Optimal Euclidean Spanners: Really Short, Thin and Lanky

[Extended Abstract]

Michael Elkin *
Ben-Gurion University of the Negev
Beer-Sheva 84105, Israel
P.O. Box 653
elkinm@cs.bgu.ac.il

Shay Solomon †
Weizmann Institute of Science
Rehovot 76100, Israel
P.O. Box 26
shay.solomon@weizmann.ac.il

ABSTRACT

The degree, the (hop-)diameter, and the weight are the most basic and well-studied parameters of geometric spanners. In a seminal STOC'95 paper, titled "Euclidean spanners: short, thin and lanky", Arya et al. [2] devised a construction of Euclidean $(1 + \epsilon)$-spanners that achieves constant degree, diameter $O(\log n)$, weight $O(\log^2 n) \cdot \omega(MST)$, and has running time $O(n \cdot \log n)$. This construction applies to n-point constant-dimensional Euclidean spaces. Moreover, Arya et al. conjectured that the weight bound can be improved by a logarithmic factor, without increasing the degree and the diameter of the spanner, and within the same running time.

This conjecture of Arya et al. became one of the most central open problems in the area of Euclidean spanners. Nevertheless, the only progress since 1995 towards its resolution was achieved in the lower bounds front: Any spanner with diameter $O(\log n)$ must incur weight $\Omega(\log n) \cdot \omega(MST)$, and this lower bound holds regardless of the stretch or the degree of the spanner [12, 1].

In this paper we resolve the long-standing conjecture of Arya et al. in the affirmative. We present a spanner construction with the same stretch, degree, diameter, and running time, as in Arya et al.'s result, but with *optimal weight* $O(\log n) \cdot \omega(MST)$. So our spanners are as thin and lanky as those of Arya et al., but they are *really* short!

Moreover, our result is more general in three ways. First, we demonstrate that the conjecture holds true not only in constant-dimensional Euclidean spaces, but also in *doubling metrics*. Second, we provide a general tradeoff between the three involved parameters, which is *tight in the entire range*. Third, we devise a transformation that decreases the lightness of spanners in *general metrics*, while keeping all their other parameters in check. Our main result is obtained as a corollary of this transformation.

*This work is supported by BSF grant No. 2008430 and ISF grant No. 87209011.

†This work is supported by the Koshland Center for basic Research. Part of this work was done while the author was a graduate student at the Ben-Gurion University of the Negev, under the support of Clore Fellowship grant No. 81265410, BSF grant No. 2008430, and ISF grant No. 87209011.

STOC'13, June 1Ű4, 2013, Palo Alto, California, USA.

Categories and Subject Descriptors

G.2.2 [**Discrete Mathematics**]: Graph Theory—*Graph algorithms*

General Terms

Algorithms, Theory

Keywords

Doubling metrics, Euclidean spaces, Euclidean spanners

1. INTRODUCTION

1.1 Euclidean Metrics. Consider a set P of n points in $\mathbb{R}^d$, $d \geq 2$, and a real number $t \geq 1$. A graph $G = (P, E, \omega)$ in which the weight $\omega(p, q)$ of each edge $e = (p, q) \in E$ is equal to the Euclidean distance $\|p - q\|$ between p and q is called a *Euclidean graph*. We say that the Euclidean graph G is a *t-spanner* for P if for every pair $p, q \in P$ of distinct points, there exists a path $\Pi(p, q)$ in G between p and q whose weight (i.e., the sum of all edge weights in it) is at most $t \cdot \|p - q\|$. The parameter t is called the *stretch* of the spanner. The path $\Pi(p, q)$ is said to be a *t-spanner path* between p and q. In this paper we focus on the regime $t = 1 + \epsilon$, for $\epsilon > 0$ being an arbitrarily small constant. We will also concentrate on spanners with $|E| = O(n)$ edges.

Euclidean spanners, introduced in [10], turned out to be a fundamental geometric construct, with numerous applications. In particular, they are useful for geometric approximation algorithms, geometric distance oracles, and network design. Various properties of Euclidean spanners are a subject of intensive ongoing research. See the book [21], which is devoted to geometric spanners, and the references therein.

In addition to stretch ($t = 1 + \epsilon$) and sparsity ($|E| = O(n)$), other fundamental properties of Euclidean spanners include their *(maximum) degree*, *(hop-)diameter*, and *lightness*. The *degree* $\Delta(G)$ of a spanner G is the maximum degree of a vertex in G. The *diameter* $\Lambda(G)$ of a $(1 + \epsilon)$-spanner G is the smallest number Λ such that for every pair of points $p, q \in P$ there exists a $(1+\epsilon)$-spanner path between p and q in G that consists of at most Λ edges. The *lightness* $\Psi(G)$ of a spanner G is defined as the ratio between the *weight* $\omega(G) = \sum_{e \in E} \omega(e)$ of G and the weight $\omega(MST(P))$ of the minimum spanning tree $MST(P)$ for the point set P.

In this section we may write "spanner" as a shortcut for a "$(1+\epsilon)$-spanner with $O(n)$ edges". Feder and Nisan devised a construction of spanners with constant degree (see also [23]). In FOCS'94, [3] devised a spanner construction with logarithmic diameter. The diameter was improved to $O(\alpha(n))$, where $\alpha(n)$ is the inverse-Ackermann function, by Arya et al. [2] in STOC'95; the tradeoff between the diameter and number of edges was further explored [5, 21, 25].

In the nineties researchers started to investigate spanners that *combine* several parameters (among degree, diameter and lightness). [11] and [18] devised constructions of spanners with constant degree and lightness, within $O(n \cdot \log^2 n)$ and $O(n \cdot \log n)$ time, respectively. Arya et al. [2] devised a construction of spanners with logarithmic diameter and logarithmic lightness. (This combination was shown to be optimal by [12] in FOCS'08; see also [1].) This construction of [2] may have, however, an arbitrarily large degree. On the other hand, [2] devised also a construction of spanners with constant degree, logarithmic diameter and lightness $O(\log^2 n)$. In the end of their seminal work Arya et al. [2] conjectured that one can obtain a spanner with constant degree, logarithmic diameter and logarithmic lightness within time $O(n \cdot \log n)$. Specifically, they wrote:

Conjecture 1 ([2]) *For any $t > 1$, and any dimension d, there is a t-spanner constructible in $O(n \log n)$ time, with bounded degree, $O(\log n)$ diameter and weight $O(\omega(MST) \log n)$.*

In this paper we prove the conjecture of Arya et al. [2], and devise a construction of $(1+\epsilon)$-spanners with bounded degree, and logarithmic diameter and lightness. The running time of our construction is $O(n \cdot \log n)$, which is optimal in the algebraic computation-tree model [9]. (Regardless of the running time, prior to our work it was unknown whether spanners with constant degree, and logarithmic diameter and lightness exist, even for 2-dimensional point sets.)

In fact, our result is far more general than this.

THEOREM 1.1. *For any set of n points in Euclidean space of any constant dimension d, any $\epsilon > 0$ and any parameter $\rho \geq 2$, one can build in $O(n \cdot \log n)$ time a $(1+\epsilon)$-spanner with $O(n)$ edges, degree $O(\rho)$, diameter $O(\log_\rho n + \alpha(\rho))$ and lightness $O(\rho \cdot \log_\rho n)$.*

Due to lower bounds by [5, 12], this tradeoff is *optimal in the entire range* of the parameter ρ.

1.2 Doubling Metrics. Our result extends in another direction. Specifically, it applies to any *doubling metric* [19].[1]

Spanners for doubling metrics were intensively studied [6, 20, 22, 16, 17, 24]. They were found useful for Approximation Algorithms [4], and for Machine Learning [15]. In SODA'05 [6] showed that for any doubling metric there exists a spanner with constant degree. In SODA'06 [5] devised a construction of spanners with diameter $O(\alpha(n))$. [24] showed that in doubling metrics a greedy construction produces spanners with logarithmic lightness. [15] devised a construction of spanners for doubling metrics with constant degree and logarithmic diameter, within $O(n \cdot \log n)$ time. To our knowledge, prior to our work, there were no known construction of spanners with $O(n)$ edges for doubling metrics that provides logarithmic diameter and lightness simultaneously (even allowing arbitrarily large degree).[2]

We show that our construction extends to doubling metrics without incurring any overhead (beyond constants) in the degree, diameter, lightness, and running time. In other words, Theorem 1.1 applies to doubling metrics.

1.3 Our and Previous Techniques. Our starting point is the paper of Chandra et al. [8] from SoCG'92. This paper devises a general transformation: given a construction of spanners with certain stretch and number of edges their transformation returns a construction with roughly the same stretch and number of edges, but with a logarithmic lightness. The drawback of their transformation is that it blows up the degree and the diameter of the original spanner.

In this paper we devise a much more refined transformation. Our transformation enjoys all the useful properties of the transformation of [8], but, in addition, it preserves (up to constants) the degree and the diameter of the original construction. We then compose our refined transformation on top of known constructions of spanners with constant degree and logarithmic diameter (due to [2] in the Euclidean case, and due to [15] in the case of doubling metrics). As a result we obtain a construction of spanners with constant degree, logarithmic diameter and logarithmic lightness.

Our transformation can be applied not only for Euclidean or doubling metrics, but rather in general metrics. In fact, in [14] we have already obtained some improved results for spanners in general graphs that are based on a variant of this transformation. We also obtained several other new results as additional corollaries of our transformation, including optimal constructions of spanners for metrics induced by graphs of bounded tree-width or bounded tree-length, and constructions of fault-tolerant spanners for doubling metrics that provide a general tradeoff between the degree, diameter and lightness. These new results are part of ongoing research, and are outside the scope of the current paper.

Next, we provide a schematic overview of the two transformations (the one due to [8], and our refined one). The transformation of [8] starts with constructing an MST T of the input metric. Then it constructs the preorder traversal path $\mathcal{L}$ of T. The path $\mathcal{L}$ is then partitioned into $c \cdot n$ intervals of length $\frac{|\mathcal{L}|}{c \cdot n}$ each, for a constant $c > 1$. This is the bottom-most level $\mathcal{F}_1$ of the hierarchy $\mathcal{F}$ of intervals that the transformation constructs. Pairs of consecutive intervals are grouped together; this gives rise to $c \cdot n/2$ intervals of length $2 \cdot \frac{|\mathcal{L}|}{c \cdot n}$ each. The hierarchy $\mathcal{F}$ consists of $\ell = \log n$ levels, with c intervals of length $\frac{|\mathcal{L}|}{c}$ each in the last level $\mathcal{F}_\ell$.

In each level $j \in [\ell]$ of the hierarchy each non-empty interval is represented by a *representative* point of the original metric. Let Q_j denote the set of j-level representatives. The transformation invokes its input black-box construction of spanners on each point set Q_j separately. Each of those ℓ auxiliary spanners is then pruned ("long" edges are removed from it). The remaining edges in all the auxiliary spanners, together with the MST T, form the output spanner.

Intuitively, the pruning step ensures that the resulting

[1] The *doubling dimension* of a metric is the smallest value d such that every ball B in the metric can be covered by at most 2^d balls of half the radius of B. This generalizes the Euclidean dimension, because the doubling dimension of Euclidean space $\mathbb{R}^d$ is proportional to d. A metric is called *doubling* if its doubling dimension is constant.

[2] On the other hand, as was mentioned in Section 1.1, for Euclidean metrics such a construction was devised in [2]. However, the degree in the latter construction is unbounded.

spanner is reasonably light. The stretch remains roughly intact, because each distance is taken care "on its own scale". The number of edges does not grow by much, because the sequence $|Q_1|, |Q_2|, \ldots, |Q_\ell|$ decays geometrically. However, the diameter is blown up, because within each interval the MST-paths (which may contain many edges) are used to reach points that do not serve as representatives. Also, the degree is blown up because the same point may serve as a representative in many different levels.

To fix the problem with the diameter we use a construction of 1-dimensional spanners to shortcut the traversal path $\mathcal{L}$. We remark that $(1+\epsilon)$-spanners with $O(n)$ edges, constant degree, logarithmic diameter and logarithmic lightness for sets of n points on a line (1-dimensional case) were devised already in 1995 by [2]. Plugging[3] this 1-dimensional spanner construction into the transformation of [8] gives rise to an improved transformation that keeps the diameter in check, but still blows up the degree.

To fix the problem with the degree, it is natural to try distributing the degree load evenly between "nearby" points along $\mathcal{L}$. Alas, if one sticks with the original hierarchy $\mathcal{F}$ of partitions of $\mathcal{L}$ into intervals, this turns out to be impossible. The problem is that the same point may be the only eligible representative for many levels of the hierarchy. Overcoming this hurdle is the heart of our paper. Instead of intervals we divide the point set into a different hierarchy $\hat{\mathcal{F}}$ of sets, which we call *bags*. On the lowest level of the hierarchy the bags and the intervals coincide. As our algorithm proceeds it carefully moves points between bags so as to guarantee that no point will ever be overloaded. At the same time we never put points that are far away from one another in the original metric into the same bag. Indeed, if remote points end up in the same bag, then the auxiliary spanners for the sets of representatives, as well as the 1-dimensional spanner for $\mathcal{L}$, cease providing short $(1+\epsilon)$-spanner paths for the original point set. On the other hand, degree constraints may force our algorithm to relocate points arbitrarily far away from their initial position on $\mathcal{L}$. Our construction balances carefully between these two contradictory requirements.

1.4 Related Work. In ESA'10 [27] the authors of the current paper devised a construction of spanners that trades gracefully between the degree, diameter and lightness. That construction, however, could only match the previous sub-optimal bounds of Arya et al. [2], but not improve them. In particular, the lightness of the construction of [27] is $\Omega(\log^2 n)$, regardless of the other parameters.

1.5 Consequent Work. A preliminary version of this paper started to circulate in April 2012 [13]. It sparked a number of follow-up papers. First, in [14] we used the technique developed in this paper to devise an efficient construction of light spanners for general graphs. Second, in [7] Chan et al. came up with an alternative construction of spanners for doubling metrics with constant degree, and logarithmic diameter and lightness. Their construction and analysis are arguably simpler than ours. In addition, they extended this result to the fault-tolerant setting. A yet alternative construction of fault-tolerant spanners with the same properties and with running time $O(n \cdot \log n)$ was devised recently by Solomon [26]. However, while our construction provides an optimal tradeoff between the diameter and lightness ($O(\log_\rho n + \alpha(\rho))$ versus $O(\rho \cdot \log_\rho n)$ for the entire range of the parameter $\rho \geq 2$), the constructions of [7, 26] do the job only for $\rho = O(1)$. As far as we know they cannot be extended to provide the general tradeoff. Finally, the constructions of [7, 26] do not provide a transformation for converting spanners into light spanners in general metrics.

We also stress that both constructions [7, 26] are consequent to our work. These constructions build upon ideas and techniques that we present in the current paper.

1.6 Structure of the Paper. In Section 2 we describe our construction (Algorithm *LightSp*). A detailed outline of Section 2 appears in the paragraph preceding Section 2.1. We analyze the properties of the spanners produced by our algorithm in Section 3. The most technically involved parts of the analysis concern the stretch and diameter (Section 3.1) and the degree (Section 3.2) of the produced spanners. The full version of this paper appears in the archive [13].

1.7 Preliminaries. The next theorem provides optimal spanners for *1-dimensional* Euclidean metrics with respect to all three parameters (degree, diameter and lightness).

THEOREM 1.2 ([2, 27]). *For any n-point 1-dimensional space M and any $\rho \geq 2$, one can build in $O(n)$ time a 1-spanner H with $|H| = O(n)$, $\Delta(H) = O(\rho)$, $\Lambda(H) = O(\log_\rho n + \alpha(\rho))$ and $\Psi(H) = O(\rho \cdot \log_\rho n)$.*

The next theorem provides spanners for doubling metrics with an optimal tradeoff between the degree and diameter. Note, however, that this tradeoff does not involve lightness.

THEOREM 1.3 ([2, 17, 27]). *For any n-point doubling metric $M = (P, \delta)$, any $\epsilon > 0$ and $\rho \geq 2$, one can build in $O(n \cdot \log n)$ time a $(1+\epsilon)$-spanner H with $|H| = O(n)$, $\Delta(H) = O(\rho)$ and $\Lambda(H) = O(\log_\rho n + \alpha(\rho))$.*

Our main transformation theorem is formulated below.

THEOREM 1.4. *Let $M = (P, \delta)$ be an arbitrary metric. Let $\rho \geq 2, t \geq 1$ be arbitrary parameters. Suppose that there is an algorithm (henceforth, Algorithm BasicSp) which builds, for any subset $Q \subseteq P$, $|Q| = n$, within $SpTm(n)$ time, a t-spanner H for the sub-metric $M[Q]$ of M induced by Q, so that $|H| \leq SpSz(n)$, $\Delta(H) \leq \Delta(n)$, $\Lambda(H) \leq \Lambda(n)$. Suppose also that all the functions $SpSz(n), \Delta(n), \Lambda(n)$ and $SpTm(n)$ are monotone non-decreasing, while the functions $SpSz(n)$ and $SpTm(n)$ are also convex and vanish at zero.*

Then there is an algorithm (Algorithm LightSp) which builds for every subset $Q \subseteq P$, $|Q| = n$, and any $\epsilon > 0$, within $O(SpTm(n) \cdot \log_\rho(t/\epsilon) + n \cdot \log n)$ time, a $(t+\epsilon)$-spanner H' for $M[Q]$ with $|H'| = O(SpSz(n) \cdot \log_\rho(t/\epsilon))$, $\Delta(H') = O(\Delta(n) \cdot \log_\rho(t/\epsilon) + \rho)$, $\Lambda(H') = O(\Lambda(n) + \log_\rho n + \alpha(\rho))$, $\Psi(H') = O(\frac{SpSz(n)}{n} \cdot \rho \cdot \log_\rho n \cdot (t^3/\epsilon))$.

Given this theorem we derive our main result by instantiating the algorithm from Theorem 1.3 as Algorithm *BasicSp* in Theorem 1.4. As a result we obtain a construction of $(1+\epsilon)$-spanners H for doubling metrics with $|H| = O(n)$, $\Delta(H) = O(\rho)$, $\Lambda(H) = O(\log_\rho n + \alpha(\rho))$, $\Psi(H) = O(\rho \cdot \log_\rho n)$, in time $O(n \cdot \log n)$.

We denote $[i, j] = \{i, i+1, \ldots, j\}, [i] = [1, i]$. For paths Π, Π' connecting vertices v and u and u and w, respectively, denote by $\Pi \circ \Pi'$ the concatenation of these paths.

[3] In fact, we use our own more recent construction [27] of 1-spanners for 1-dimensional spaces with the above properties. Having stretch 1 instead of $(1+\epsilon)$ simplifies the analysis.

2. ALGORITHM *LIGHTSP*

Let $M = (P, \delta)$ be any metric, and let $Q \subseteq P$ be any subset of n points from P. Algorithm *LightSp* starts with computing an MST, or an approximate MST, T, for the metric $M[Q]$. In low-dimensional Euclidean and doubling metrics an $O(1)$-approximate MST can be computed in $O(n \cdot \log n)$ time. In general, a t-approximate MST can be computed in $O(SpTm(n) + n \cdot \log n)$ time. (See [13] for details.)

Let $\mathcal{L}$ be the Hamiltonian path of $M[Q]$ obtained as the preorder traversal of T. Define $L = \omega(\mathcal{L})$; it is well known that $L \le 2 \cdot \omega(T)$, thus $L = O(t \cdot \omega(MST(M[Q])))$. Write $\mathcal{L} = (q_1, q_2, \ldots, q_n)$, and let $M_{\mathcal{L}} = (Q, \delta_{\mathcal{L}})$ be the 1-dimensional space induced by the path $\mathcal{L}$, where $\delta_{\mathcal{L}}$ is the distance in $\mathcal{L}$. We use Theorem 1.2 to build in $O(n)$ time a 1-spanner $H_{\mathcal{L}}$ for $M_{\mathcal{L}}$ with $|H_{\mathcal{L}}| = O(n)$, $\Delta(H_{\mathcal{L}}) = O(\rho)$, $\Lambda(H_{\mathcal{L}}) = O(\log_\rho n + \alpha(\rho))$ and $\Psi(H_{\mathcal{L}}) = O(\rho \cdot \log_\rho n)$. Let $H = (Q, E_H)$ be the graph obtained from $H_{\mathcal{L}}$ by assigning weight $\delta(p,q)$ to each edge $(p,q) \in H_{\mathcal{L}}$. Since edge weights in H are no greater than the respective edge weights in $H_{\mathcal{L}}$, we have (i) $\omega(H) \le \omega(H_{\mathcal{L}}) = O(\rho \cdot \log_\rho n) \cdot L$, and (ii) for any $p, q \in Q$, there is a path $\Pi_H(p,q)$ in H with weight at most $\delta_{\mathcal{L}}(p,q)$ and $O(\log_\rho n + \alpha(\rho))$ edges. We call H the *path-spanner*. We define an order relation $\prec_{\mathcal{L}}$ on Q, namely, we write $q_i \prec_{\mathcal{L}} q_j$ (respectively, $q_i \preceq_{\mathcal{L}} q_j$) iff $i < j$ (resp., $i \le j$).

Let $\ell = \lceil \log_\rho n \rceil$. Define $Q_0 = Q$, let $n_0 = |Q_0| = n$, and define the *0-level threshold* $\tau_0 = 2 \cdot \frac{L}{n} \cdot t \cdot (1 + \frac{1}{c})$, where $c = \lceil \frac{4 \cdot (t+1)}{\epsilon} \rceil = \Theta(t/\epsilon)$. For $j \in [\ell]$, we define $\xi_j = \rho^{j-1} \cdot \frac{L}{n}$. We divide the path $\mathcal{L}$ into $n_j = \frac{c \cdot L}{\xi_j} = \frac{c \cdot n}{\rho^{j-1}}$ intervals of length $\mu_j = \frac{\xi_j}{c}$ each.[4] Define also the *j-level threshold* $\tau_j = 2\mu_j \cdot \rho \cdot t \cdot (c+1) = 2 \cdot \frac{L}{n} \cdot t \cdot (1 + \frac{1}{c}) \cdot \rho^j$. These intervals induce a partition of the point set Q in the obvious way; denote these intervals and the corresponding point sets by $I_j^{(1)}, I_j^{(2)}, \ldots, I_j^{(n_j)}$ and $Q_j^{(1)}, Q_j^{(2)}, \ldots, Q_j^{(n_j)}$, respectively.

We define $\mathcal{I}_j = \{I_j^{(1)}, \ldots, I_j^{(n_j)}\}$, and $\mathcal{I} = \bigcup_{j=1}^{\ell} \mathcal{I}_j$. Note that, for each $j \in [2, \ell]$, every j-level interval I is a union of ρ consecutive $(j-1)$-level intervals. The interval I is called the *parent* of these $(j-1)$-level intervals, and they are called its *children*. This nested hierarchy of intervals defines in a natural way a forest $\mathcal{F}$ of ρ-ary trees, whose vertices (henceforth, *bags*) correspond to intervals from $\mathcal{I}$. With a slight abuse of notation we denote by $\mathcal{I}$ also the set of bags in $\mathcal{F}$, and by $\mathcal{F}_j = \mathcal{I}_j$ the set of j-level bags in $\mathcal{F}$, for each $j \in [\ell]$. Each of the trees in $\mathcal{F}$ is rooted at an ℓ-level interval. Thus, the number of trees in $\mathcal{F}$ is $|\mathcal{F}_\ell| = n_\ell = \frac{c \cdot n}{\rho^{\ell-1}}$, and so $c < n_\ell \le c \cdot \rho$. Denote the interval that corresponds to a bag v of $\mathcal{F}$ by $I(v)$, and denote the point set of $I(v)$ by $N(v)$. We call the point set $N(v)$ the *native point set* of v. For an inner bag v in $\mathcal{F}$ with ρ children $c_1(v), \ldots, c_\rho(v)$, we have $I(v) = \bigcup_{i=1}^{\rho} I(c_i(v))$, and $N(v) = \bigcup_{i=1}^{\rho} N(c_i(v))$. Note that $\bigcup_{v \in \mathcal{F}_j} I(v) = [q_1, q_n]$, and $\bigcup_{v \in \mathcal{F}_j} N(v) = Q$. Also, for any pair of distinct j-level bags $u, v \in \mathcal{F}_j$, $N(u) \cap N(v) = \emptyset$.

In Algorithm *LightSp* we (implicitly) maintain another forest $\hat{\mathcal{F}}$ over the same bag set $\mathcal{I}$. Specifically, a j-level bag v, for some $j \in [\ell - 1]$, may become a child of some $(j+1)$-level bag u, other than the parent $\pi(v)$ of v in $\mathcal{F}$. If this happens we say that u becomes a *step-parent* of v in $\mathcal{F}$ (and u is a parent of v in $\hat{\mathcal{F}}$), and v becomes a *step-child* of u in $\mathcal{F}$ (and v is a child of u in $\hat{\mathcal{F}}$). As a result the points associated with the bag v become associated with u. We will soon provide more details on this.

In $\mathcal{F}$ each bag v corresponds to an interval $I(v) \in \mathcal{I}$, and contains only points that lie within this interval (i.e., the points of $N(v)$). On the other hand, each bag $v \in \hat{\mathcal{F}}$ may contain points from many different intervals of $\mathcal{I}$. We denote by $\hat{\mathcal{F}}_j$ the set of j-level bags of $\hat{\mathcal{F}}$. If a bag $v \in \mathcal{F}_j$ becomes a child in $\hat{\mathcal{F}}$ of a bag u, then it will hold that $u \in \mathcal{F}_{j+1}$. This guarantees that $\mathcal{F}_j = \hat{\mathcal{F}}_j$, for every $j \in [\ell]$.

The rest of this section is organized as follows. In Section 2.1 we describe point sets which are associated with bags of $\hat{\mathcal{F}}$. During the execution of the algorithm some bags are labeled as *zombies* or *incubators*. These notions are discussed in Section 2.2. In Section 2.3 we provide an outline of our algorithm. The algorithm itself is described in Section 2.4.

2.1 Point Sets.

In addition to the native point set $N(v)$, the algorithm maintains for each bag v three more point sets: the *base point set* $B(v)$, the *kernel set* $K(v)$, and the *point set* $Q(v)$. These sets will satisfy $B(v) \subseteq K(v) \subseteq Q(v)$. It will also hold that $B(v) \subseteq N(v)$.

Algorithm *LightSp* processes the forest $\mathcal{F}$ bottom-up. We refer to the processing of bags of $\mathcal{F}_j$ as the *j-level processing*, for each $j \in [\ell]$. (It will be described in Section 2.4.) The algorithm maintains the point sets $B(v), K(v)$ and $Q(v)$ of all bags $v \in \mathcal{F}_j$ during the j-level processing as follows.

For a bag $v \in \mathcal{F}_1$, we set $B(v) = K(v) = Q(v) = N(v)$.

A bag v is called *empty* if $Q(v) = \emptyset$. A non-empty $(j-1)$-level bag z, $j \in [2, \ell]$, may become a step-child of some j-level bag v, other than the parent $\pi(z)$ of z in $\mathcal{F}$. If this happens, we say that *z is disintegrated from* $\pi(z)$, and also that *z joins* v. Denote by $\mathcal{J}(v)$ the set of all bags z that join v. They will be called the *(joining) step-children* of v. Denote also by $\mathcal{S}(v)$ the set of *surviving children* of v, i.e., the non-empty bags z with $v = \pi(z)$ that did not join some other j-level bag v', $v' \ne v$ ($v' \in \mathcal{F}_j$). Let $\chi(v) = \mathcal{S}(v) \cup \mathcal{J}(v)$ be the set of *extended children* of v. Observe that $\chi(v) \subseteq \mathcal{F}_{j-1}$, and that all bags in $\chi(v)$ are non-empty.

The *base point set* $B(v)$ (respectively, *point set* $Q(v)$) of a bag $v \in \mathcal{F}_j, j \in [2, \ell]$, is defined as the union of the base point sets (resp., point sets) of its surviving (resp., extended) children, i.e., $B(v) = \bigcup_{z \in \mathcal{S}(v)} B(z)$, $Q(v) = \bigcup_{z \in \chi(v)} Q(z)$.

Each bag z will be a step-child of at most one bag v. Also, for any bag v, each non-empty child u of v which is not surviving, will be necessarily a step-child of some other bag $v' \ne v$. (The bags v and v' are of the same level.) Hence, for each level $j \in [\ell]$, the set $\{Q(v) \mid v \in \mathcal{F}_j\}$ is a partition of Q. In particular, for distinct $u, v \in \mathcal{F}_j$, $Q(u) \cap Q(v) = \emptyset$.

The *kernel set* $K(v)$ of v is an intermediate set, in the sense that $B(v) \subseteq K(v) \subseteq Q(v)$. Representative points will always be chosen from $K(v)$ (see Section 2.3). The points of $K(v)$ will be used to alleviate the degree load from the points of $B(v)$, and thus it is important that $K(v)$ will contain enough points. (If $|B(v)| \ge \ell$, the kernel set $K(v)$ is useless. The kernel set is useful whenever $|B(v)| < \ell$.) Intuitively, all points of $K(v)$ will always be pretty close to the base point set $B(v)$, both in terms of the metric distance in M, and in terms of the hop-distance. This will guarantee that points of $K(v)$ provide good substitutes for points of $B(v)$.

The kernel set $K(v)$ of a bag $v \in \mathcal{F}_j, j \in [2, \ell]$, is defined as follows. The *surviving kernel set* $K'(v)$ is given by $K'(v) = \bigcup_{z \in \mathcal{S}(v)} K(z)$. If $|K'(v)| \ge \ell$, set $K(v) = K'(v)$. If $|K'(v)| < \ell$, set $K(v) = K'(v) \cup \bigcup_{z \in \mathcal{J}(v)} K(z) = \bigcup_{z \in \chi(v)} K(z)$.

[4] In this extended abstract we ignore all integrality issues. In particular, ρ and n_j, for $j \in [\ell]$, are assumed to be integers.

The intuition behind increasing the kernel set $K(v)$ beyond its surviving kernel set $K'(v)$ (i.e., setting $K(v) = K'(v) \cup \bigcup_{z \in \mathcal{J}(v)} K(z)$) whenever $|K'(v)| < \ell$ is that in this case the surviving kernel set is too small. Hence one needs to add to it more points to alleviate the degree load. In the complementary case ($|K'(v)| \geq \ell$), one can distribute the load of the $O(\ell)$ auxiliary spanners that Algorithm *LightSp* constructs (see Section 2.4) among the points of $K'(v)(= K(v))$ in such a way that no kernel point is overloaded.

A bag v is called *small* if $|Q(v)| < \ell$, and *large* otherwise. The next lemma follows from the definitions by induction.

LEMMA 2.1. *Fix any $j \in [\ell]$, and let v be a j-level bag. If v is small, then $K(v) = Q(v)$. If v is large, then $|K(v)| \geq \ell$.*

2.2 Zombies and Incubators, and Procedure *Attach*. In this section we describe several key notions (including zombie, incubator and attachment), and describe a related subroutine of our algorithm, called Procedure *Attach*.

As discussed above, a j-level bag z may join as a step-child of some $(j+1)$-level bag v, $v \neq \pi(z)$, as a result of an attachment. We now take a closer look at this process.

Each bag $v \in \mathcal{F}$ may hold a *label* of one of two types, a *zombie* or an *incubator*. Initially, all bags are unlabeled.

It may happen that an i-level bag v is *attached* to an i-level bag u. It must hold that $\pi(v) \neq \pi(u)$. We also say that v is *adopted* by $\pi(u) \in \mathcal{F}_{i+1}$. We denote the *attachment* of v to u by $\mathcal{A}(u, v)$. We also call it an *adoption* of v by $\pi(u)$. However, the attachment and adoption come with an *incubation period*. Specifically, there is a positive integer γ which determines the length of the incubation period. The $(i+\gamma-1)$-level ancestor v' of v will actually be *disintegrated* from its parent $\pi(v')$, and *join* the $(i+\gamma)$-level ancestor u' of u. The bag u' is called the *actual adopter*. It will become clear later that adoption rules (which we still did not finish to specify) imply that $\pi(v') \neq u'$. Attachments occur only for $i \leq \ell - \gamma$. The sets $B(u'), K(u'), Q(u')$ and $B(\pi(v')), K(\pi(v')), Q(\pi(v'))$ are computed according to the rules specified in Section 2.1.

The $\gamma - 1$ immediate ancestors of $v = v^{(0)}$, namely, the bags $v^{(1)} = \pi(v), v^{(2)} = \pi(v^{(1)}), \ldots, v^{(\gamma-1)} = \pi(v^{(\gamma-2)}) = v'$, change their status as a a result of this attachment. They will be now labeled as *zombies*. The bag v' is called a *disappearing zombie*, because it joins u' rather than its original parent $\pi(v')$. We will refer to v as an *attached bag*. Similarly, the $\gamma - 1$ immediate ancestors of $u = u^{(0)}$, namely, the bags $u^{(1)} = \pi(u), u^{(2)} = \pi(u^{(1)}), \ldots, u^{(\gamma-1)} = \pi(u^{(\gamma-2)})$, change their status as well. They will be labeled as *incubators*. The $(i+\gamma)$-level bag $u' = u^{(\gamma)}$ is *not* labeled as an incubator. This bag is called the *actual adopter*. We remark that the same bag may become an adopter (and incubator) of several different descendants. The i-level bag $u = u^{(0)}$ will be called the *initiator* of the attachment $\mathcal{A}(u, v)$. (See Figure 1.)

To determine which bags will participate in attachments, Algorithm *LightSp* uses a simple graph procedure, called Procedure *Attach*. It accepts as input an n-vertex graph $G = (V, E)$, whose vertices are labeled by either *safe* or *risky*. (The meaning of these labels will become clear in Section 2.4.) The procedure returns a *star forest*, i.e., a collection Γ of vertex disjoint stars, that satisfies the following two conditions: (1) $\bigcup_{S \in \Gamma} V(S)$ contains the set $R \subseteq V$ of vertices which are not isolated[5] in G, and labeled as risky.

[5]A vertex v in a (possibly directed) graph G is called *isolated* if there are no edges incident on v in G.

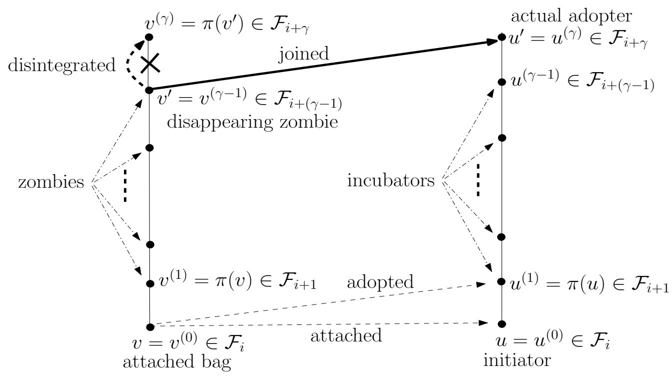

Figure 1: An illustration of an attachment $\mathcal{A}(u, v)$.

(2) Each star $S \in \Gamma$ contains a *center* $s \in V$ labeled as either safe or risky, and one or more leaves $z_1, \ldots, z_k \in V$ labeled as risky. The edge set $E(S)$ of a star S is given by $E(S) = \{(z_i, s) \mid i \in [k]\}$.

Intuitively, Procedure *Attach* attaches each risky vertex to another vertex. Each star of Γ will eventually be merged into a single super-vertex in a certain supergraph in our algorithm. This will be our way to "get rid" of risky vertices. It is instructive to view each star center s as an attachment initiator, and leaves of the star centered at s as zombies that will eventually join an appropriate ancestor of s as its step-children. The procedure itself (as well as $O(|V|)$ time implementation) is trivial, and is deferred to the full version.

2.3 Additional Ingredients, and Outline. Algorithm *LightSp* also computes a set of edges $\mathcal{B}$, called the *base edge set* of the spanner. For each bag $v \in \mathcal{F}$, the base edge set $\mathcal{B}$ will connect the base point set $B(v)$ of v via a simple path $P(v)$. If $p_1, \ldots, p_k$ are the points of $B(v)$ ordered with respect to the order relation $\prec_{\mathcal{L}}$, then $P(v) = (p_1, \ldots, p_k)$. It is easy to build the base edge set $\mathcal{B}$ in linear time, so as to guarantee that $\Delta(\mathcal{B}) \leq 2, \Psi(\mathcal{B}) = O(\ell)$. (See Section 2.2 in the full version [13] for the details.)

Algorithm *LightSp* starts with computing the base edge set $\mathcal{B}$ and the path-spanner $H = (Q, E_H)$ (see the beginning of Section 2). Next, it invokes Algorithm *BasicSp* to build a t-spanner $G'_0 = (Q_0, E'_0)$ for the sub-metric $M[Q_0]$ of M induced by $Q = Q_0$. Define $\tilde{E}_0$ to be the edge set obtained by *pruning* E'_0, i.e., removing all edges of weight greater than the *0-level threshold* τ_0. The corresponding graph $\tilde{G}_0 = (Q_0, \tilde{E}_0)$ is called the *0-level auxiliary spanner*. In a similar way (details will be provided in Section 2.4), the algorithm builds auxiliary j-level spanners $\tilde{G}_j = (Q_j, \tilde{E}_j)$, for each $j \in [\ell]$. The spanner $\tilde{G}_j$ is a graph over the set $Q_j = \{r(w) \mid w \in \mathcal{F}_j, Q(w) \neq \emptyset\}$ of *representatives* of the non-empty j-level bags. For each non-empty bag $v \in \mathcal{F}_j$, its *representative* $r(v)$ is the least loaded[6] point from the kernel set $K(v)$. (Ties are broken arbitrarily.) The *load* of a point p at the beginning of the j-level processing, denoted $load\_ctr_{j-1}(p)$, is the number of levels $i \in [j-1]$ in which the point p has positive degree in the i-level auxiliary spanner $\tilde{G}_i$. The union $\mathcal{B} \cup E_H \cup \bigcup_{j=0}^{\ell} \tilde{E}_j$ is the ultimate spanner $\tilde{G} = (Q, \tilde{E})$ that Algorithm *LightSp* returns.

[6]In fact, it turns out to be more convenient for analysis purposes to select the representatives using a more elaborate rule. (This issue will be discussed in Section 3.2.) Both variants of the algorithm work equally well.

2.4 ***j*-level processing.** The routine that performs j-level processing (henceforth, Procedure $Process_j$), for $j \in [\ell]$, accepts as input the forest $\hat{\mathcal{F}}$ that was processed by $Process_1, Process_2, \ldots, Process_{j-1}$. That is, for each j-level bag w, Procedure $Process_j$ accepts as input the sets $B(w), K(w), Q(w)$, and its representative $r(w) \in K(w) \subseteq Q$. It is also known to the procedure if this bag is a zombie, a disappearing zombie, an incubator, or an actual adopter.

Procedure $Process_j$ consists of three parts. *Part I* of Procedure $Process_j$ invokes Algorithm *BasicSp* for the metric $M[Q_j]$. The algorithm constructs a t-spanner $G'_j = (Q_j, E'_j)$. It then *prunes* G'_j, i.e., it removes from it all edges e with $\omega(e) > \tau_j$. Denote the pruned graph $G^*_j = (Q_j, E^*_j)$. The edge set E^*_j is inserted into the spanner $\tilde{G}$. This completes the description of Part I.

While the spanner G'_j is connected, some points of Q_j may be isolated in G^*_j. Denote by Q^*_j the subset of Q_j of all points $q \in Q_j$ which are not isolated in G^*_j.

If $j \leq \ell - \gamma$, then Procedure $Process_j$ enters *Part II*. (Otherwise, Part II is skipped.) We need to introduce some definitions. First, for a point $p \in Q$ and an index $j \in [\ell]$, denote by $v_j(p)$ the j-level *host bag* of p, i.e., the unique bag $v_j(p)$ that satisfies $p \in Q(v_j(p))$, $v_j(p) \in \mathcal{F}_j = \hat{\mathcal{F}}_j$.

A bag v is called *useless* if it is either empty or a zombie. Otherwise it is called *useful*. For a bag $v \in \mathcal{F}_i$, $i \leq \ell - \gamma$, its $(i + \gamma)$-level ancestor $v^{(\gamma)}$ is called the *cage-ancestor* of v. The set of all i-level descendants of $v^{(\gamma)}$, denoted $\mathcal{C}(v)$, is called the *cage* of v. If v is the only useful bag in its cage, it is called a *lonely* bag; otherwise it is called a *crowded* bag.

A non-empty bag v is called *safe* if it satisfies at least one of the following conditions: (1) v is large, (2) v is crowded, (3) v is an incubator or a zombie. Otherwise v is called *risky*. For v to be risky it must be small (i.e., $|Q(v)| < \ell$), lonely, and neither an incubator nor a zombie. A representative $r(v)$ of a safe (respectively, risky; useful; zombie) bag v is called *safe* (resp., *risky*; *useful*; *zombie*) as well.

Intuitively, for a *safe* bag v, there is no danger that one of the points $p \in Q(v)$ of v will become overloaded, i.e., that its degree in the spanner will be too large. Indeed, if v is a large bag, it contains many points which can share the load. If v is a crowded bag or an incubator or a zombie, it will soon merge with at least one other non-empty bag u, and through this merge it will acquire additional points that can participate in sharing the load. If v is a crowded bag, u is a bag that belongs to v's cage. Otherwise v is an incubator or a zombie. In this case u does not belong to v's cage. However, in either case, we will argue later that any point in $Q(v)$ is pretty close to any point in $Q(u)$ in the original metric M. These points will also stay close in the spanner.

Part II of Procedure $Process_j$ computes the set $\hat{Q}_j$, which is the subset of Q^*_j that contains only useful representatives. Then it invokes Algorithm *BasicSp*, this time with input $M[\hat{Q}_j]$. As a result, a graph $\breve{G}_j = (\hat{Q}_j, \breve{E}_j)$ is constructed. Next, it prunes $\breve{G}_j$ (removing from it all edges e with $\omega(e) > \tau_j$). Denote by $\hat{G}_j = (\hat{Q}_j, \hat{E}_j)$ the resulting pruned graph. The edge set $\hat{E}_j$ is also inserted into the output spanner $\tilde{G}$. We define the *j-level auxiliary spanner* $\tilde{G}_j = (Q_j, \tilde{E}_j)$ to be the graph obtained from the union of the graphs $G^*_j = (Q_j, E^*_j)$ and $\hat{G}_j = (\hat{Q}_j, \hat{E}_j)$. (If $j > \ell - \gamma$, we set $\tilde{G}_j = G^*_j$.)

Next, Part II of Procedure $Process_j$ constructs the j-level *attachment graph* $\mathcal{G}_j = (\hat{Q}_j, \mathcal{E}_j)$, which is the restriction of the j-level auxiliary spanner $\tilde{G}_j$ to the set $\hat{Q}_j$, i.e., $\mathcal{E}_j = \tilde{E}_j(\hat{Q}_j) = \hat{E}_j \cup E^*_j(\hat{Q}_j)$. (Note that all points of Q_j, and so all vertices of $\mathcal{G}_j$, are labeled as either safe or risky.) Then it invokes Procedure *Attach* on the graph $\mathcal{G}_j = (\hat{Q}_j, \mathcal{E}_j)$. This procedure returns a star forest Γ_j that that covers R_j (i.e., $R_j \subseteq \bigcup_{S \in \Gamma_j} V(S)$), where $R_j \subseteq \hat{Q}_j$ is the set of all risky points in $\hat{Q}_j$. Each star $S \in \Gamma_j$ is centered at a center $s \in \hat{Q}_j$, which is either safe or risky. The star S also contains one or more leaves $q_1, \ldots, q_k \in \hat{Q}_j$, $k \geq 1$, which are all risky.

Intuitively, risky bags cannot be left on their own, because the degrees of their points will inevitably explode. (See Section 1.3.) Hence the algorithm merges them either with one another, or with some safe bags. The attachment graph $\mathcal{G}_j$ is used to determine which bags will merge. A special care is taken to exclude zombie representatives from $\mathcal{G}_j$. Recall that a zombie bag is already on its way to be merged with some other bag. If another bag were merged into a zombie bag, this would ultimately lead to the creation of "zombie paths", i.e., paths $(z_1, z_2, \ldots, z_h)$ of zombie bags, where z_1 merges into z_2, ..., z_{h-1} merges into z_h. This would result in an uncontrolled growth of the diameter.

Next, Part II of Procedure $Process_j$ performs attachments. Specifically, for each star $S \in \Gamma_j$ with center s and leaves $q_1, \ldots, q_k$, the host bags $v(q_1), \ldots, v(q_k)$ of $q_1, \ldots, q_k$, respectively, are attached to the host bag $v(s)$ of s. As a result the parent bag $\pi(v(s)) = v^{(1)}(s)$ of $v(s)$ adopts the bags $v(q_1), \ldots, v(q_k)$. In other words, the $\gamma - 1$ immediate $\mathcal{F}$-ancestors (ancestors in $\mathcal{F}$) of $v(s)$, $v^{(1)}(s) = \pi(v(s)), \ldots, v^{(\gamma-1)}(s)$, are labeled as incubators, and the $\gamma - 1$ immediate $\mathcal{F}$-ancestors $v^{(1)}(q_i), \ldots, v^{(\gamma-1)}(q_i)$ of $v(q_i)$, for each $i \in [k]$, are labeled as zombies. For each $i \in [k]$, $v^{(\gamma-1)}(q_i)$ is a disappearing zombie, and $v^{(\gamma)}(s)$ is the actual adopter. We say that $v(s)$ *performs k attachments* $\mathcal{A}(v(s), v(q_1)), \ldots, \mathcal{A}(v(s), v(q_k))$. The bag $v(s)$ is the initiator of all these k attachments. The edges $\{(s, q_i) \mid i \in [k]\}$ that connect the center s of the star S with the leaves q_i of S, $i \in [k]$, belong to the attachment graph $\mathcal{G}_j = (\hat{Q}_j, \mathcal{E}_j)$, and they are inserted into the auxiliary spanner $\tilde{G}_j$, and consequently, into the spanner $\tilde{G}$. For each $i \in [k]$, we say that the spanner edge $(s, q_i) = (r(v(s)), r(v(q_i)))$ is a *representing edge* of the attachment $\mathcal{A}(v(s), v(q_i))$. This completes the description of Part II of Procedure $Process_j$.

Next, if $j \in [\ell - 1]$, Procedure $Process_j$ moves to *Part III* of Procedure $Process_j$. (If $j = \ell$, Part III is skipped.) Specifically, it computes the sets $\mathcal{S}(v)$ and $\mathcal{J}(v)$ of surviving children and step-children, respectively, for every bag $v \in \mathcal{F}_{j+1}$. This is done according to the set of attachments which were computed in previous levels. In particular, a child w of v which joins some other $(j + 1)$-level vertex $u, u \neq v$, is excluded from $\mathcal{S}(v)$. Such a bag w is a disappearing zombie, and a step-child of u. Similarly, a bag $z \in \mathcal{F}_j$ with $\pi(z) \neq v$, which is a step-child of v, joins the set $\mathcal{J}(v)$. Given the sets $\mathcal{S}(v)$ and $\mathcal{J}(v)$, Procedure $Process_j$ computes the sets $B(v), K(v), Q(v)$ (according to the rules specified in Section 2.1), and selects the representative $r(v)$ of v (as the least loaded point in $K(v)$). This completes the description of Part III (the final part) of Procedure $Process_j$.

3. ANALYSIS

This section is devoted to the analysis of the spanner $\tilde{G}$ constructed by Algorithm *LightSp*.

To analyze the number of edges and weight in $\tilde{G} = (Q, \tilde{E})$, note that, roughly speaking, $\tilde{G}$ is a union of $\ell+1$ auxiliary spanners $\tilde{G}_j = (Q_j, \tilde{E}_j), j \in [0,\ell]$. (It also contains the base edge set $\mathcal{B}$ and the path-spanner H. However, their contribution can be neglected.) For the case of Euclidean and doubling metrics, $|\tilde{E}_j| = O(|Q_j|)$, for each $j \in [0,\ell]$. Hence $|\tilde{E}| \approx \sum_{j=0}^{\ell} |\tilde{E}_j| = O(\sum_{j=0}^{\ell} |Q_j|)$. Note that $|Q_j| \le \min\{n, n_j\} = \min\left\{n, \frac{c \cdot n}{\rho^{j-1}}\right\}$. Since the sequence $|Q| = |Q_0|, |Q_1|, \ldots, |Q_\ell|$ decays geometrically, $|\tilde{E}| = O(|Q_0|) = O(n)$. (Formally, the sequence starts to decay from the $O(\log_\rho(t/\epsilon))$th element, and thus $|\tilde{E}| = O(n \cdot \log_\rho(t/\epsilon))$.) For the weight analysis, recall that each auxiliary spanner $\tilde{G}_j$ is pruned according to the weight threshold τ_j. These thresholds grow geometrically, at the same rate as the cardinalities of the sets Q_j decay. That is, $|Q_j| \le \frac{O(|Q_0|)}{\rho^{j-1}}$, and $\tau_j = \rho^j \cdot \tau_0$. Hence, roughly speaking, $\omega(\tilde{E}) \approx \sum_{j=0}^{\ell} \omega(\tilde{E}_j) = \sum_{j=0}^{\ell} O(|Q_j| \cdot \tau_j) = O(\rho \cdot \log_\rho n) \cdot \omega(MST(M[Q]))$. For general metrics the analysis is very similar to the above. The analysis of the running time follows the same pattern. (A rigorous analysis of the number of edges, weight and running time is simple, and is deferred to the full version [13].)

Some basic properties of zombie bags are required for the analysis of the stretch, the diameter, and the degree of $\tilde{G}$.

First, it is easy to see that if v is a non-empty j-level bag then it has a useful (i.e., non-zombie) descendant v' in some level $j', j-\gamma+1 \le j' \le j$. Moreover, there is a path $\Upsilon_{v'}$ of useful bags leading down from v' to some 1-level bag. Now suppose that v has a zombie child z. For z to become a zombie, some relatively close (at distance at most $\gamma-1$ in $\mathcal{F}$) descendant z' of z must be a lonely attached bag. It follows that all other children u of v are empty, as otherwise z' would not have been a lonely bag. Also, by a similar argument, v has no step-children. Therefore, if v has a zombie child z then $\mathcal{S}(v) = \{z\}$ and $\mathcal{J}(v) = \emptyset$.

Consider a j-level disappearing zombie $w, j \ge \gamma$. Let $\tilde{w}$ be the lonely attached descendant of w, i.e., $\tilde{w} \in \mathcal{F}_{j-(\gamma-1)}$. Denote by $\tilde{w}^{(0)} = \tilde{w}, \tilde{w}^{(1)} = \pi(\tilde{w}^{(0)}), \ldots, w = \tilde{w}^{(\gamma-1)} = \pi(\tilde{w}^{(\gamma-2)})$ the path in $\mathcal{F}$ between $\tilde{w}$ and w. Then, by the above argument, for every $i \in [\gamma-1]$, $\tilde{w}^{(i-1)}$ is the only child of $\tilde{w}^{(i)}$, and $\tilde{w}^{(i)}$ has no step-children. Hence the point sets of all these bags are equal, i.e., $Q(\tilde{w}^{(0)}) = Q(\tilde{w}^{(1)}) = \ldots = Q(\tilde{w}^{(\gamma-1)})$. These bags are called *identical copies of* w. (A more detailed analysis of properties of zombies and incubators is deferred to the full version [13].)

The incubation period γ is set as $\gamma = c_0 \cdot (\lceil \log_\rho t \rceil + \lceil \log_\rho c \rceil + 1)$, for a sufficiently large constant c_0. (Recall that $c = \Theta(t/\epsilon)$, hence $\gamma = \Theta(\log_\rho(t/\epsilon))$.)

3.1 Stretch and Diameter.

Lemma 3.1. *Fix any $j \in [\ell]$, and let $v \in \mathcal{F}_j, Q(v) \ne \emptyset$. For every $p \in Q(v)$, there is a path $\Pi_j(p)$ in $\tilde{G}$ that leads to a point $b_j(p)$ in the base point set $B(v)$ of v, having weight at most $\frac{1}{2} \cdot \mu_j$ and at most 3ℓ edges. Moreover, if $p \in K(v)$, then $\Pi_j(p)$ has at most 2ℓ edges. All points in $\Pi_j(p)$ belong to $Q(v)$. (The point $b_j(p)$ is called the* base point *of p.)*

Proof. The proof is by induction on j. The basis $j = 1$ is immediate. For the induction step, let $j \ge 2$, and $p \in Q(v)$. Also, let $u \in \chi(v) \subseteq \mathcal{F}_{j-1}$ be the $(j-1)$-level host bag of p.

Suppose first that $u \in \mathcal{S}(v)$, i.e., u is a surviving child of v in $\mathcal{F}$. In this case, $B(u) \subseteq B(v), K(u) \subseteq K(v), Q(u) \subseteq Q(v)$. Consider the path $\Pi_{j-1}(p)$ between p and its base

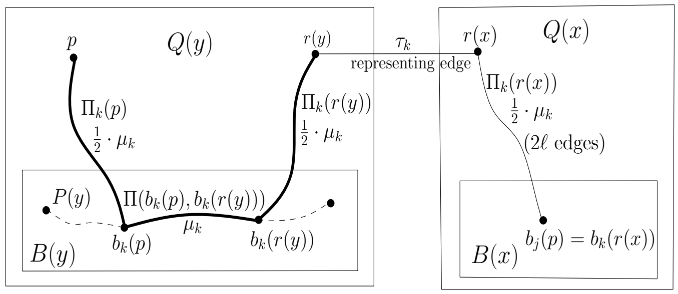

Figure 2: The path $\Pi(p, r(y))$ is depicted by a bold solid line. It is a sub-path of the path $\Pi_j(p)$, which connects p with $b_j(p)$, and is depicted by a solid line.

point $b_{j-1}(p) \in B(u) \subseteq B(v)$ guaranteed by the induction hypothesis for u. Its weight is at most $\frac{1}{2} \cdot \mu_{j-1} = \frac{1}{2} \cdot \frac{\mu_j}{\rho} < \frac{1}{2} \cdot \mu_j$ and it consists of at most 3ℓ edges. Also, all points of $\Pi_{j-1}(p)$ belong to $Q(u) \subseteq Q(v)$. Suppose now that $p \in K(v)$. Recall that u is the unique $(j-1)$-level bag such that $p \in Q(u)$. Since $K(v) \subseteq \bigcup_{z \in \chi(v)} K(z)$ and each kernel set $K(z)$ is contained in $Q(z)$, it follows that $p \in K(u)$. By the induction hypothesis, $\Pi_{j-1}(p)$ consists of at most 2ℓ edges. Thus, we set $\Pi_j(p) = \Pi_{j-1}(p)$ and $b_j(p) = b_{j-1}(p)$.

We henceforth assume that u is a disappearing zombie, i.e., $u \in \mathcal{J}(v)$ is a step-child of v. Since $v \in \mathcal{F}_j$ is an actual adopter, $j \ge \gamma + 1$. For each $i \in [0, \gamma-1]$, let $y^{(i)}$ be the $(j-\gamma+i)$-level *copy* of u. (For each of these copies $y^{(i)}$, $Q(y^{(i)}) = Q(u)$.) By construction, an attachment $\mathcal{A}(x,y)$, for some $(j-\gamma)$-level bag $x = x^{(0)}$, occurs during the $(j-\gamma)$-level processing. As a result of this attachment, $y = y^{(0)}$ became an attached bag, and the corresponding disappearing zombie is $y^{(\gamma-1)} = u$. The initiator bag x of this attachment is a descendant of the actual adopter v. The bags $x^{(1)} = \pi(x^{(0)}), x^{(2)} = \pi(x^{(1)}), \ldots, x^{(\gamma-1)} = \pi(x^{(\gamma-2)})$ are labeled as incubators as a result of this attachment. Observe that $v = x^{(\gamma)} = \pi(x^{(\gamma-1)})$. The attachment $\mathcal{A}(x,y)$ is represented by the edge $(r(x), r(y))$ in $\tilde{G}$.

We will use the next claim to prove Lemma 3.1. (The proof of the claim is omitted; see Figure 2 for an illustration.)

Claim 3.2. *Let $k = j - \gamma$. There is a path $\Pi(p, r(y))$ in $\tilde{G}$ between p and $r(y)$ with weight at most $2 \cdot \mu_k$ and at most $\ell - 2$ edges. All points of $\Pi(p, r(y))$ belong to $Q(y) \subseteq Q(v)$.*

At this point we have a "good path" $\Pi(p, r(y)) \circ (r(y), r(x))$ from p to $r(x)$. We now need to "connect" $r(x)$ to a base point $b_j(p) \in B(v)$. Since x is a descendant of v, $B(x) \subseteq B(v), K(x) \subseteq K(v), Q(x) \subseteq Q(v)$. Also, as a representative of a bag belongs to its kernel, $r(x) \in K(x)$. By the induction hypothesis for x, there exists a path $\Pi_k(r(x))$ between $r(x)$ and its base point $b_k(r(x)) \in B(x) \subseteq B(v)$ in the spanner $\tilde{G}$. Moreover, all points of this path belong to $Q(x) \subseteq Q(v)$. In addition, the weight of this path is at most $\frac{1}{2} \cdot \mu_k = \frac{1}{2} \cdot \frac{\mu_j}{\rho^\gamma}$, and since $r(x) \in K(x)$, it consists of at most 2ℓ edges. We set $b_j(p) = b_k(r(x)) \in B(v)$, and $\Pi_j(p) = \Pi(p, r(y)) \circ (r(y), r(x)) \circ \Pi_k(r(x))$. (See Figure 2.) Notice that $\Pi_j(p)$ is a path between p and its base point $b_j(p) = b_k(r(x))$, and that all points of $\Pi_j(p)$ belong to $Q(v)$. Note also that $\omega(r(y), r(x)) \le \tau_k$. Therefore, the total weight $\omega(\Pi_j(p))$ of the path $\Pi_j(p) = \Pi(p, r(y)) \circ (r(y), r(x)) \circ \Pi_k(r(x))$ satisfies (for sufficiently large c_0) $\omega(\Pi_j(p)) \leq 2\cdot\mu_k+\tau_k+\frac{1}{2}\cdot\mu_k < \frac{1}{2}\cdot\mu_j$. Also, we have $|\Pi_j(p)| \leq (\ell-2)+1+2\ell \leq 3\ell$.

Suppose now that $p \in K(v)$. We argue that in this case x is a small bag. (This case is characterized by $u \in \mathcal{J}(v), p \in Q(u) \cap K(v)$.) Suppose for contradiction otherwise, and consider the $(j-1)$-level ancestor $x^{(\gamma-1)}$ of x. Observe that $K'(v) = \bigcup_{z\in\mathcal{S}(v)} K(z) \supseteq K(x^{(\gamma-1)}) \supseteq K(x)$. By Lemma 2.1, $|K'(v)| \geq |K(x)| \geq \ell$. By construction, $K(v) = K'(v) = \bigcup_{z\in\mathcal{S}(v)} K(z)$. Hence $K(v)$ contains only points from the kernel sets of v's surviving children, and contains no points from v's step-children. However, $p \in Q(u)$, and u is a step-child of v. Hence $p \notin K(v)$, a contradiction. Therefore x is a small bag. Since all points of $\Pi_k(r(x))$ belong to $Q(x)$ and $|Q(x)| < \ell$, it follows that $|\Pi_k(r(x))| \leq \ell-2$ (while in general $|\Pi_k(r(x))| \leq 2\ell$). Consequently, $|\Pi_j(p)| \leq (\ell-2)+1+(\ell-2) \leq 2\ell$. $\square$

Lemma 3.1 implies the following corollary.

Corollary 3.3 (Proof omitted). *Fix any $j \in [\ell]$, and let $v \in \mathcal{F}_j, Q(v) \neq \emptyset$. There is a path in $\tilde{G}$ between every pair of points in $Q(v)$, having weight at most $2\cdot\mu_j$ and $O(\log_\rho n + \alpha(\rho))$ edges. In particular, the metric distance between any two points in $Q(v)$ is at most $2\cdot\mu_j$.*

Recall that $\Lambda(n)$ is an upper bound on the diameter of the auxiliary spanners, produced by Algorithm *BasicSp*.

Lemma 3.4. *For any $p, q \in Q$, there is a $(t+\epsilon)$-spanner path in $\tilde{G}$ with $O(\Lambda(n) + \log_\rho n + \alpha(\rho))$ edges.*

Proof. We start the proof with the next observation.

Observation 3.5 (Proof omitted). *Fix any index $j \in [0,\ell]$. For any pair $u, v \in \mathcal{F}_j$ of non-empty j-level bags, such that $\delta(r(u), r(v)) \leq \frac{\tau_j}{t}$, there is a t-spanner path in G_j^* between $r(u)$ and $r(v)$ with at most $\Lambda(n)$ edges.*

Now we prove Lemma 3.4. Let $p, q \in Q$.
Suppose first that $\delta(p,q) \leq \frac{L}{n} < \frac{\tau_0}{t}$. By Observation 3.5, the 0-level auxiliary spanner $\tilde{G}_0 = G_0^*$ provides a t-spanner path between p and q with at most $\Lambda(n)$ edges.
We henceforth assume that $\delta(p,q) > \frac{L}{n}$. Let $j \in [\ell]$ be the index such that $\rho^{j-1}\cdot\frac{L}{n} < \delta(p,q) \leq \rho^j\cdot\frac{L}{n}$, i.e., $\xi_j < \delta(p,q) \leq \rho\cdot\xi_j$. Let $u = v_j(p)$ (respectively, $w = v_j(q)$) be the j-level host bag of p (resp., q). By Corollary 3.3, the metric distance between every pair of points in the same j-level bag is at most $2\cdot\mu_j < \xi_j$. Since $\delta(p,q) > \xi_j$, it follows that $u \neq w$. Consider the representative $r(u) \in Q_j$ (respectively, $r(w) \in Q_j$) of u (resp., w); by Corollary 3.3, $\delta(p, r(u)), \delta(q, r(w)) \leq 2\cdot\mu_j = 2\cdot\frac{\xi_j}{c}$. It follows that

$$\begin{aligned}\delta(r(u), r(w)) &\leq \delta(r(u),p)+\delta(p,q)+\delta(q,r(w))\\ &\leq \delta(p,q)+4\cdot\frac{\xi_j}{c} \;\leq\; 2\rho^j\cdot\frac{L}{n}\cdot\left(\frac{1}{2}+\frac{2}{\rho\cdot c}\right)\\ &\leq 2\rho^j\cdot\frac{L}{n}\cdot\left(1+\frac{1}{c}\right) \;=\; \frac{\tau_j}{t}.\end{aligned}$$

By Observation 3.5, there is a t-spanner path between $r(u)$ and $r(w)$ in G_j^* (and thus in $\tilde{G}$) with at most $\Lambda(n)$ edges, denoted by $\Pi^*(r(u), r(w))$; note that $\omega(\Pi^*(r(u), r(w))) \leq t\cdot\delta(r(u), r(w))$. By Corollary 3.3, the spanner $\tilde{G}$ contains a path $\Pi(p, r(u))$ (respectively, $\Pi(q, r(w))$) between p and $r(u)$ (resp., between q and $r(w)$) that has weight at most $2\cdot\mu_j = 2\cdot\frac{\xi_j}{c}$ and $O(\log_\rho n + \alpha(\rho))$ edges.

Let $\Pi(p,q) = \Pi(p, r(u)) \circ \Pi^*(r(u), r(w)) \circ \Pi(q, r(w))$. The path $\Pi(p,q)$ connects between p and q in $\tilde{G}$, has weight $\omega(\Pi(p,q)) \leq t\cdot\delta(r(u), r(w)) + 4\cdot\frac{\xi_j}{c}$, and consists of $O(\Lambda(n) + \log_\rho n + \alpha(\rho))$ edges. We showed that $t\cdot\delta(r(u), r(w)) \leq t\cdot(\delta(p,q) + 4\cdot\frac{\xi_j}{c})$. Recall that $c = \lceil\frac{4\cdot(t+1)}{\epsilon}\rceil$. It follows that

$$\omega(\Pi(p,q)) \;\leq\; \left(t + \frac{4\cdot(t+1)}{c}\right)\cdot\delta(p,q) \;\leq\; (t+\epsilon)\cdot\delta(p,q).$$

Hence $\Pi(p,q)$ is a $(t+\epsilon)$-spanner path between p and q. $\square$

3.2 Degree. Instead of selecting the representative $r(v)$ of a bag v to be the least loaded point from $K(v)$, we specify a more elaborate (yet similar) rule for Algorithm *LightSp* to select representatives. This modification is not required, but it simplifies the degree analysis to some extent.

As mentioned in Section 2.3, the algorithm maintains the *load counter* $load\_ctr_j(p)$, for each point $p \in Q$. The algorithm also maintains some refined load counters for each $p \in Q$. Let the *small counter* $ctr_j(p)$ (respectively, *large counter* $CTR_j(p)$) denote the number of indices $i \in [j]$, such that p is not isolated in $\tilde{G}_i$ *and* its host bag $v_i(p)$ is small (resp., large). Note that $load\_ctr_j(p) = ctr_j(p) + CTR_j(p)$. The algorithm also counts the number of indices $i \in [j]$, such that p is not isolated in $\tilde{G}_i$ *and* its host bag $v_i(p)$ satisfies $Q(v_i(p)) = \{p\}$. This counter $single\_ctr_j(p)$ is called the *single counter* of p. The complementary counter (with respect to $ctr_j(p)$), denoted by $plain\_ctr_j(p) = ctr_j(p) - single\_ctr_j(p)$, is called the *plain counter* of p.

The representative $r(v)$ of a non-empty 1-level bag $v \in \mathcal{F}_1$ is selected arbitrarily from $K(v) = Q(v)$. Next, consider a non-empty j-level bag, $j \in [2,\ell]$. If v is large (i.e., $|Q(v)| \geq \ell$), Algorithm *LightSp* appoints a point $p \in K(v)$ with the smallest large counter $CTR_{j-1}(p)$ as its representative $r(v)$.

If v is small (i.e., $1 \leq |Q(v)| < \ell$), the algorithm checks if it is a growing bag or a stagnating one. The bag v is said to be *growing* if $|\chi(v)| \geq 2$. (The way in which $\chi(v)$ is computed was described in Section 2.) Otherwise, v is called *stagnating*. If v is stagnating, it is easy to see that $\mathcal{J}(v) = \emptyset$, and $\mathcal{S}(v) = \{w\}$, for some $(j-1)$-level bag w. In this case Algorithm *LightSp* sets $r(v) = r(w)$. Otherwise, v is a growing small bag. In this case Algorithm *LightSp* appoints a point $p \in K(v)$ with the smallest plain counter $plain\_ctr_{j-1}(p)$ as its representative $r(v)$.

In what follows we prove the following statement.

Proposition 3.6. *For each $q \in Q$, $load\_ctr_\ell(q) = O(\gamma)$.*

Proposition 3.6 implies that each point $q \in Q$ is loaded by at most $O(\gamma)$ auxiliary spanners. Each auxiliary spanner has degree $O(\Delta(n))$. Hence the degree of the union of all auxiliary spanners is $O(\Delta(n)\cdot\gamma)$. The degree $\Delta(\mathcal{B})$ of the base edge set $\mathcal{B}$ is $O(1)$, and the path spanner H has degree $\Delta(H) = O(\rho)$. Hence the ultimate spanner $\tilde{G}$ produced by the algorithm has degree $\Delta(\tilde{G}) = O(\Delta(n)\cdot\gamma+\rho) = O(\Delta(n)\cdot\log_\rho(t/\epsilon)+\rho)$. Thus, we only need to prove Proposition 3.6.

Recall that a bag v is a parent of u in $\hat{\mathcal{F}}$ iff $u \in \chi(v)$, i.e., u is an extended child of v (either a surviving child or a step-child of v). We denote the parent-child relation in $\hat{\mathcal{F}}$ by $\hat{\pi}(\cdot)$, i.e., we write $v = \hat{\pi}(u)$. Note that the forests $\mathcal{F}$ and $\hat{\mathcal{F}}$ are very similar. The only bags v that have step-parents (different from their parents) are disappearing zombies.

Observation 3.7. *For a bag $v \in \mathcal{F}_j, j \in [\ell-1]$, which is not a disappearing zombie, $\pi(v) = \hat{\pi}(v)$.*

Our algorithm makes a persistent effort to merge small bags together to form large bags. Intuitively, a large bag is easy to handle because its kernel contains enough (at least ℓ) points to share the load. In fact, our construction guarantees that any large bag will be assigned a "fresh representative" (with large counter 0). Hence large counters are at most 1.

LEMMA 3.8. *For each $q \in Q$, $CTR_\ell(q) \leq 1$.*

Next, we bound the small counter of points.

For a small bag v, its representative $r(v)$ is loaded by a j-level auxiliary spanner only if $r(v)$ is not isolated in G_j^* (see Section 2.4). It means that there is another j-level representative $r(u)$, such that $\delta(r(v), r(u)) \leq \tau_j$; in other words, $r(u)$ is close to $r(v)$. Intuitively, we will want the bags v and u to merge, as this would increase the pool of eligible representatives. We cannot merge them right away, however, because this would blow up the weighted diameters of the $(j+1)$-level bags. Instead we wait for $\gamma = O(1)$ levels, and then merge v into the $(j+\gamma)$-level ancestor u' of u. (Or the other way around, merge u into the $(j+\gamma)$-level ancestor v' of v.) The weighted diameters of the j-level bags are proportional to the length μ_j of the j-level intervals, i.e., they grow geometrically with the level j. Hence when v is merged into u', it contributes only an $\exp(-\Omega(\gamma))$-fraction to the weighted diameter of the $(j+\gamma)$-level bag u'. In this way we control the weighted diameters of bags, while always maintaining sufficiently large pools of eligible representatives. During the γ levels $j, j+1 \ldots, j+\gamma-1$, points of v do accumulate some extra degree; however, since $\gamma = O(1)$, they are overloaded by a constant factor. Next, we formalize this intuition.

A bag $v \in \mathcal{F}_j$ is called *active* if $r(v)$ is not isolated in G_j^*. To control small counters of points, we show that once a small bag becomes active, it will soon get merged into a larger bag. These merges will allow for a uniform load-sharing, resulting in load $O(\gamma)$ for all points.

For an integer $\beta \geq 1$, we say that a bag $v \in \mathcal{F}_j$ is β-*prospective*, if one of its β immediate $\hat{\mathcal{F}}$-ancestors $\hat{v}^{(1)} = \hat{\pi}(v), \hat{v}^{(2)} = \hat{\pi}(\hat{v}^{(1)}), \ldots, \hat{v}^{(\beta)} = \hat{\pi}(\hat{v}^{(\beta-1)})$ is a growing bag. (For $j > \ell - \beta$, all bags $v \in \mathcal{F}_j$ are called β-prospective.)

LEMMA 3.9. *Any active small bag is $O(\gamma)$-prospective.*

PROOF. First, we argue that any small safe bag is γ-prospective. Such a bag v is either crowded, or a zombie, or an incubator. We show how to handle the case where v is crowded. The cases where v is a zombie or an incubator are handled similarly. Suppose that $v \in \mathcal{F}_j$, with $j \leq \ell - \gamma$. (The case $j > \ell - \gamma$ is trivial.) Since v is crowded, its cage $\mathcal{C}(v)$ contains another useful bag $u \in \mathcal{F}_j$. Note that u is crowded as well, and thus both v and u are safe. Let $w \in \mathcal{F}_k$ be the least common $\mathcal{F}$-ancestor of v and u. The index k satisfies $j+1 \leq k \leq j+\gamma$. Write $v = v^{(0)}, u = u^{(0)}$, and consider the $(k-j)$ immediate $\mathcal{F}$-ancestors of v and u, $v^{(1)} = \pi(v^{(0)}), \ldots, v^{(k-j)} = \pi(v^{(k-j-1)}) = w$ and $u^{(1)} = \pi(u^{(0)}), \ldots, u^{(k-j)} = \pi(u^{(k-j-1)}) = w$, respectively. It is easy to see by induction that all these bags, except maybe w itself, are crowded and safe. Hence none of them is a zombie, and so for each $i \in [k-j], v^{(i-1)} \in \mathcal{S}(v^{(i)}) \subseteq \chi(v^{(i)}), u^{(i-1)} \in \mathcal{S}(u^{(i)}) \subseteq \chi(u^{(i)})$. It follows that $v^{(k-j-1)}$ (respectively, $u^{(k-j-1)}$) is an $\hat{\mathcal{F}}$-ancestor of v (resp., u), and $v^{(k-j-1)}, u^{(k-j-1)} \in \mathcal{S}(w) \subseteq \chi(w)$. Hence w is the least common $\hat{\mathcal{F}}$-ancestor of v and u, and $|\chi(w)| \geq |\mathcal{S}(w)| \geq 2$. Thus w is a growing bag, and v is γ-prospective.

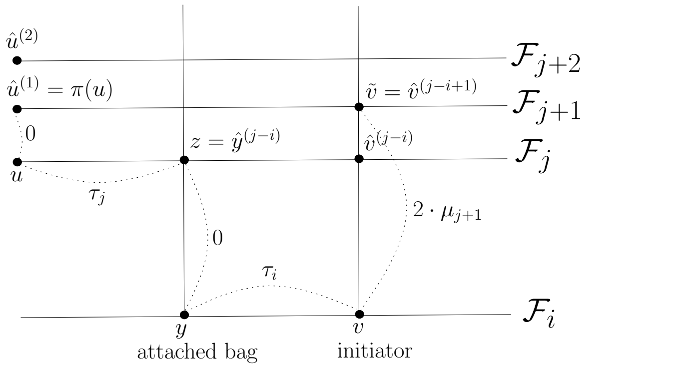

Figure 3: An illustration for the proof of Lemma 3.9.

We will assume that $t \leq \rho/2$. As ρ can be taken to be an arbitrarily large constant, this would cover the case of constant stretch t. In particular, to prove Theorem 1.1 one only needs $t = 1+\epsilon$. The argument for any t (regardless of ρ) is more involved, and is deferred to the full version [13].

Consider an active small bag $u \in \mathcal{F}_j$, and its two immediate $\hat{\mathcal{F}}$-ancestors $\hat{u}^{(1)} = \hat{\pi}(u), \hat{u}^{(2)} = \hat{\pi}(\hat{u}^{(1)})$. We assume that $j \leq \ell - \gamma$, and that $\hat{u}^{(1)}$ is stagnating. (The complementary cases are trivial.) To prove the lemma, we show that at least one among the bags $u, \hat{u}^{(1)}, \hat{u}^{(2)}$ is safe.

Suppose for contradiction that $u, \hat{u}^{(1)}, \hat{u}^{(2)}$ are all risky. Since u is active, these bags are non-empty. Recall that zombie bags are safe, which implies that these bags are useful. By Observation 3.7, we have $\hat{u}^{(1)} = u^{(1)} = \pi(u), \hat{u}^{(2)} = u^{(2)}$.

Consider the j-level processing (see Section 2.4). Since u is active, $r(u)$ is non-isolated in G_j^*. As u is useful, $r(u) \in \hat{Q}_j$.

Assume first that $r(u)$ is non-isolated in the j-level attachment graph G_j. Since u is risky, it is either an attachment initiator or an attached bag, and its parent $\pi(u) = \hat{u}^{(1)}$ is safe. This is a contradiction.

We henceforth assume that $r(u)$ is isolated in G_j. By construction, $r(u)$ must be incident to a zombie representative $r(z)$ in G_j^*, and so $\delta(r(u), r(z)) \leq \tau_j$. (See Figure 3.) Consider the corresponding attachment between an i-level descendant y of z and some i-level initiator bag v, where $j-(\gamma-1) \leq i < j$. By construction, y is an identical copy of z, and so $r(y) = r(z)$. Also, we have $\delta(r(y), r(v)) \leq \tau_i$. Denote by $\tilde{v}$ the $(j+1)$-level $\hat{\mathcal{F}}$-ancestor of v. Since $j+1-i \leq \gamma$, it follows that $\tilde{v}$ is either an incubator or the actual adopter of z. Hence $\tilde{v}$ cannot be a zombie (though it may become an attached bag during the $(j+1)$-level processing). Recall that $\pi(u) = \hat{u}^{(1)}$ is a stagnating bag. Hence it is a copy of u, and so $r(\pi(u)) = r(u)$. Observe that $\pi(u) \neq \tilde{v}$.

Next, we argue that $\delta(r(\pi(u)), r(\tilde{v})) \leq \frac{\tau_{j+1}}{t}$. Since $\tilde{v}$ is an $\hat{\mathcal{F}}$-ancestor of v, we have $r(v) \in K(v) \subseteq Q(v) \subseteq Q(\tilde{v})$. By Corollary 3.3, the metric distance between every pair of points in the same $(j+1)$-level bag is at most $2 \cdot \mu_{j+1}$. Consequently, $\delta(r(\tilde{v})), r(v)) \leq 2 \cdot \mu_{j+1}$. By the triangle inequality,

$$\begin{aligned}
\delta(r(\pi(u)), r(\tilde{v})) &\leq \delta(r(\pi(u)), r(u)) + \delta(r(u), r(z)) \\
&\quad + \delta(r(z), r(y)) + \delta(r(y), r(v)) + \delta(r(v), r(\tilde{v})) \\
&\leq 0 + \tau_j + 0 + \tau_i + 2 \cdot \mu_{j+1} \\
&= \frac{\tau_{j+1}}{\rho} + \frac{\tau_{j+1}}{\rho^{j+1-i}} + \frac{\tau_{j+1}}{\rho \cdot t \cdot (c+1)} \\
&\leq \frac{\tau_{j+1}}{t} \cdot \left(\frac{1}{2} + \frac{1}{4t} + \frac{1}{4}\right) \leq \frac{\tau_{j+1}}{t}.
\end{aligned}$$

(The one before last inequality holds for $\rho \geq 2t \geq 2, c \geq 1$.)

Notice that $r(\pi(u)), r(\tilde{v}) \in Q_{j+1}$. Since $\delta(r(\pi(u)), r(\tilde{v})) \leq \frac{\tau_{j+1}}{t}$, it follows that $r(\pi(u))$ and $r(\tilde{v})$ are non-isolated in G^*_{j+1}. (Indeed, the t-spanner path between these points in G'_{j+1} has total weight at most τ_{j+1}, and thus it will not get pruned.) Since $\pi(u)$ and $\tilde{v}$ are non-zombies, $r(\pi(u)), r(\tilde{v}) \in \hat{Q}_{j+1}$. Using again the fact that $\delta(r(\pi(u)), r(\tilde{v})) \leq \frac{\tau_{j+1}}{t}$, we conclude that $r(\pi(u))$ is non-isolated in the $(j+1)$-level attachment graph. Since $\pi(u) = \hat{u}^{(1)}$ is risky, it must be either an attachment initiator or an attached bag, and its parent $\hat{u}^{(2)}$ is safe. This again leads to a contradiction. □

Note that for any $\hat{\mathcal{F}}$-ancestor v' of a growing bag v, it must hold that $|Q(v')| \geq |Q(v)| \geq 2$. Hence, Lemma 3.9 directly implies that the single counters are bounded by $O(\gamma)$.

Finally, we bound the plain counter of points. It is easy to verify (via a simple induction) that for any growing small bag v, its kernel set $K(v)$ contains at least two points with plain counter 0. Our rule therefore guarantees that, for any growing small bag, we will always select a representative with plain counter 0. The plain counter of such a representative $q \in Q$ can increase only as long as its host bag remains stagnating. Lemma 3.9 implies that this can carry on for at most $O(\gamma)$ levels. From that stage on, this point q will not be selected again as the representative of any growing small bag, because other points in the kernel set will have smaller plain counters (namely, plain counter 0). Therefore, the plain counters are bounded by $O(\gamma)$ too.

We showed that $CTR_\ell(q)$,$single\_ctr_\ell(q)$ and $plain\_ctr_\ell(q)$ are all bounded by $O(\gamma)$, for each $q \in Q$. Hence their sum $load\_ctr_\ell(q) = CTR_\ell(q) + single\_ctr_\ell(q) + plain\_ctr_\ell(q)$ is bounded by $O(\gamma)$ too. Proposition 3.6 now follows.

This concludes the proof of Theorem 1.4.

Acknowledgments. The second-named author is indebted to Michiel Smid for many helpful comments. We thank Adi Gottlieb for referring us to [17] and for helpful discussions.

4. REFERENCES

[1] P. K. Agarwal, Y. Wang, and P. Yin. Lower bound for sparse Euclidean spanners. In *Proc. of 16th SODA*, pages 670–671, 2005.

[2] S. Arya, G. Das, D. M. Mount, J. S. Salowe, and M. H. M. Smid. Euclidean spanners: short, thin, and lanky. In *Proc. of 27th STOC*, pages 489–498, 1995.

[3] S. Arya, D. M. Mount, and M. H. M. Smid. Randomized and deterministic algorithms for geometric spanners of small diameter. In *Proc. of 35th FOCS*, pages 703–712, 1994.

[4] Y. Bartal, L. Gottlieb, and R. Krauthgamer. The traveling salesman problem: low-dimensionality implies a polynomial time approximation scheme. In *Proc. of 44th STOC*, pages 663–672, 2012.

[5] H. T.-H. Chan and A. Gupta. Small hop-diameter sparse spanners for doubling metrics. In *Proc. of 17th SODA*, pages 70–78, 2006.

[6] H. T.-H. Chan, A. Gupta, B. M. Maggs, and S. Zhou. On hierarchical routing in doubling metrics. In *Proc. of 16th SODA*, pages 762–771, 2005.

[7] T.-H. H. Chan, M. Li, and L. Ning. Incubators vs zombies: Fault-tolerant, short, thin and lanky spanners for doubling metrics. *Tech. Rep., CoRR*, abs/1207.0892, July, 2012.

[8] B. Chandra, G. Das, G. Narasimhan, and J. Soares. New sparseness results on graph spanners. In *Proc. of 8th SOCG*, pages 192–201, 1992.

[9] D. Z. Chen, G. Das, and M. H. M. Smid. Lower bounds for computing geometric spanners and approximate shortest paths. *Discrete Applied Mathematics*, 110(2-3):151–167, 2001.

[10] L. P. Chew. There is a planar graph almost as good as the complete graph. In *Proc. of 2nd SOCG*, pages 169–177, 1986.

[11] G. Das and G. Narasimhan. A fast algorithm for constructing sparse Euclidean spanners. In *Proc. of 10th SOCG*, pages 132–139, 1994.

[12] Y. Dinitz, M. Elkin, and S. Solomon. Shallow-low-light trees, and tight lower bounds for Euclidean spanners. In *Proc. of 49th FOCS*, pages 519–528, 2008.

[13] M. Elkin and S. Solomon. Optimal Euclidean spanners: really short, thin and lanky. Tech. Rep. CS12-04, Ben-Gurion University (April, 2012). Also Tech. Rep. CoRR, abs/1207.1831 (July, 2012).

[14] M. Elkin and S. Solomon. Fast constructions of light-weight spanners for general graphs. In *Proc. of 24th SODA*, pages 513–525, 2013.

[15] L. Gottlieb, A. Kontorovich, and R. Krauthgamer. Efficient regression in metric space via approximate lipschitz extension. Manuscript, 2012.

[16] L. Gottlieb and L. Roditty. Improved algorithms for fully dynamic geometric spanners and geometric routing. In *Proc. of 19th SODA*, pages 591–600, 2008.

[17] L. Gottlieb and L. Roditty. An optimal dynamic spanner for doubling metric spaces. In *Proc. of 16th ESA*, pages 478–489, 2008.

[18] J. Gudmundsson, C. Levcopoulos, and G. Narasimhan. Fast greedy algorithms for constructing sparse geometric spanners. *SIAM J. Comput.*, 31(5):1479–1500, 2002.

[19] A. Gupta, R. Krauthgamer, and J. R. Lee. Bounded geometries, fractals, and low-distortion embeddings. In *Proc. of 44th FOCS*, pages 534–543, 2003.

[20] S. Har-Peled and M. Mendel. Fast construction of nets in low-dimensional metrics and their applications. *SIAM J. Comput.*, 35(5):1148–1184, 2006.

[21] G. Narasimhan and M. Smid. *Geometric Spanner Networks*. Cambridge University Press, 2007.

[22] L. Roditty. Fully dynamic geometric spanners. In *Proc. of 23rd SoCG*, pages 373–380, 2007.

[23] J. S. Salowe. On Euclidean spanner graphs with small degree. In *Proc. of 8th SoCG*, pages 186–191, 1992.

[24] M. H. M. Smid. The weak gap property in metric spaces of bounded doubling dimension. In *Proc. of Efficient Algorithms*, pages 275–289, 2009.

[25] S. Solomon. An optimal-time construction of sparse Euclidean spanners with tiny diameter. In *Proc. of 22nd SODA*, pages 820–839, 2011.

[26] S. Solomon. Fault-tolerant spanners for doubling metrics: Better and simpler. *Tech. Rep., CoRR*, abs/1207.7040, July, 2012.

[27] S. Solomon and M. Elkin. Balancing degree, diameter and weight in Euclidean spanners. In *Proc. of 18th ESA*, pages 48–59, 2010.