

We now prove the general case when $i > \ell$.

Case $h-1 \geq i > \ell$. Let $x' \in A_{i-\ell}(t)$ be an arbitrary entry point to some level-$(i-\ell)$ cluster $C \ni t$ in some tree; hence $d(x',t) \leq 2\eta_{i-\ell} \leq \phi\varepsilon\rho^i$ since $x',t \in C$. Applying the triangle inequality, we have $d(s,x') \leq d(s,t) + d(t,x') \leq \varepsilon\rho^i$; thus all shortest paths from s to x', for all $x' \in A_{i-\ell}(t)$, are contained in $\mathbf{B}(s,\varepsilon\rho^i)$.

The properties of the (ρ,ε)-PPHD ensure that there is at least one tree T_q such that $\mathbf{B}(s,\varepsilon\rho^i)$ is not cut in the level-i partition $\Pi_i^{(q)}$; let $C_i(s) \in \Pi_i^{(q)}$ be the level-i cluster in T_q such that $\mathbf{B}(s,\varepsilon\rho^i) \subseteq C_i(s)$. Since $d(s,t) \leq (1-\phi)\varepsilon\rho^i$, we have $t \in \mathbf{B}(s,\varepsilon\rho^i) \subseteq C_i(s) \in \Pi_i^{(q)}$. Let $C_{i-\ell}(t) \in \Pi_{i-\ell}^{(q)}$ be the level-$(i-\ell)$ cluster in T_q containing t; we know that $C_{i-\ell}(t) \subseteq C_i(s)$, since $t \in \{C_{i-\ell}(t) \cap C_i(s)\}$ and T_q represents a laminar decomposition. Hence we have $C_i(s) = C_i(t) = C_i(s,t)$ in the level-i partition $\Pi_i^{(q)}$ in tree T_q.

The $\mathsf{ValidPath}_s(C_{i-\ell}(t))$ bit must be set `true` in Route_s by the distributed Bellman-Ford algorithm in node s, since (a) $\mathbf{B}(s,\varepsilon\rho^i) \subseteq C_i(s,t) \in \Pi_i^{(q)}$, and (b) $\mathsf{HF}(s,C_{i-\ell}(t)) \leq \varepsilon\rho^i$ in Route_s for entry $C_{i-\ell}(t) \in \Pi_{i-\ell}^{(q)}$ in tree T_q at equilibrium, given that all shortest paths from s to an entry point x', for all $x' \in A_{i-\ell}(t) \cap C_{i-\ell}(t)$, are internal to $\mathbf{B}(s,\varepsilon\rho^i)$. Thus $\mathsf{PrefMatch}_s(t)$ can (and may indeed) just return $\mathsf{addr}(C_{i-\ell}(t))$ given no "better" choices, in which case, $i' = i-\ell$ and $k \leq i$.

However, $\mathsf{PrefMatch}_s(t)$ always finds a cluster $C'(t) \in \Pi_{i'}^{(j)}$ at the *lowest* level across all trees, such that $t \in C'(t)$ and $\mathsf{ValidPath}_s(C'(t))$ is `true` in Route_s; hence $C'(t)$ is at level $i' \leq i-\ell$.

We know that $\mathbf{B}(s,\varepsilon\rho^r) \subseteq C_r(s,t) \in \Pi_r^{(j)}$ and $\mathsf{HF}(s,C'(t)) \leq \varepsilon\rho^r$ must both hold, for some $l_0 \leq r \leq i'+\ell$, in order for $\mathsf{ValidPath}_s(C'(t))$ bit to be `true`, due to the specification of the distributed Bellman-Ford algorithm. Let k be the lowest such r; we have $k \leq i'+\ell \leq i$ for $i > \ell$.

Case $i = h$. We have $k \leq h$ and $i' \leq h-\ell$ trivially, since both holds for all possible distances of $d(s,t)$ up to Δ, which is the diameter of the network G. ■

CLAIM 5.5. *When $C'(t) \in \Pi_1^{(j)}$ is at level* 1*, $d(s,t) > \varepsilon\rho^{\ell}$.*

Proof. Prove by contradiction. Assume that $d(s,t) \leq \varepsilon\rho^{\ell}$. By Claim 5.4, we have $C'(t) = C_0(t)$, contradicting the assumption that $C'(t)$ is at level 1. ■

LEMMA 5.2. *Let $C'(t) \in \Pi_{i'}^{(j)}$ be the cluster returned by function $\mathsf{PrefMatch}_s(t)$ at Step 3 in the Forwarding algorithm and its level be i', where $h - \ell \geq i' \geq 1$. Then $d(s,t) > (1-\phi)\varepsilon\rho^{i'+\ell-1}$.*

Proof. Prove by contradiction. Assume that $d(s,t) \leq (1-\phi)\varepsilon\rho^{i'+\ell-1}$. By lemma 5.1, the cluster $C'(t) \ni t$ that has the longest valid prefix matching with t with the $\mathtt{ValidPath}_s(C'(t))$ bit set `true` in Route_s is at level at most $i'-1$, thus contradicting the assumption that $C'(t)$ is at level i'. ■

We next prove the following lemma regarding the level of $C'(t)$ given the level of $\hat{C}_k(s,t)$.

LEMMA 5.3. *Let a level-k cluster $\hat{C}_k(s,t) \in \Pi_k^{(j)}$ in tree T_j, where $h-1 \geq k \geq \ell$, be the lowest-level common cluster of s and t such that a shortest path from s to $C'(t) \in \Pi_{i'}^{(j)}$ is contained in $\mathbf{B}(s,\varepsilon\rho^k) \subseteq \hat{C}_k(s,t) \in \Pi_k^{(j)}$. Then $C'(t) \in \Pi_{i'}^{(j)}$ is at either level $k-\ell$ or level $k-\ell+1$.*

Proof. Be definition of $\hat{C}_k(s,t)$, we know that $l_0 \leq k \leq i'+\ell$ and $l_0 \geq i'+1$, where l_0 is the level of $\mathsf{lca}^j(s,C'(t))$. Thus $k-\ell \leq i' \leq k-1$. The lowest level that $C'(t)$ can be is at $k-\ell$, and we argue that $C'(t)$ can not be at a level higher than $k-\ell+1$.

Let e_0 be a closest entry point to $C'(t)$ for s, such that $e_0 \in A_{i'}(t) \cap C'(t)$ and the shortest path from s to e_0 is internal to $\mathbf{B}(s,\varepsilon\rho^k) \subseteq \hat{C}_k(s,t) \in \Pi_k^{(j)}$; hence $d(s,e_0) \leq \varepsilon\rho^k$. Since $C'(t)$ is at least one level below $\hat{C}_k(s,t)$ in T_j and $e_0, t \in C'(t)$, we have $d(e_0,t) \leq 2\eta_{k-1}$. Note that $C'(t) \subseteq \hat{C}_k(s,t)$ by Fact 5.3. Applying the triangle inequality, we have $d(s,t) \leq d(s,e_0) + d(e_0,t) \leq \varepsilon\rho^k + 2\eta_{k-1}$.

Now we examine the distance of $d(s,y')$ for all $y' \in A_{k-\ell+1}(t)$. Given that $d(t,y') \leq 2\eta_{k-\ell+1}$, we apply the triangle inequality and obtain:

$$d(s,y') \leq d(s,t) + d(t,y') \leq d(s,e_0) + d(e_0,t) + d(t,y') \leq \varepsilon\rho^k + 2\eta_{k-1} + 2\eta_{k-\ell+1} \leq \varepsilon\rho^{k+1},$$

where $\ell \geq 2$ and $\rho = \Theta(\frac{1}{\varepsilon})$.

Thus all shortest paths from s to y', for all $y' \in A_{k-\ell+1}(t)$, are contained in $\mathbf{B}(s,\varepsilon\rho^{k+1})$. The properties of the (ρ,ε)-PPHD ensure that there is at least one tree $T_{j'}$ such that $\mathbf{B}(s,\varepsilon\rho^{(k+1)}) \subseteq C_{k+1}(s) \in \Pi_{k+1}^{(j')}$. Let $C_{k-\ell+1}(t) \in \Pi_{k-\ell+1}^{(j')}$ be the level-$(k-\ell+1)$ cluster that contains t in $T_{j'}$. Given that $t \in \mathbf{B}(s,\varepsilon\rho^{k+1}) \subseteq C_{k+1}(s)$, we know that $C_{k-\ell+1}(t) \subseteq C_{k+1}(s) \in \Pi_{k+1}^{(j')}$ since $t \in \{C_{k-\ell+1}(t) \cap C_{k+1}(s)\} \neq \emptyset$ and $T_{j'}$ represents a laminar decomposition. Thus $C_{k-\ell+1}(t) \in \Pi_{k-\ell+1}^{(j')}$ must appear in s' routing table with $\mathtt{ValidPath}_s(C_{k-\ell+1}(t))$ set `true`, since $C_{k-\ell+1}(t) \subseteq C_{k+1}(s)$ is within ℓ levels below $C_{k+1}(s)$ in $T_{j'}$ and all shortest paths from s to $C_{k-\ell+1}(t)$ are contained in $\mathbf{B}(s,\varepsilon\rho^{k+1}) \subseteq C_{k+1}(s)$ in $T_{j'}$.

Thus the level i' of $C'(t)$ must satisfy $k-\ell \leq i' \leq k-\ell+1$ for $C'(t) \in \Pi_{i'}^{(j)}$ to be returned by $\mathsf{PrefMatch}_s(t)$. ■

FACT 5.6. *When $k = h$ and $\hat{C}_k(s,t) \in \Pi_k^{(j)}$ is $C_h(s,t) \in \Pi_h^{(j)}$, which is the entire network G, we know that $C'(t) \in \Pi_{i'}^{(j)}$ is at level $i' = h-\ell = k-\ell$.*

The next lemma shows the path characteristics from s to t up till entry point e_0 of $C'(t) \in \Pi_{i'}^{(j)}$.

LEMMA 5.4. *All messages to be forwarded or sent from node s to node t will follow the same shortest path up to the closest entry point e_0 of $C'(t) \in \Pi_{i'}^{(j)}$ to s. The shortest path from s to e_0 is internal to $\hat{C}_k(s,t) \in \Pi_k^{(j)}$ in T_j; it has a length of $h_{se_0}^i$ that satisfies:*

$$h_{se_0}^i = min_{e_z \in A_{i'}(t) \cap C'(t)} \{h_{se_z}^i\}, \tag{5.3}$$

where i' is the level of $C'(t) \in \Pi_{i'}^{(j)}$ and $k \leq i' + \ell$, and $\hat{C}_k(s,t) \in \Pi_k^{(j)}$, $A_{i'}(t)$, and $h_{se_z}^i$ are as defined above, and $h_{se_z}^i = \infty$ when the shortest path from s to e_z is not contained in $\hat{C}_k(s,t)$. At equilibrium, $h_{se_0}^i = \mathsf{HF}(s, C'(t)) = d(s,e_0)$. Finally, all vertices v on the shortest path from s to e_0 have a non-null $\mathsf{NextHop}_v(\mathsf{addr}(C'(t)))$ and share the same closest entry point e_0 to cluster $C'(t)$.

Proof. By the definition of $\hat{C}_k(s,t)$, for $k \leq h-1$, we know that $\hat{C}_k(s,t) \in \Pi_k^{(j)}$ is the level-k cluster, where $l_0 \leq k \leq i' + \ell$, in tree T_j, that is the lowest-level common cluster of s and t such that $\mathbf{B}(s, \varepsilon\rho^k) \subseteq \hat{C}_k(s,t) \in \Pi_k^{(j)}$ and $\mathbf{B}(s, \varepsilon\rho^k)$ contains a shortest path from s to $C'(t) \in \Pi_{i'}^{(j)}$ in T_j; specifically, $\mathbf{B}(s, \varepsilon\rho^k)$ contains a shortest paths from s to e_0, and $d(s,e_0) \leq \varepsilon\rho^k$. By Fact 5.3, we have $C'(t) \subseteq \hat{C}_k(s,t)$. When $k = h$, $\hat{C}_k(s,t) = C_h(s,t) \in \Pi_h^{(j)}$ is the root cluster X and naturally contains all shortest paths from s to $C'(t) \subseteq C_h(s,t)$, given that G is a connected graph.

All the nodes in $\hat{C}_k(s,t) \in \Pi_k^{(j)}$ contain one entry for $C'(t) \in \Pi_{i'}^{(j)}$ in their routing tables, since $k \leq i' + \ell$ and $C'(t) \subseteq \hat{C}_k(s,t)$ is a cluster within ℓ levels below $\hat{C}_k(s,t) \in \Pi_k^{(j)}$ in tree T_j. Propagation and subsequent updating of routing information among nodes of $\hat{C}_k(s,t) \in \Pi_k^{(j)}$ in T_j is equivalent to finding the minimum path internal to $\hat{C}_k(s,t)$ from any node $u \in \{\hat{C}_k(s,t) - C'(t)\}$ to an entry point of $C'(t)$ that is closest to node u; for s, the closest entry point to $C'(t)$ is e_0 such that $h_{se_0}^i = min_{e_z \in A_{i'}(t) \cap C'(t)} \{h_{se_z}^i\}$. Hence, at equilibrium, e_0 is on the minimal path from s to $C'(t)$ and $h_{se_0}^i = \mathsf{HF}(s, C'(t))$ represents the length of such minimal path.

All shortest paths of length $d(s,e_0)$ from s to e_0 are internal to $\hat{C}_k(s,t)$; when $k \leq h-1$, it is within $\mathbf{B}(s, \varepsilon\rho^k) \subseteq \hat{C}_k(s,t) \in \Pi_k^{(j)}$. Hence, at equilibrium, within $\mathbf{B}(s, \varepsilon\rho^k) \in \hat{C}_k(s,t) \in \Pi_k^{(j)}$ for $k \leq h-1$, or within $\hat{C}_k(s,t) = X$ for $k = h$, a shortest path of length $h_{se_0}^i = d(s,e_0)$ is formed between s and e_0 among nodes within a connected component, that share a common entry for $C'(t) \subseteq \hat{C}_k(s,t)$ in their routing tables. Thus we have $\mathsf{HF}(s, C'(t)) = h_{se_0}^i = d(s,e_0)$.

For any node v on one of these shortest paths from s to e_0, s and v must share the same closest entry point e_0 to $C'(t)$ at equilibrium, due to the execution of the distributed Bellman-Ford algorithm; furthermore, intermediate nodes v will be able to route the packet toward $C'(t) \in \Pi_{i'}^{(j)}$ in $\hat{C}_k(s,t)$ consistently since they each contain an entry for $C'(t) \in \Pi_{i'}^{(j)}$ with a non-null $\mathsf{NextHop}_v(\mathsf{addr}(C'(t)))$ field, given that these paths stay within $\hat{C}_k(s,t) \in \Pi_k^{(j)}$, where $k \leq i' + \ell$. The Forwarding algorithm will forward messages from node s destined to node t along the shortest path thus formed to first reach $C'(t)$ in tree T_j. ∎

The process of finding the next entry point repeats by the time the packet reaches e_0, an entry point to $C'(t) \in \Pi_{i'}^{(j)}$ in tree T_j, until the packet reaches its destination t. For example, e_0 selects a new tree T_l that contains the next cluster $C''(t) \in \Pi_{i''}^{(l)}$ with a longer prefix matching with t than $C'(t)$, and updates the packet header with $C''(t)$ accordingly. Note that $C''(t)$ and $C'(t)$ may belong to two different trees; hence while intermediate nodes between one entry point and another never switch trees, upon reaching an entry point, it is free to switch. The next lemma states the upper bound on the level i'' of $C''(t) \in \Pi_{i''}^{(l)}$, and the level of the common cluster $\hat{C}_k(x,t) \in \Pi_k^{(l)}$ that contains a shortest path from x to $C''(t)$ in $\mathbf{B}(x, \varepsilon\rho^k) \in \hat{C}_k(x,t) \in \Pi_k^{(l)}$. If x is t itself, we are done with forwarding.

LEMMA 5.5. *Let $1 \leq i' \leq h - \ell$ be the level of $C'(t) \in \Pi_{i'}^{(j)}$. Once the packet from s reaches an entry point x in $A_{i'}(t) \cap C'(t)$, including e_0, x will find a new level-(i'') cluster $C''(t) \in \Pi_{i''}^{(l)}$ at level $i'' \leq \max(0, i'-2)$ in some tree T_l, and the common cluster $\hat{C}_k(x,t) \in \Pi_k^{(l)}$ as defined above is at a level $k \leq i' - 2 + \ell$.*

Proof. We have $d(x,t) \le 2\eta_{i'} \le \phi\varepsilon\rho^{i'+\ell}$, since $x \in A_{i'}(t) \cap C'(t)$ is an entry point to some level-(i') cluster $C'(t) \in \Pi_{i'}^{(j)}$ containing t. We have $d(x,t) \le (1-\phi)\varepsilon\rho^{i'-2+\ell}$ so long as $\rho^2 \le \frac{1-\phi}{\phi}$, which can be satisfied when suitable constants are chosen for $\ell = \Theta(\log_\rho 1/\varepsilon\tau)$ and $\rho = \Theta(\frac{1}{\varepsilon})$. Lemma 5.1 tells us that $k' \le i'-2+\ell$ and $C''(t) \in \Pi_{i''}^{(l)})$ is at level $\le \max(0, i'-2)$. ∎

We are now ready for the main theorem that summarizes the path properties.

THEOREM 5.1. *Follow the Forwarding algorithm in Section 4.3, for all $k \le h$, the path from s to t as derived from the routing information at node s satisfies the recursive equation below, $h^c_{st} = h^i_{se_0} + h^c_{e_0t}$, where the shortest path $h^i_{se_0}$ from s to e_0 is contained in $\hat{C}_k(s,t)$ and its properties are as specified in Lemma 5.4. Secondly, the lookup path has a stretch of at most $(1+\tau)$. Finally, the algorithm switches trees for at most $\max(0, k-\ell+1)$ times. When $d(s,t) \le (1-\phi)\varepsilon\rho^n$, where $n \le h$, we have $k \le n$; otherwise, $k \le h$.*

Proof. The proof of the theorem is by induction on k, which is the level of the lowest common cluster $\hat{C}_k(s,t)$ of s and $C'(t)$ such that a shortest path from s to $C'(t)$ is contained in (a) $\mathbf{B}(s,\varepsilon\rho^k) \subseteq \hat{C}_k(s,t)$ for $k \le h-1$, or in (b) $\hat{C}_k(s,t) = C_h(s,t)$ for $k = h$. Recall i' is the level of $C'(t) \in \Pi_{i'}^{(j)}$, and $e_0 \in A_{i'}(t) \cap C'(t)$ is the closest entry point to $C'(t)$ for node s within $\hat{C}_k(s,t) \in \Pi_k^{(j)}$.

Base Case: $k \le \ell - 1$.

We first prove the following claim.

CLAIM 5.7. *If $\hat{C}_k(s,t)$ is at level $k \le \ell-1$, then $C'(t) = C_0(t)$ and $d(s,t) \le \varepsilon\rho^{\ell-1}$.*

Proof. By the definition of $\hat{C}_k(s,t)$, we know $\mathbf{B}(s,\varepsilon\rho^k) \subseteq \hat{C}_k(s,t) \in \Pi_k^{(j)}$ in T_j and $d(s,e_0) \le \varepsilon\rho^k$, where $e_0 \in C'(t)$ is the closest entry point to $C'(t)$ for node s. Thus $d(s,e_0) \le \varepsilon\rho^{\ell-1}$ for $k \le \ell-1$. Since $C'(t) \in \Pi_{i'}^{(j)}$ is a descendant of $\hat{C}_k(s,t) \in \Pi_k^{(j)}$ in T_j, it must be at a level lower than k; hence $d(e_0,t) \le 2\eta_{\ell-2} \le \frac{2\rho^{\ell-1}}{\rho-1} \le 4\rho^{\ell-2}$, since $e_0, t \in C'(t)$, and $C'(t)$ is at level $i' \le \ell-2$.

Applying the triangle inequality, we have $d(s,t) \le d(s,e_0) + d(e_0,t) \le \varepsilon\rho^{\ell-1} + 4\rho^{\ell-2} \le \varepsilon\rho^\ell$. Thus by Claim 5.4, we have $C'(t) = C_0(t)$, which is t itself; furthermore, $e_0 = t$ and $d(s,t) = d(s,e_0) \le \varepsilon\rho^{\ell-1}$. ∎

The above claim shows that $\hat{C}_k(s,t) \in \Pi_k^{(j)}$ in tree T_j contains a shortest path from s to $C'(t) = C_0(t) \in \Pi_0^{(j)}$, and t is the closest entry point to $C_0(t)$, which is t itself. Thus $h^c_{e_0t} = h^c_{tt} = 0$, since a node's distance to itself is zero. It remains to show that $h^c_{st} = h^i_{se_0} = h^i_{st}$; recall h^i_{st} refers to the shortest path from s to t as included in $\hat{C}_k(s,t)$. This is true since the routing table of every node v in $\hat{C}_k(s,t) \in \Pi_k^{(j)}$ for $k \le \ell-1$ contains an entry for $C_0(t) = t$, and a shortest path from s to t is contained in $\mathbf{B}(s,\varepsilon\rho^k) \subseteq \hat{C}_k(s,t)$ in tree T_j; hence at equilibrium, the clustered path between s and t as derived from Route_s is the shortest path from s to t, and it is internal to $\hat{C}_k(s,t)$, i.e., $h^c_{st} = \mathsf{HF}(s, C_0(t)) = h^i_{st} = d(s,t)$, where $k \le \ell-1$. The stretch is exactly 1 since $\frac{h^c_{st}}{d(s,t)} = \frac{h^i_{st}}{d(s,t)} = 1$. The forwarding algorithm does not switch tree at all.

Case $k = \ell$. By Lemma 5.3, $C'(t)$ is at level 0 or 1. When $C'(t)$ is at level $k-\ell = 0$, the proof is the same as that in the base case.

When $C'(t)$ is at level $k-\ell+1 = 1$, we have $d(s,t) > \varepsilon\rho^\ell$ by Claim 5.5. All messages to be forwarded or sent from node s to node t will first follow the same shortest path of length $h^i_{se_0} = d(s,e_0)$, that is internal to $\hat{C}_\ell(s,t)$, up to the closest entry point e_0 of $C'(t)$, as specified in Lemma 5.4.

Upon reaching e_0, Lemma 5.5 can be applied to show that $h^c_{e_0t}$, the clustered path from e_0 to t, is entirely contained in a level-(k') cluster $\hat{C}_{k'}(e_0,t)$ in some tree $T_{j'}$, where $k' \le i'-2+\ell = \ell-1$; thus as proved in the base case, $h^c_{e_0t} = h^i_{e_0t} = d(e_0,t)$. The clustered path from s to t as derived from Route_s indeed satisfies $h^c_{st} = h^i_{se_0} + h^c_{e_0t}$, where $h^i_{se_0} = d(s,e_0)$ and $h^c_{e_0t} = d(e_0,t)$.

Hence we obtain the bound on the entire path: $h^c_{st} = d(s,e_0) + d(e_0,t) \leq d(s,t) + 2d(e_0,t)$, where $d(s,e_0) \leq d(s,t) + d(e_0,t)$ by triangle inequality. And the path stretch is: $\frac{h^c_{st}}{d(s,t)} = 1 + \frac{2d(e_0,t)}{d(s,t)} \leq 1 + \frac{2(\rho+1)}{\varepsilon\rho^\ell} \leq 1+\tau$, where $\ell \geq 1 + (\log_\rho 4/\varepsilon\tau)$. The algorithm switches trees at most once.

Case $k \geq \ell + 1$**.** First we assume that the theorem is true up to $k-1$, let us show that it is true for k.

Let $\hat{C}_k(s,t) \in \Pi^{(j)}_k$ be the k^{th} level cluster that contains a shortest path from s to $C'(t) \in \Pi^{(j)}_{i'}$ in T_j. According to Lemma 5.4, all messages to be forwarded or sent from node s to node t will first follow the same shortest path of length $h^i_{se_0}$, that is internal to $\hat{C}_k(s,t)$, up to the closest entry point e_0 of $C'(t)$. By Lemma 5.3, we know that $C'(t)$ is at level $k-\ell$ or $k-\ell+1$ when $k \leq h-1$. When $k = h$, $C'(t)$ is at level $h - \ell = k - \ell$.

Upon reaching e_0, Step 3 of the forwarding algorithm is applied to find $C''(t)$, that has the longest valid matching with t in e_0's routing table. Since $C'(t)$ is at level $k-\ell$ or $k-\ell+1$, Lemma 5.5 shows the lowest common cluster $\hat{C}_{k'}(e_0,t)$ of e_0 and t, such that $\mathbf{B}(e_0,\varepsilon\rho^{k'}) \subseteq \hat{C}_{k'}(e_0,t)$ and $\mathbf{B}(e_0,\varepsilon\rho^{k'})$ contains a shortest path from e_0 to $C''(t)$, is at level $k' \leq i' - 2 + \ell \leq k-1$. Thus $h^c_{e_0t}$ is known from the induction hypothesis and $h^c_{st} = h^i_{se_0} + h^c_{e_0t}$.

Now we proceed to prove the bound on the stretch for level k. Let $C'(t)$ be at level $\beta - \ell$, where β is k or $k+1$; hence $d(e_0,t) \leq 2\eta_{\beta-\ell}$ given that $e_0, t \in C'(t)$. By Lemma 5.2, $d(s,t) > (1-\phi)\varepsilon\rho^{\beta-1}$, where $i' = \beta - \ell \geq 1$ for all $k \geq \ell+1$.

By Lemma 5.4, $h^i_{se_0}$ is the shortest path from s to e_0 that is internal to $\hat{C}_k(s,t)$ and $h^i_{se_0} = d(s,e_0)$; applying the triangle inequality, we obtain:

$$h^i_{se_0} = d(s,e_0) \leq d(s,t) + d(e_0,t) \leq d(s,t) + 2\eta_{\beta-\ell}. \tag{5.4}$$

By the induction hypothesis, we know

$$h^c_{e_0t} \leq (1+\tau)d(e_0,t) \leq 2(1+\tau)\eta_{\beta-\ell}. \tag{5.5}$$

Finally, we get the bound on the total path length from s to t:

$$h^c_{st} = h^i_{se_0} + h^c_{e_0t} \leq d(s,t) + 2(2+\tau)\eta_{\beta-\ell} \tag{5.6}$$

Now using the fact that $d(s,t) \geq (1-\phi)\varepsilon\rho^{\beta-1}$, and the fact that $\tau \leq 1$, we obtain the path stretch from s to t:

$$\frac{h^c_{st}}{d(s,t)} = 1 + \frac{2(2+\tau)\eta_{\beta-\ell}}{d(s,t)} \leq 1 + \frac{6\eta_{\beta-\ell}}{(1-\phi)\varepsilon\rho^{\beta-1}} \leq 1+\tau, \tag{5.7}$$

where $\ell \geq (\log_\rho 8/\varepsilon\tau) + 2$.

Finally, the algorithm switches trees for at most $k-\ell$ times to finally route within a level ℓ cluster, after which it switches tree at most once, thus adding up to a total number of $k-\ell+1$ times.

Now we look at the bound on k itself. When $d(s,t) \leq (1-\phi)\varepsilon\rho^n$, for all $n \leq h$, we have $k \leq n$ by Lemma 5.1. We now verify that all statements in the theorem still apply, for the clustered path h^c_{st}, when $d(s,t) > (1-\phi)\varepsilon\rho^{(h)}$ and $\hat{C}_k(s,t)$ is $C_h(s,t)$. First of all, when $\hat{C}_k(s,t) = C_h(s,t)$, following the Forwarding algorithm in Section 4.3, we know that $C'(t)$ is at level $h-\ell$ and hence $d(s,t) > (1-\phi)\varepsilon\rho^{(h-1)}$ by Lemma 5.2. The shortest path from s to $e_0 \in C'(t)$, $h^i_{se_0}$, is internal to $C_h(s,t)$, the entire network G. Upon reaching e_0, $h^c_{e_0t}$ is known by applying the theorem directly since $d(e_0,t) \leq 2\eta_{h-\ell} \leq (1-\phi)\varepsilon\rho^{h-1}$. Thus we have $h^c_{e_0t} \leq (1+\tau)d(e_0,t) \leq (1+\tau)2\eta_{h-\ell}$. Hence, the entire path satisfies the equation, $h^c_{st} = h^i_{se_0} + h^c_{e_0t}$.

Second, with the same calculation as the proof above, it is easy to verify that the entire path h^c_{st} has a stretch of at most $(1+\tau)$ given that $d(s,t) \geq (1-\phi)\varepsilon\rho^{(h-1)}$ and $h^c_{e_0 t} \leq (1+\tau)d(e_0,t) \leq (1+\tau)2\eta_{h-\ell}$. The algorithm switches trees for at most $(h-\ell+1)$ times. ■

COROLLARY 5.1. *For all t such that* $d(s,t) \leq \varepsilon\rho^{\ell}$, *path stretch is* 1.

Proof. Node s has a routing table entry for all t such that $d(s,t) \leq \varepsilon\rho^{\ell}$, since $\mathbf{B}(s,d(s,t)) \subseteq \mathbf{B}(s,\varepsilon\rho^{\ell})$ is fully contained in some level-ℓ cluster $C_\ell(s) \in \Pi^{(j)}_\ell$ in some tree T_j, and $C'(t)$ is $C_0(t) \in \Pi^{(j)}_0$; the base case of the above proof shows that path stretch is 1. ■

References

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

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

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

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

[5] P. B. Callahan and S. Rao Kosaraju. A decomposition of multidimensional point sets with applications to *k*-nearest-neighbors and *n*-body potential fields. *JACM*, 42(1):67–90, 1995.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[21] 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.

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

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