EMBEDDING, DISTANCE ESTIMATION AND OBJECT LOCATION IN NETWORKS

A Dissertation
Presented to the Faculty of the Graduate School
of Cornell University
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

by
Aleksandrs Slivkins

August 2006

EMBEDDING, DISTANCE ESTIMATION AND OBJECT LOCATION IN NETWORKS

Aleksandrs Slivkins, Ph.D.
Cornell University 2006

Concurrent with numerous theoretical results on metric embeddings, a growing body of research in the networking community has studied the distance matrix defined by node-to-node latencies in the Internet, resulting in a number of recent approaches that approximately embed this distance matrix into low-dimensional Euclidean space. A fundamental distinction between the theoretical approaches to embeddings and this recent Internet-related work is that the latter operates under the additional constraint that it is only feasible to measure a linear number of node pairs, and typically in a highly structured way. Indeed, the most common framework here is a *beacon-based* approach: one randomly chooses a small number of nodes ('beacons') in the network, and each node measures its distance to these beacons only. Moreover, beacon-based algorithms are also designed for the more basic problem of *triangulation*, in which one uses the triangle inequality to infer the distances that have not been measured.

We give algorithms with provable performance guarantees for triangulation and embedding. We show that in addition to multiplicative error in the distances, performance guarantees for beacon-based algorithms typically must include a notion of "slack" – a certain fraction of all distances may be arbitrarily distorted.

For arbitrary metrics, we give a beacon-based embedding algorithm that achieves constant distortion on a $(1-\epsilon)$-fraction of distances; this provides some theoretical justification for the success of the recent networking algorithms, and forms an interesting contrast with lower bounds showing that it is not possible to embed all distances with constant distortion. For doubling metrics (which have been proposed as a reasonable abstraction of Internet latencies), we show that triangulation with a constant number of beacons can achieve multiplicative error $1+\delta$ on a $(1-\epsilon)$-fraction of distances, for arbitrarily small constants ϵ, δ.

We extend these results in a number of directions: embeddings with slack that work for all ϵ at once; distributed algorithms for triangulation and embedding with low overhead on all participating nodes; distributed triangulation with guarantees for all node pairs; node-labeling problems for graphs and metrics; systems project on location-aware node selection in a large-scale distributed network.

Biographical Sketch

Aleksandrs Slivkins was born December 1, 1978 in Riga, Latvia (then Soviet Union). He lived in Riga until he finished high school in August 1996. Then he went to California Institute of Technology; he graduated in June 2000 with B.S. in Mathematics. From August 2000 till present Alex has been a graduate student with the Computer Science department of Cornell University. Following his candidacy exam, he received a M.S. in Computer Science in 2004. He expects to graduate with a Ph.D. in August 2006. After Cornell he is going for a one-year postdoc at Brown University. He has accepted a research staff position at Microsoft Research, Silicon Valley Center starting from July 2007.

Acknowledgements

First of all, I would like to thank my thesis advisor Jon Kleinberg for his invaluable support and mentorship in both research- and career-related issues. I have benefited tremendously from interacting with my other two thesis committee members, Eva Tardos and Emin Gün Sirer. I additionally thank Jon, Eva and Gun for their advice during the job search process.

Throughout my graduate career I have been fortunate to collaborate with many wonderful researchers. It is my pleasure to acknowledge my faculty coauthors (Jon Kleinberg and Gun Sirer from Cornell, Matthew Andrews from Bell Labs, Shuki Bruck from Caltech, and Anupam Gupta from Carnegie Mellon), and thank them for their guidance and collegiality. Furthermore, I would like to thank Shuki Bruck for being my undergraduate research mentor.

I thank graduate students at Cornell Computer Science department for creating a fruitful academic environment. I will try to avoid listing too many names here, but I would like to mention Elliot Anshelevich, Anirban Dasgupta, Ara Hayrapetyan, Martin Pal, Mark Sandler, Zoya Svitkina, and Tom Wexler, as well as several older students: Aaron Archer, David Kempe, Tim Roughgarden, and Chaitanya Swamy.

Finally, I thank my parents and grandparents for supporting and motivating me throughout my entire life, and my fiancee Koralai for making me happy.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The past decade has seen many significant and elegant results in the theory of metric embeddings (for recent surveys, see [Ind01, Lin02, Mat02a, IM04]). Embedding techniques have been valuable in the design and analysis of algorithms that operate on an underlying metric; many optimization problems become more tractable when the given metric is embedded into one that is structurally simpler.

Meanwhile, an active line of research in the networking community has studied the distance matrix defined by node-to-node latencies in the Internet [FJJ$^+$01, GSG02, GS95, HFP$^+$02, KSB01, VPSV02], resulting in a number of recent approaches that approximately embed this distance matrix into low-dimensional Euclidean space [DCKM04, NZ02, PCW$^+$03, ST03].[1] However, there is a fundamental distinction between this Internet-related work and the large body of theoretical work on embedding, due to the following intrinsic problem: *in any analysis of the distance matrix of the Internet, most distances are not available.* The cost of measuring all node-to-node distances is simply too expensive; instead, we have a setting where it is generally feasible to measure the distances among only a linear (or near-linear) number of node pairs, and typically in a highly structured way. Indeed, the most common framework for Internet measurements of this type is a *beacon-based* approach: one chooses uniformly at random a constant number of nodes ('beacons') in the network, each node measures its distance to all beacons, and one then has access to only these $O(n)$ measurements for the remainder of the algorithm. (For example, the data can be shared among the beacons, who then perform computations on the data locally.)

This inability to measure most distances is the inherent obstacle that stands in the way of applying algorithms developed from the theory of metric embeddings, which assume (and use) access to the full distance matrix. Thus, to obtain insight at a theoretical level into recent Internet measurement studies, we need to consider problems in following two genres.

(i) What performance guarantees can be achieved by metric embedding algorithms when only a sparse (beacon-based) subset of the distances can be measured?

(ii) At an even more fundamental level, many Internet measurement algorithms are seeking not to embed but simply to reconstruct the unobserved distances with reasonable accuracy (see e.g. [FJJ$^+$01, GSG02, GS95, KSB01]). Can we give provable guarantees for this type of *reconstruction* task?

Reconstruction via triangulation. Within this framework, we discuss the reconstruction problem (ii) first, as it is a more basic concern. Motivated by the research of Francis et al. on IDMaps [FJJ$^+$01], and

[1] We speak of Internet latencies as defining as a "distance matrix" rather than a metric, since the triangle inequality is not always observed; however, one can view the recent networking research as indicating that severe triangle inequality violations are not widespread enough to prevent the matrix of node-to-node latencies from being usefully modeled using notions from metric spaces.

subsequent work, we formalize the reconstruction problem here as follows. Let S be the set of beacons; and suppose for each node u, and each beacon $b \in S$, we know the distance $d(u, b)$. What can we infer from this data about the remaining unobserved distances $d(u, v)$ (when neither v nor v is a beacon), assuming we know only that we have points in an arbitrary metric space? The triangle inequality implies that

$$\max_{b \in S} |d(u, b) - d(v, b)| \leq d(u, v) \leq \min_{b \in S} d(u, b) + d(v, b), \tag{1.1}$$

and it is easy to see that these are the tightest bounds that can be provided on $d(u, v)$ if we assume only that the underlying metric is arbitrary subject to the given distances. We will say that $d(u, v)$ is reconstructed by *triangulation*[2], with distortion $\Delta \geq 1$, if the ratio between the upper and lower bounds in (1.1) is at most Δ. Since it is much cheaper for nodes to exchange messages than to actually estimate their round-trip distance on the Internet (the latter typically requires a significant measurement period to produce a stable estimate), triangulation can be valuable as a way to assign each node a short *label* — its distances to all beacons — in such a way that the distance $d(u, v)$ can later be estimated by a third party (or by one of v or v) just from their labels. This can be viewed as a kind of *distance labeling*, and we discuss related work on this topic (e.g. [GPPR04]) below.

To give performance guarantees for triangulation, we also need a notion of *slack*. Even in very simple metrics, there will be some distance pairs that cannot be reconstructed well using only a constant number of beacons. Consider for example a set of regularly spaced points on a line (or in a d-dimensional lattice); points u and v that are very close together will have a distance $d(u, v)$ that is much smaller than the distance to the nearest beacon, rendering the upper bound obtainable from (1.1) useless. We therefore say that a set of beacons achieves a triangulation with distortion Δ and slack ϵ if all but an ϵ fraction of node pairs in the metric are reconstructed with distortion Δ.

A fundamental question is then the following. Suppose we have an underlying metric space M, and desired levels of precision $\epsilon > 0$ and $\delta > 0$. Is there a function $f(\cdot, \cdot)$ (independent of the size of M) so that $f(\epsilon, \delta)$ beacons suffice to achieve a triangulation with distortion $1 + \delta$ and slack ϵ? Clearly such a guarantee is not possible for every metric; in the n-point uniform metric, with all distances equal to 1, any distance that is not directly measured will have a lower bound from (1.1) equal to 0. Thus we ask: are there are natural classes of metrics that *are* triangulable in this way?

Beacon-based embedding. The recent work of Ng and Zhang on Global Network Positioning (*GNP*) [NZ02] showed how a beacon-based set of measurements could embed all but a small fraction of Internet distances with constant distortion in low-dimensional Euclidean space, and this result touched off an active line of follow-up embedding studies in the networking literature (e.g. [DCKM04, PCW$^+$03, ST03]). Note that the empirical guarantee for *GNP* naturally defines a notion of ϵ slack for embeddings: an ϵ fraction of all node pairs may have their distances arbitrarily distorted. Again, it is easy to see that this notion of slack is necessary for a beacon-based approach. The *GNP* algorithm forms an interesting contrast with the algorithms of Bourgain and Linial, London, and Rabinovich [Bou85, LLR95] for embedding arbitrary metrics. These latter algorithms use access to the full distance matrix and build coordinates in the embedding by measuring the distance from a point to a *set* — in effect, sets that can be as large as a constant fraction of the space thus act as "super-beacons" in a way that would not be feasible to implement for all nodes in the context of Internet measurement.

In order to understand why beacon-based approaches in general, or the *GNP* algorithm in particular, achieve good performance for Internet embedding in practice, a basic question is the following: are there natural classes of metrics that are embeddable with constant distortion and slack ϵ, using a constant number of beacons?

[2] Note that this is one of several standard uses of the term "triangulation" in the literature; it should not be confused with the process of dividing up a region into simplices, which goes by the same name.

1.1 Overview of results

In this dissertation we provide a theoretical foundation for distributed distance reconstruction approaches. We begin by showing that distances in a metric space M whose doubling dimension is bounded by k can be reconstructed by triangulation with distortion $1 + \delta$ and slack ϵ, using a number of beacons that depends only on δ, ϵ, and dimension k, independent of the size of M. We define the *doubling dimension* here to be the smallest k such that every ball can be covered by at most 2^k balls of half the radius (see Section 2.3.2 for more background); we also call such a metric 2^k-*doubling*. The point here is that we are not assuming a reconstruction method that explicitly knows anything about the doubling properties of M; rather, as long as the number of beacons is simply large enough relative to the doubling dimension, one obtains accurate reconstruction using upper and lower bounds obtained from the triangle inequality alone. Doubling metrics, which generalize the distance matrices of finite d-dimensional point sets, have been the subject of recent theoretical interest in the context of embedding, nearest-neighbor search, and other problems [GKL03, KR02, GKL03, KL04, Tal04]; and an increasing amount of work in the networking community has suggested that the bounded growth rate of balls may be a useful way to capture the structural properties of the Internet distance matrix (see e.g. [FkcHM01, NZ02, PV03, ZHR$^+$04]). Thus, given that strong triangulation performance guarantees are not possible for general metrics (as noted above via the uniform metric), this positive result for doubling metrics serves as a plausible theoretical underpinning for the success of beacon-based triangulation in practice.

Certain non-trivial metrics exhibit a stronger phenomenon that we term *perfect triangulation*: on all but an ϵ-fraction of node pairs, the upper and lower bounds from the triangle inequality agree exactly (i.e. with distortion 1). For example, one can show that $f(d, \epsilon)$ beacons suffice to achieve perfect triangulation with slack ϵ on the points of a d-dimensional lattice under the ℓ_1 metric. It is thus natural to ask how generally this phenomenon holds. Perfect triangulation turns out not be possible for all point sets in the ℓ_1 metric, but we show that it can be achieved for all *dense* point sets in ℓ_1; by a dense point set we mean an n-point subset of $\mathbb{R}^d$ in which the ratio of the largest to the smallest distance is $\Theta(n^{1/d})$.

We next move on to results for beacon-based embedding. We show that every metric can be embedded into ℓ_p (for any $p \geq 1$) with constant distortion and slack ϵ, using a constant number of beacons, where the constants here depend only on ϵ. Moreover, for doubling metrics we show that an embedding with these properties can be achieved by a close analog of the actual *GNP* algorithm of Ng and Zhang, providing further theoretical explanation for its success in practice. It is interesting to note that arbitrary metrics (and even arbitrary doubling metrics) cannot be embedded into Euclidean space (or into ℓ_p for any $p \geq 2$) with constant distortion (see Chapter 2 for more background), so this is a case where allowing slack leads to a qualitatively different result.

While beacon-based algorithms perform a manageable set of measurements, they do so by choosing a small set of nodes and placing a large computational and measurement load on them. Several recent networking papers [DCKM04, PCW$^+$03, ST03] address the unbalanced load of beacon-based methods using *uniform probing*: each node selects a small number of virtual 'neighbors' uniformly at random and measures distances to them; all nodes then run a distributed algorithm that uses the measured distances. We show how an extension of our techniques here can be used to give performance guarantees for distributed algorithms such as these.

In particular, to analyze these uniform-probing embedding algorithms, we build on the techniques we develop for reasoning about triangulation. We consider subgraphs G' on the set of nodes with the property that embeddings that approximately preserve all edge lengths in G' must have constant distortion with slack ϵ for the full distance matrix. This is a kind of "rigidity" property (with slack) that follows naturally from the analysis of triangulation, and we can show that subgraphs consisting of node-to-beacon measurements, as well as subgraphs built in a more distributed fashion, can be usefully analyzed in terms of this property.

We then simulate a beacon-based algorithm: instead of measuring distances to beacons directly, nodes cooperatively infer them from the probed distances via an appropriate distributed algorithm. The inferred distances to beacons are in fact upper and lower bounds on the true distances that are sufficiently precise to yield a good triangulation. To obtain an embedding from these bounds, one needs somewhat more elaborate technique than the one for the 'pure' beacon-based result; this is because the inferred distances do not quite obey the triangle inequality.

We show that stronger guarantees can be obtained in the more restrictive class of *growth-constrained metrics*, in which doubling the radius of a ball increases its cardinality by at most a constant factor. We obtain an embedding with a more "gracefully degrading" notion of slack: all but an ϵ-fraction of distances are embedded with distortion $\Delta = O(\log \frac{1}{\epsilon})$; all but an ϵ-fraction of the remainder are embedded with distortion 2Δ; and in general, all but an ϵ^j fraction are embedded with distortion $j\Delta$. We also show that the following simple *nearest-beacon* embedding is effective in strongly doubling metrics: select k beacons uniformly at random, embed the beacons, and then simply position each other node at the embedded location of its nearest beacon.

Finally, we derive *lower bounds* on embeddings with slack, showing that our embedding result for arbitrary metrics is essentially optimal in terms of distortion. Moreover, we give a very general theorem that allows us to convert the previously known lower bounds on distortion and dimension of embeddings into lower bounds in terms of the slack parameter ϵ. This result works under very mild conditions, and in particular allows us to prove matching or nearly matching lower bounds for our further results on embeddings with slack (see Section 1.1.1).

1.1.1 Extensions

Our results on triangulation and embeddings described above are gathered in Chapter 3. We extend these results in several directions, each direction constituting a separate chapter.

Embeddings with gracefully degrading distortion. Recall that in our result on embeddings with slack ϵ we provide a different embedding for each ϵ. A much more flexible and powerful alternative would be one embedding that works for all ϵ at once; informally, say that such embedding has *gracefully degrading distortion*. We obtained gracefully degrading distortion for growth-constrained metrics as an elaboration of Bourgain's embedding. Extending it to more general metrics is much more challenging. In Chapter 4 we present such embedding for *decomposable metrics*, a notion from previous work on embeddings that we specify precisely in Section 2.3.3; this includes several well-studied classes of metrics including doubling metrics and shortest-paths metrics of planar graphs. The proof of this result is technically the most involved part of this thesis; at a high level, we develop a set of scale-based embeddings which are then combined together (as in most previous embeddings)—however, since the existing ways to perform this do not seem to guarantee gracefully degrading distortion, we construct new ways of defining distance scales.

We further show that gracefully degrading distortion can be achieved for all metrics; however, this result only works for embeddings into ℓ_1 (as opposed to an arbitrary ℓ_p, $p \geq 1$ space), and the resulting embedding is high-dimensional. This question has been subsequently solved in full by Abraham et al. [ABN06], providing, for an arbitrary target space ℓ_p, $p \geq 1$, a low-dimensional embedding with gracefully degrading distortion. It is interesting to note that we provide essentially matching lower bounds for all these embeddings.

Triangulation with guarantees for all node pairs. In Chapter 5 we obtain improvements for distributed algorithms that induce low load on all participating nodes: specifically, we obtain triangulation with distortion $1 + \delta$ for *all node pairs*. To achieve such a result, we need to elaborate both the way we take measurements and the way we handle beacons: neither the uniform probing nor using a single global set of beacons are adequate for the task at hand. Instead, we use a hierarchical approach, whereby each node probes its immediate neighborhood more densely than the faraway regions. A crucial obstacle is that the nodes do not know any distance information in advance (and they are allowed only a poly-logarithmic storage throughout the algorithm), so they have to cooperatively infer this throughout the algorithm.

A crucial element of our construction is *rings of neighbors*, a sparse distributed data structure which captures the distance information in the network. This is also (essentially) the data structure that underlies *Meridian*, a network positioning system described in Chapter 6. The idea is that every node u stores pointers to some nodes called 'neighbors'; these pointers are partitioned into several 'rings' so that the neighbors in the i-th ring are selected near-uniformly in a ball of radius 2^i around u. In effect, rings of neighbors form an overlay network with a certain structure imposed by the rings.[3]

The beacon selection is also hierarchical. First each node selects itself (independently at random) as a *level-i beacon*; we make sure that level-i beacons are sufficiently dense on the distance scale of $\Delta/2^i$ (where Δ is the maximal distance in the metric) and yet sufficiently sparse for the purposes of load-balancing. Then level-i beacons declare themselves to other nodes via a special broadcast, so that each node finds out about the nearby level-i beacons, and also forms upper and lower bounds on distances to these beacons. With some more fine-tuning, these bounds give rise to a triangulation with good guarantees for all node pairs. We conjecture that this approach can lead to an embedding with a similar property.

Meridian: a framework for location-aware node selection. In Chapter 6 we address similar issues in the context of a systems project on location-aware node selection in a large-scale distributed network. Specifically, we discuss our work on *Meridian* [WSS05], a framework for performing node selection based on network location. *Meridian* is a lightweight, scalable, and accurate system for keeping track of location information for participating nodes that does not require computing a network embedding as in [NZ02, DCKM04]; in effect, here we explore an approach which is *alternative* to the work on network embeddings. Our system is simple, loosely-structured, and entails modest resources for maintenance. It can efficiently find the closest node to a target, the latency minimizing node to a given set of nodes, and the set of nodes that lie in a region defined by latency constraints, which are frequently encountered building block operations in many location-sensitive distributed systems. Although less general than virtual coordinates, *Meridian* incurs significantly less error.

In this thesis we outline the system, and then proceed with the analysis that shows that Meridian provides robust performance, high scalability and good load balancing. This analysis focuses on doubling and growth-constrained metrics, and uses some of the techniques from our work on triangulation. For a large body of experimental results and further work on this system see the original paper [WSS05].

Node labeling problems. In Chapter 7, we interpret triangulation as a problem of assigning short informative labels to nodes, and provide near-optimal constructions for doubling metrics. We extend our techniques to other *node labeling* problems, where one needs to assign short labels to nodes of a graph so that they capture some (problem-specific) global information about distances and routes in the graph. Specifically, we consider three types of node-labeling problems: low-stretch routing schemes [PU89], distance labeling [GPPR04], and searchable small worlds [Kle00b]. We focus on weighted graphs that induce a doubling metric. The concrete problems, specific results and relevant background are discussed further in Section 7.1.

We approach triangulation and the three node-labeling problems mentioned above with a common technique: each construction is based on a version of *rings of neighbors*, a sparse distributed data structure described earlier. Recall that in *rings of neighbors*, the i-ring neighbors of a given node u lie in a ball B_i

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

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

In a more abstract view, a collection of rings of neighbors is a tractable representation for the fine structure of the underlying graph. The idea of using a tractable structure-preserving representation as a unifying technique for various problems on graphs is not new; several representations have been suggested in the literature, e.g. [AGLP89, ABNLP90] for general graphs and [Tal04, MHP05] for doubling graphs. Rings of neighbors seems to be a particularly suitable representation for the types of problems that we consider here.

1.1.2 Related work

To understand the technical contents of this dissertation, a reader needs to be familiar with the basic concepts of metrics and metric embeddings, as well as with some tools from probability and graph theory; see Chapter 2 for self-contained background on those.

As discussed above, the questions we consider here differ from the bulk of algorithmic embedding research (as surveyed in [Ind01, Lin02, Mat02a, IM04]) because we are able to measure only a small subset of the distances, and we allow a notion of slack in the performance guarantee. Indeed the whole problem of triangulation, which seeks simply to reconstruct the distances, would not be of interest if we already had access to all distances. Allowing slack changes the kinds of performance guarantees one can achieve; for example, as mentioned above, doubling metrics become embeddable with constant distortion in Euclidean space once a small slack is allowed. At the same time, we find that techniques from the body of previous work on embedding, combined with our results on triangulation, are useful in designing algorithms under these new constraints.

Work on distance labeling [GPPR04] seeks to assign a short label to each node in a graph so that the distance between u and v can be (approximately) determined from their labels alone. This is of course analogous to our goals in triangulation. In the most closely related work in this vein, Talwar investigated distance labels for doubling metrics [Tal04]. Both the objective and the techniques in [Tal04] differ considerably from our work on network triangulation here, however: in [Tal04], the concern is with labels of low bit complexity, but the encoding of distances into short labels there makes extensive use of the full distance matrix, and it is thus not adaptable to our setting in which distances to only a few beacons can be measured. The more extensive use of the distance matrix in [Tal04] comes in pursuit of a stricter goal: distance labels in which there is no notion of slack in the performance guarantee. We also consider this type of problems (in Chapter 7), and in particular improve over the result of [Tal04].

Work on property testing [GGR98] makes use of a somewhat different notion of slack in its performance guarantees: can an ϵ-fraction of the input be changed so that a given property holds? There has been some research on property testing in metric spaces (see e.g. [KS03, PR03], and related work on sampling for approximating metric properties in [Ind99]), but this work has considered problems quite different from what study here, and makes use of different sampling models and objective functions. Metric Ramsey theory [BLMN05] also seeks subsets of a metric satisfying specific properties, but it tends to operate in a qualitatively different part of the parameter space, exploring properties that hold on the sub-metric induced by relatively small subsets of the nodes, rather than properties that hold on a large fraction of the edges.

Finally, *distance geometry* [CH88] is a large area concerned with reconstructing point sets from sparse and imprecise distance measurements; our use of triangulation here corresponds to the notion of *triangle inequality bounds smoothing* in [CH88], but beyond this connection we are not aware of closely related work in the distance geometry literature.

1.2 Definitions and theorems: embeddings

Before we formally present our results, let us present some of the notions that will be used throughout the thesis. We will assume that the metric (V, d) is also represented as a graph on the nodes V, with the length of edge (u, v) being $d(u, v) = d_{uv}$. We imagine this graph as having n^2 edges, one for each pair $u, v \in V \times V$; this makes the exposition cleaner and does not change the results in any significant way. For a map $\varphi : V \to V'$ let us define the notion of the *distortion of a set S of edges* under embedding φ as the smallest $D \geq 1$ such that for some positive constant K and all edges $(u, v) \in S$ we have

$$d(u, v) \leq d'(\varphi(u), \varphi(v))/K \leq D \cdot d(u, v).$$

Note that the distortion of φ is the same as the distortion of the set of all edges.

Definition (ϵ-slack distortion). Given ϵ, an embedding $\varphi : V \to V'$ has *distortion D with ϵ-slack* if a set of all but an ϵ-fraction of edges has distortion at most D under φ.

We will also consider a stronger notion of slack, for which we need the following definition. Let $r_u(\epsilon)$ be the radius of the smallest ball around u that contains at least ϵn nodes. Call an edge uv *ϵ-long* if $d(u, v) \geq \min(r_u(\epsilon), r_v(\epsilon))$. Note that there are at least $(1 - \epsilon)\, n^2$ edges that are ϵ-long. For any such edge (u, v), at least one endpoint u is at least as far from the other endpoint v as the (ϵn)-th closest neighbor of v.

Definition (ϵ-uniform slack distortion). Given ϵ, an embedding $\varphi : V \to V'$ has *distortion D with ϵ-uniform slack* if the set of all ϵ-long edges has distortion at most D.

While the above notions of embeddings with slack allow the map φ to depend on the slack ϵ, the following notion asks for a single map that is *good for all ϵ simultaneously*.

Definition (gracefully degrading distortion). An embedding $\psi : V \to V'$ has a *gracefully degrading distortion $D(e)$* if for each $\epsilon > 0$, the distortion of the set of all ϵ-long edges is at most $D(\epsilon)$.

We now make precise the main results described above, and also describe some further results in the thesis. Our first result shows that if we are allowed constant slack, we can embed any metric into constant dimensions with constant distortion:

Theorem 1.1. *For any source metric (V, d), any target metric ℓ_p, $p \geq 1$ and any parameter $\epsilon > 0$, we give the following two $O(\log \frac{1}{\epsilon})$-distortion embeddings:*
(a) with ϵ-slack into $O(\log^2 \frac{1}{\epsilon})$ dimensions, and
(b) with ϵ-uniform slack into $O(\log n \log \frac{1}{\epsilon})$ dimensions.
Both embeddings can be computed with high probability by randomized beacon-based algorithms.

These results extend Bourgain's theorem on embedding arbitrary metrics into ℓ_p, $p \geq 1$ with distortion $O(\log^2 n)$ [Bou85], and are proved in a similar manner.

Note that the bounds on both the distortion as well as the dimension in part (a) of the above theorem are independent of the number of nodes n, which suggests that they could be extended to infinite metrics; this is further discussed in Section 3.3.3. In part (b), the dimension is proportional to $\log n$; we show that, for arbitrary metrics, this dependence on n is indeed inevitable. Let us mention that doubling metrics do not need such a dependence on n: in Section 3.3.2, these metrics are embedded into any ℓ_p, $p \geq 1$ with ϵ-uniform slack, distortion $O(\log \frac{1}{\epsilon} \log\log \frac{1}{\epsilon})$ and dimension $(\log \frac{1}{\epsilon})^{O(\log \frac{1}{\epsilon})}$.

We then study *embeddings into trees*. We extend the known results of probabilistic embedding into trees [Bar96, Bar98, FRT04] to obtain embeddings with slack. In particular, we use the technique of Fakcharoenphol et al. [FRT04] to obtain the following two results:

Theorem 1.2. *For any input metric (V, d) and any parameter $\epsilon > 0$ there exists an embedding into a tree metric with ϵ-uniform slack and distortion $O(\frac{1}{\epsilon} \log \frac{1}{\epsilon})$.*

In fact, the tree metric in Theorem 1.2 is induced by a *Hierarchically Separated Tree* (*HST*) [Bar96], which is a rooted tree with edge-weights w_e such that $w_e < w_{e'}/2$ whenever edge e' is on the path from the root to edge e.

Theorem 1.3. *For any input metric (V, d), the randomized embedding of [FRT04] into tree metrics has expected gracefully degrading distortion $D(\epsilon) = O(\log \frac{1}{\epsilon})$. It follows that we can embed any metric into ℓ_1 with gracefully degrading distortion $D(\epsilon) = O(\log \frac{1}{\epsilon})$.*

However, the dimension of the above embedding into L_1 may be prohibitively large. To overcome this hurdle, and to extend this embedding to ℓ_p, $p > 1$, we explore a different approach:

Theorem 1.4. *Consider a metric (V, d) which admits β-padded decompositions. Then it can be embedded into ℓ_p, $p \geq 1$ with $O(\log^2 n)$ dimensions and gracefully degrading distortion $D(\epsilon) = O(\beta)(\log \frac{1}{\epsilon})^{1/p}$.*

For the reader unfamiliar with padded decompositions, let us mention that doubling metrics and metrics induced by planar graphs have $\beta = O(1)$ (refer to Section 2.3 for more background); hence Theorem 1.4 implies that such metrics admit embeddings into ℓ_p, $p \geq 1$ with gracefully degrading distortion $O(\log \frac{1}{\epsilon})^{1/p}$. Note that for $p > 1$ this result can be seen as a strengthening of Theorem 1.1(b) on embeddings with ϵ-uniform slack.

Finally, we prove *lower bounds* on embeddings with slack: we give a very general theorem that allows us to convert lower bounds on distortion and dimension of embeddings that depend only on n into lower bounds in terms of the slack parameter ϵ.

Theorem 1.5. *Suppose for each k there exists a k-node metric H_k such that any (probabilistic) embedding of H_k into trees has distortion at least $D(k)$. Then for an arbitrarily small positive ϵ there exist finite metrics M, M^* on arbitrarily large number of nodes such that:*

(a) any (probabilistic) embedding of M into trees has ϵ-slack distortion $\Omega(D(\frac{1}{3\sqrt{\epsilon}}))$.

(b) any (probabilistic) embedding of M^* into trees has ϵ-uniform slack distortion $\Omega(D(\frac{1}{3\epsilon}))$.

Moreover, if metrics $\{H_k\}$ are planar (resp. K_r-minor-free, doubling, ℓ_p^d) then so are M and M^*.

A very similar result applies to (probabilistic) embeddings into trees. These two results allow us to prove a number of lower bounds; some of them are summarized in Table 1.1 on page 9. In particular, we obtain matching or nearly matching lower bounds for all our results on ϵ-slack embeddings.

1.3 Definitions and theorems: distributed algorithms

Let us state our results on distributed triangulation and embedding. We start with our main result on beacon-based triangulation:

Type of Embedding	Our Lower Bound	Original Example
All metrics into ℓ_p, $p \geq 1$	$\Omega(\frac{1}{p})(\log\frac{1}{\epsilon})$	Constant-degree expanders [Mat97]
$\mathcal{F}$ into ℓ_p, $p \in (1,2]$	$\Omega(1-p)\sqrt{\log 1/\epsilon}$	Laakso fractal [LMN04]
Growth-constrained ℓ_1-metrics into ℓ_1^d	$\Omega(\sqrt{\log_d 1/\epsilon})$	Laakso fractal [LMN04]
$\mathcal{F}$ into distributions of dominating trees	$\Omega(\log\frac{1}{\epsilon})$	$n \times n$ grid [AKPW95]
All metrics into tree metrics	$\Omega(1/\sqrt{\epsilon})$	n-cycle [RR98, Gup01]
ℓ_2^{2m+1} into ℓ_2^{2m}	$\Omega(1/\sqrt{\epsilon})^{1/m}$	[Mat90]

Here $\mathcal{F}$ is the family of doubling metrics that are shortest-paths metrics of planar graphs.
In the last two lines, bounds for ϵ-uniform slack can be obtained by replacing $\sqrt{\epsilon}$ by ϵ.

Table 1.1: Embeddings with slack ϵ: lower bounds on distortion

Theorem 1.6. *In any doubling metric M, a constant number of randomly selected beacons achieves an (ϵ,δ)-triangulation with probability $1-\gamma$, where the constant depends on δ, ϵ, and γ. Moreover, for any metric a constant number of randomly selected beacons achieves an upper bound estimate $D^+_{uv} \leq 3d_{uv}$ for all but an ϵ-fraction of pairs (u,v) with probability at least $1-\gamma$, where the constant depends on ϵ and γ.*

To formulate our result on perfect triangulation, we use the following notion of a *dense point set* as a generalization of the d-dimensional lattice: We say that a finite subset of $\mathbb{R}^d$ under the ℓ_1 metric is *dense* if the coordinates of all points lie in the interval $[0, kn^{1/d}]$ for a constant k, and the minimum distance between each pair of points is 1. (We will refer to k as the *density parameter*.)

Theorem 1.7. *In any dense point set M under the L_1 metric, a constant number of randomly selected beacons achieves a perfect triangulation with ϵ slack and with probability $1-\gamma$, where the constant depends on ϵ, γ, the dimension, and the density parameter.*

In addition to the result on beacon-based embeddings from Section 1.2, we provide a result that follows more closely the framework from the *GNP* algorithm [NZ02]. Let us say that a beacon-based algorithm is *GNP-type* if it conforms to the following framework: beacons are embedded first (by inspecting only the distances between the beacons), and then the coordinates of every non-beacon node u are computed separately by some black-box procedure that inspects the distances from u to the beacons and minimizes distortion on these distances.

Theorem 1.8. *For any doubling metric and any slack parameter $\epsilon > 0$ there exists a GNP-type algorithm that uses $k = O(1/\epsilon)^{O(\log\log 1/\epsilon)}$ beacons and computes an embedding into ℓ_p, $p \geq 1$ with dimension $O(k \log k)$ and ϵ-slack distortion $O(\log k)$.*

To state our further results on triangulation, we need to provide a more general definition thereof. A *triangulation* of order k is a labeling of nodes such that a label of a given node u consists of upper and lower bounds on distances from u to each node in a set S_u of at most k other nodes; for each $b \in S_u$ we denote these bounds by D^+_{ub} and D^-_{ub}. Then any two nodes (u,v) can exchange their labels and use the triangle inequality to upper-bound the (u,v)-distance by $D^+_{uv} = \min(D^+_{ub} + D^+_{vb})$, and lower-bound it by $D^-_{uv} = \max(D^-_{ub} - D^+_{vb}, D^-_{vb} - D^+_{ub})$, where the $\max$ and $\min$ are taken over all $b \in S_u \cap S_v$. An (ϵ,δ)-*triangulation* is a triangulation such that $D^+_{uv} \leq (1+\delta)\, D^-_{uv}$ for all but an ϵ-fraction of node pairs (u,v). Note that either bound can be seen as a $(1+\delta)$-approximate estimate on the (u,v)-distance, and, moreover, these bounds provide a ”quality certificate” for the estimate.

The following is our main result on fully distributed triangulation and embedding:

Theorem 1.9. *Let M be a doubling metric, and suppose that every node has $k = (2 \log n)^{\Omega(1)}$ neighbors chosen independently and uniformly at random. Then for any ϵ and δ that are each at least $(\log n)^{-O(1)}$ there exist a fully distributed algorithm that with high probability constructs an (ϵ, δ)-triangulation and an $O(k \log k)$-dimensional embedding into ℓ_p, $p \geq 1$ which has distortion $O(\log k)$ with ϵ-uniform slack. In this algorithm the per-node load and the total completion time are poly-logarithmic in n.*

For growth-constrained metrics we can obtain a triangulation with guarantees for all node pairs:

Theorem 1.10. *Consider a growth-constrained metric with polynomially bounded aspect ratio. Suppose each node has links to 3 nodes sampled independently at random in the network. Then for any $\delta > 0$ there exists a fully distributed algorithm that computes a $(0, \delta)$-triangulation of degree $(1/\delta)^{O(1)} (\log^2 n)$. The running time and per-node load is $(1/\delta)^{O(1)} (\log^7 n)$.*

In a centralized setting, we can extend this result to doubling metrics:

Theorem 1.11. *For any $\delta \in (0, 1)$ any doubling metric has a $(0, \delta)$-triangulation of order $(1/\delta)^{O(1)} (\log n)$. Moreover, such triangulation can be efficiently computed.*

1.4 Bibliographic notes

This dissertation is based on a line of work that has been started by [KSW04] and includes three other theoretical papers [Sli05b, CDG$^+$05, Sli05a], an unpublished manuscript [Sli06], and a collaboration on a related systems project [WSS05]. All above papers have appeared in conferences; their journal versions are forthcoming (that of [Sli05a] has been accepted to *Distributed Computing* and is currently under revision, and the other papers will be submitted to journals in the very near future).

The results in [CDG$^+$05] have been obtained independently by I. Abraham, Y. Bartal and O. Neiman, which lead to a merged conference publication [ABC$^+$05]. The results on lower bounds and on embeddings into trees (respectively, Section 3.6 and Section 4.8 in this thesis) were proved similarly by both groups. For the rest of the results in this paper, the techniques are quite different. The two groups of authors has agreed to write up the full versions of their results separately.

Chapters in this thesis correspond to publications as follows. Chapter 3 is mainly based on [KSW04] but includes portions of [Sli05b] and [CDG$^+$05]; this chapter has given me an opportunity to better organize the results on network triangulation and embedding from the above three papers. Chapter 4 is based entirely on [CDG$^+$05]. Chapter 5 represents a very recent work [Sli06] that has not yet been published. Chapter 6 is adapted from [WSS05], and Chapter 7 is from [Sli05a].

Chapter 2

Background and preliminaries

In order to understand the technical contents of this dissertation, a reader needs to be familiar with the basic concepts of metrics and metric embeddings, as well as with some tools from probability and graph theory; in this chapter we give a self-contained background on those. We include proofs whenever they are sufficiently short; we provide references for more complicated results.

Let us start with some notation that will be used throughout the thesis. Unless specified otherwise, we denote the underlying metric by (V, d), so that $d(u, v)$ denotes the distance between nodes u and v; we also use d_{uv} whenever typographically convenient. Let $B_u(r)$ be the closed ball of radius r around node u, i.e. $B_u(r) = \{v \in V : d_{uv} \leq r\}$. Let $r_u(\epsilon)$ be the radius of the smallest closed ball around u that contains at least ϵn nodes. The *open ball* of radius r around node u is the set of all nodes within distance strictly less than r from u. The term *ball* in a metric refers to a *closed* ball unless specified otherwise.

For $k \in \mathbb{N}$ define $[k]$ as the set $\{0, 1 \ldots k-1\}$. Throughout the paper, n denotes the number of nodes in the input graph or metric, and Δ denotes the *aspect ratio*, which is the largest distance divided by the smallest distance.

2.1 Expander graphs and Probability

Throughout the thesis we use Chernoff Bounds, a standard result which says that the sum of bounded independent random variables is close to its expectation with high probability (e.g. see the textbook of Motwani and Raghavan [MR95] for the proof).

Theorem 2.1 (Chernoff Bounds). *Consider the sum X of n independent random variables $X_i \in [0, y]$.*

(a) for any $\mu \leq E(X)$ and any $\epsilon \in (0, 1)$ we have $\Pr[X < (1-\epsilon)\mu] \leq exp(-\epsilon^2 \mu / 2y)$.

(b) for any $\mu \geq E(X)$ and any $\beta \geq 1$ we have $\Pr[X > \beta\mu] \leq \left[\frac{1}{e}(e/\beta)^\beta\right]^{\mu/y}$.

For an undirected graph, the *expansion* is defined as $\min \frac{|\partial(S)|}{|S|}$, where the minimum is over all nonempty sets S of at most $n/2$ vertices, and $\partial(S)$ stands for the set of edges with exactly one endpoint in S. We can generalize this definition to *weighted* undirected graphs, or, equivalently, to symmetric non-negative matrices: we just define $\partial(S)$ to be the total weight of all edges with exactly one endpoint in S. We can further extend this definition to directed graphs (non-symmetric matrices) by considering the weight of all edges leaving S.

For a pre-defined absolute constant, an *expander* is an undirected graph whose expansion is at least this constant. Expanders are well-studied and have rich applications, see [LW02, AS00, MR95, Xia03] for more background. We will use the following two standard results:

Theorem 2.2 (Folklore). *An undirected graph of degree d and expansion γ has diameter at most $\frac{2d}{\gamma}\log n$.*

Proof. Let $x = 1 + \frac{\gamma}{d}$. Fix any node u. We claim that $|B_u(k)| \geq x^k$ for any integer k such that the ball $B_u(k-1)$ contains at most $n/2$ nodes. Indeed, let us use induction on k: suppose that $x^k \leq |B_u(k)| \leq n/2$ for some integer k. Then by definition of expansion there are at least $\gamma|B_u(k)|$ edges with exactly one end in $B_u(k)$. Since the degree of the graph is d, these edges go to at least $\frac{\gamma}{d}\,|B_u(k)|$ distinct nodes outside of $B_u(k)$. It follows that the ball $B_u(k+1)|$ contains at least $x\,|B_u(k)| \geq x^{k+1}$ nodes, claim proved.

Note that $\gamma \leq d$. This is because for any set $S \subset V$ there are at most $d|S|$ edges with at least one end in S. It follows that for $y = d/\gamma$ we have $(1+1/y)^y \geq 2$, so $x^k \geq 2^{k/y}$. Therefore for $k = y\log n$ the ball $B_u(k)$ contains at least $\frac{n}{2}+1$ nodes. So for any two nodes u, v the balls $B_u(k)$ and $B_v(k)$ overlap, hence the two nodes are connected by a path of length at most $2k$. □

Theorem 2.3 (Folklore). *Fix node set V. Suppose for each node u we choose three nodes independently and uniformly at random from V, and create undirected links between u and these three nodes. Then the resulting graph is an expander with high probability.*

See e.g. page 10 of [GMS04] for the proof. We will actually need a slightly stronger version where we select nodes from (and construct an expander on) any given subset Q of nodes, whereas we need the failure probability to be low in terms of n, not the size of Q. Hence we create $O(\log n)$ links per node instead of just three.

Theorem 2.4. *Fix node set V of n nodes, and a subset $Q \subset V$. Suppose for each node $u \in Q$ we choose at least $3\log n$ nodes independently from a near-uniform distribution on Q, and create undirected links between u and these nodes. Then the induced graph on Q is an expander with high probability.*

In Theorem 2.3 and Theorem 2.4 expanders have degree at most $O(\log n)$. We note in passing that for many applications it is useful to have constant-degree expanders. Indeed, such graphs exist; for instance, for large enough d a random d-regular graph is an expander with high probability [Fri03].

A graph (V, E) induces a Markov chain on V as follows: for any edge $(u, v) \in E$, the transition probability $u \to v$ is set as $1/\deg(u)$. In particular, undirected graphs with low degree and high expansion gives rise to a Markov chains whose transition matrix has high expansion.

The following seminal result connects the mixing time of a Markov chain with the expansion of its transition matrix; we state it in a somewhat simplified form which is suitable for the purposes of this chapter.

Theorem 2.5 (Rapid mixing, Sinclair and Jerrum [SJ89]). *Consider an ergodic time-reversible n-state Markov chain with a uniform stationary distribution. Suppose that for every node the probability of stalling is at least $\frac{1}{2}$. Let γ be the expansion of the transition matrix. Then for any $k \geq O(\gamma^{-2})(\log n)$ and any initial distribution the k-step distribution of this Markov chain is near-uniform.*

The phenomenon when an n-state Markov chain achieves a near-stationary distribution in $O(\log n)$ steps is known as *rapid mixing*. In fact, the original formulation of the above theorem extends to arbitrary stationary distributions.

2.2 Metric embeddings

Metric space. A metric space is a pair (V, d), where V is a set of nodes, and d is a *metric*, i.e. a symmetric non-negative mapping $V \times V \to \mathbb{R}$ which satisfies the *triangle inequality*:

$$d(x,y) \leq d(x,z) + d(y,z) \text{ for any three nodes } x,y,z \in V. \qquad (2.1)$$

Let us define the distance from node u to subset $S \subset V$ as $d(u,S) = \inf_{v\in S} d(u,v)$. Let us rewrite the triangle inequality as $d(x,y) \geq |d(x,z) - d(y,z)|$. In this formulation, we can replace $d(\cdot, z)$ by the distance to an arbitrary set:

Lemma 2.6. *In any metric, for any two nodes* (x,y) *and any set* S *we have* $d(x,y) \geq |d(x,S) - d(y,S)|$.

Proof. We will only use this lemma for finite S. In this case we can pick node $u \in S$ such that $d(x,S) = d(x,u)$, and node $v \in S$ such that $d(y,S) = d(y,v)$. Then

$$d(x,y) \geq d(u,y) - d(u,x) \geq d(v,y) - d(u,x) = d(y,S) - d(x,S),$$

and similarly $d(x,y) \geq d(x,S) - d(y,S)$.

For the sake of completeness, let us consider the case of infinite S. Then there exists a sequence $\{u_i\}$ of nodes such that $d(x,u_i) \to d(x,S)$ and a sequence $\{v_i\}$ of nodes such that $d(y,v_i) \to d(y,S)$. In the latter sequence, let us choose a subsequence $\{v_i^*\}$ such that $d(y,u_i) \geq d(y,v_i^*)$. for each i. Then

$$d(x,y) \geq d(u_i,y) - d(u_i,x) \geq d(v_i^*,y) - d(u_i,x) \to d(y,S) - d(x,S),$$

and similarly $d(x,y) \geq d(x,S) - d(y,S)$. □

Metric embeddings. An *embedding* of a finite metric space (V,d) into a *target* metric space (V^*, d^*) is a map $\varphi : V \to V^*$. Ideally, such map would preserve distances exactly, although this is typically not possible; see a book by Deza and Laurent [DL97] for more background on isometric (exact distance preserving) embeddings.

Recent work on embeddings has used *distortion* as the fundamental measure of quality; the distortion of an embedding is the worst multiplicative factor by which distances are increased by the embedding. Formally, for an embedding $\varphi : V \to V^*$, the *distortion* is the smallest D so that for some constant C we have $d(x,y) \leq C\, d^*(\varphi(x), \varphi(y)) \leq D d(x,y)$ for all pairs $x,y \in V \times V$. Note that this definition is invariant under arbitrary scaling. The popularity of distortion has been driven by its applicability to approximation algorithms: informally, if the embedding $\varphi : V \to V^*$ has distortion D, then the cost of solutions to some optimization problems on (V,d) and on $(\varphi(V), d^*)$ can only differ by some function of D.

Typically, the goal of a metric embedding is to map a ”complicated” metric space into a ”simpler” one. The most popular target spaces in the literature are ℓ_p spaces and distributions over tree metrics. We will mainly focus on the former, as it is more relevant to this dissertation.

ℓ_p spaces. For any $p \geq 1$, let d_p be the metric induced by the p-norm: for any $d \leq \infty$ and $x,y \in \mathbb{R}^d$,

$$d_p(x,y) := \|x-y\|_p := \left(\sum_{i=1}^{d} (x_i - y_i)^p \right)^{1/p}.$$

For any $d \in \mathbb{N}$, a d-dimensional ℓ_p space (denoted ℓ_p^d) is a metric space $(\mathbb{R}^d, d_p)$. An infinite-dimensional ℓ_p space (just denoted ℓ_p) is a metric space (V, d_p), where V is the set of all vectors in $\mathbb{R}^\infty$ that have a finite p-norm. Note that every finite-dimensional ℓ_p space can be cast as a sub-space of ℓ_p. For applications, the most useful ℓ_p spaces are ℓ_1 and ℓ_2; note that ℓ_2 is the familiar Euclidean space.

2.2.1 Relations between different ℓ_p norms

Let us give some background on how different p-norms relate to one another. In many cases, one can switch between different p-norms, or reduce the dimension, incurring arbitrarily low distortion; typically in such results ℓ_2 is the easiest space to embed from, and ℓ_1 is the easiest space to embed into. We state all such results as two theorems. In the first theorem, the source space is infinite, yet the target dimension is finite:

Theorem 2.7. *Let $p, q \in [1, \infty)$. An ℓ_p^d space can be embedded into $\ell_q^{d^*}$ with distortion $1+\epsilon$ if:*

(a) [BS82] $q < p \leq 2$ and $d^* = c(\epsilon)\, d$,

(b) [Dvo59] $p = 2$ and $d^* = 2^{O(d/\epsilon^2)}$.

The second theorem shows that a finite ℓ_2 metric can be approximated, to a fixed arbitrary precision, in ℓ_1 and ℓ_2 using only $O(\log n)$ dimensions [FLM77, JL84]. However, a similar dimension reduction is impossible for ℓ_1 metrics [BC03, LN04].

Theorem 2.8 (Dimension reduction). *Any n-node subset of ℓ_2 can be embedded into ℓ_q, $q \in \{1, 2\}$ with distortion $1+\epsilon$ and dimension $O(\frac{1}{\epsilon}\log n)$. However, there exist arbitrarily large n-point subsets of ℓ_1 for which any embedding into ℓ_1 with distortion D requires $n^{\Omega(1/D^2)}$dimensions.*

2.2.2 Embeddings of finite metrics into ℓ_p spaces

The following seminal result is due to Bourgain [Bou85] and Linial et al. [LLR95].[1]

Theorem 2.9 (Bourgain's embedding). *Any n-node metric can be embedded into ℓ_p, $p \geq 1$ space with dimension $O(\log^2 n)$ and distortion $O(\log n)$. Moreover, such embedding can be efficiently computed.*

Bourgain's embedding technique is essential to this thesis. Accordingly, we will give a complete proof of the above theorem. We start with a sampling lemma which is implicitly used in [LLR95], but neither proved nor explicitly stated. We state and prove it here for the sake of completeness.

Lemma 2.10. *Consider disjoint events E and E' such that $\Pr[E] \geq \gamma$ and $\Pr[E'] \leq 2\gamma$. Let S be a set of $1/\gamma$ points sampled independently from this probability distribution. Then S hits E and misses E' with at least a constant probability $c = (e-1)\, e^{-1.5}$.*

Proof. Let $p = \Pr[E]$ and $p' = \Pr[E']$. Treat sampling a given point as two independent random events: first it misses E' with probability $1-p'$, and then (if it indeed misses) it hits E with probability $\frac{p}{1-p'}$. Without loss of generality let us rearrange the order of events: first for each point we choose whether it misses E', so that

$$\Pr[\text{all points miss } E'] = (1-p')^{1/\gamma} \approx e^{-p'/\gamma} \geq e^{-1/2}.$$

Then upon success choose whether each point hits E. Then at least one point hits E with probability at least $1-(1-p)^{1/\gamma} \geq 1-e^{-1}$. So the total success probability is at least $c = (1-e^{-1})\, e^{-1/2}$. □

Proof of Theorem 2.9: Let us fix $k = c\log n$, for a constant c to be determined later. For each $i \in [\log n]$ and $j \in [k]$, let S_{ij} be the set of $n/2^i$ nodes chosen independently and uniformly at random.

We define an embedding $f : V \to \ell_p$ with $k \log n$ dimensions, indexed by pairs (i, j) as above, so that for each node u and each set S_{ij} the corresponding coordinate is $f_{ij}(u) = k^{-1/p}\, d(u, S_{ij})$, where $d(u, S) = \min_{v\in S} d(u, v)$ is the distance between node u and set S.

[1] The original result in Bourgain [Bou85] was a (high-dimensional and non-algorithmic) embedding into ℓ_2; Linial et al. [LLR95] fine-tuned Bourgain's technique to yield the present formulation.

Let us fix a node pair (u,v). Let $d = d(u,v)$ be the original (u,v)-distance, and let $d^* = \|f(u)-f(v)\|_p$ be the embedded (u,v)-distance. For simplicity we will consider the case $p=1$ first. Let

$$x_{ij} = |d(u,S_{ij}) - d(v,S_{ij})|$$

be the contribution to d^* of the ij-th coordinate. By Lemma 2.6 this contribution is upper-bounded by d, so $d^* \leq O(d\log n)$. The hard part is the lower bound: $d^* \geq \Omega(d)$.

For each $i \in [\log n]$, let

$$\rho_i = \min(d/2,\, r_u(2^i/n),\, r_v(2^i/n)).$$

Note that the sequence $\{\rho_i\}$ is increasing with $\rho_0 = 0$ and $\rho_{i_0} = d/2$ for some i_0.

Claim 2.11. *For each i with high probability we have $\sum_{j\in[k]} x_{ij} \geq \Omega(k)(\rho_{i+1}-\rho_i)$.*

Proof. Fix i and $\gamma = 2^i/n$. Without loss of generality let us assume that the ball around u reaches size γn before the ball around v does: $\rho_i = r_u(\gamma) \leq r_v(\gamma)$. A given set S_{ij} contributes $x_{ij} \geq \rho_{i+1} - \rho_i$ as long as it has the following property: it hits the ball $B = B_u(\rho_i)$ and misses the open ball B^* of radius ρ_{i+1} around v. By Lemma 2.10 the probability of this happening is at least a positive constant c_0 (since the two balls are disjoint, $|B| \geq \gamma n$ and $|B^*| \leq 2\gamma n$). Thus the expected number of sets S_{ij} with this property is $c_0 k$, so applying the Chernoff bound, for big enough $k = O(1/c_0)(\log n)$ with high probability at least $c_0 k/2$ of sets S_{ij} have this property. □

Now with high probability the sum $\sum x_{ij}$ telescopes:

$$d^* = \tfrac{1}{k}\textstyle\sum_{ij} x_{ij} \geq \sum_i \Omega(\rho_{i+1}-\rho_i) \geq \Omega(\rho_{i_0}-\rho_0) = \Omega(d).$$

This completes the proof for the case $p=1$.

To extend the theorem to general $p \geq 1$, let d_p^* be the embedded (u,v)-distance and let $x = \log n$. Then

$$d_p^* = x^{1/p}\left(\frac{1}{xk}\sum_{ij} x_{ij}^p\right)^{1/p} \geq x^{1/p}\left(\frac{1}{xk}\sum_{ij} x_{ij}\right) = x^{1/p-1}\, d_{uv}^1 = x^{1/p-1}\,\Omega(d).$$

For a lower bound, recall that $x_{ij} \leq d$, so $d_p^* \leq \left(\frac{1}{k}\sum_{ij} d^p\right)^{1/p} = x^{1/p}\, d.$ □

In the above theorem, the distortion is optimal up to a constant factor [LLR95, Mat97]:

Theorem 2.12. *Let M be the shortest-paths metric of any constant-degree expander graph. Then for any $p \geq 1$, any embedding of M into ℓ_p space has distortion $\Omega(1/p)(\log n)$.*

Several results on embeddings into ℓ_p with sub-logarithmic distortion are known for restricted families of metrics. For instance, shortest-paths metrics of planar graphs can be embedded into ℓ_2 with distortion $O(\sqrt{\log n})$ [Rao99]; shortest-paths metrics of series-parallel graphs can be embedded into ℓ_1 with constant distortion [GNRS04].

For more background on metric embeddings and their algorithmic applications refer to a number of recent surveys [Ind01, Lin02, Mat02a, IM04].

2.2.3 Embeddings into tree metrics

A *tree metric* is a shortest-paths metric of a (positive-weighted) tree.

Lemma 2.13 (Folklore). *Tree metrics are isometrically embeddable into ℓ_1^{n-1}.*

Proof Sketch. Consider a tree $T = (V, E)$ with positive edge-weights w_e, $e \in E$. Let us number edges from 1 to $n-1$: let e_i be the i-th edge. Fix any node $r \in V$ as a root. We define the embedding into ℓ_1^{n-1} as follows: for each node $u \in V$, the i-th coordinate is equal to w_e if edge $e = e_i$ lies on the path from u to r, and 0 otherwise. □

Moreover, tree metrics are embeddable into ℓ_2 with distortion $O(\sqrt{\log\log n})$, and this is optimal up to constant factors [Bou86, Mat99]. Recall that by the dimension reduction result (Theorem 2.8) the dimension can be made as small as $O(\log n)$. For a *constant* target dimension, the best known result is an $O(n^{1/d})$-distortion embedding into ℓ_2^{d+1}, for any constant d [Gup00].

Tree metrics are very tractable algorithmically, but are not rich enough to accommodate even very simple metrics: e.g. a cycle on n nodes needs distortion $\Omega(n)$ to be embedded into trees metrics. However, the following approach has been very fruitful: embed into *distributions* over tree metrics, so that the original distance is approximated by the expected embedded distance; this is known as *probabilistic embedding*. For instance, for an n-node cycle consider a uniform distribution over the n line metrics produced by cutting some edge; then for any given edge the expected embedded distance is $2(1-\frac{1}{n})$. It is known that for any metric space on n nodes there exists an $O(\log n)$-distortion probabilistic embedding into trees. This result is a culmination of a line of work in [AKPW95, Bar96, Bar98, FRT04]; it is optimal up to constant factors, the counterexample being constant-degree expanders. Probabilistic embeddings into trees has led to numerous approximation algorithms, see e.g. [Ind01] for a survey.

2.3 Combinatorial notions of low dimensionality in metrics

In this section we describe three combinatorial notions of low dimensionality in metrics: grid dimension, doubling dimension, and decomposability parameter. These notions of low dimensionality induce three families of metrics that are instrumental to our results: respectively, growth-constrained metrics, doubling metrics, and decomposable metrics (growth-constrained metrics are metrics of constant grid dimension, etc.). These families of metrics are nesting, decomposable metrics being the most general.

2.3.1 Growth-constrained metrics

For n-dimensional grid and $\alpha = n+O(1)$, the following property holds: for any $x \geq 2$ the cardinality of any ball is at most x^α times smaller than the cardinality of a ball with the same center and x times the radius.[2] This motivates the following definition: the *grid dimension* of a metric is the infimum of all α such that the above property holds. Clearly, grid dimension of any n-node metric is at most $\log n$. *Growth-constrained metrics* are metrics of bounded (and, intuitively, low) grid dimension.

Growth-constrained metrics can be seen as generalized grids; they have been used as a reasonable abstraction of Internet latencies in the long line of work on DHTs started by Plaxton et al. [PRR99] (see the intro of [HKMR04] for a short survey). Growth-constrained metrics have also been considered in the theoretical computer science literature in the context of compact data structures [KR02], routing schemes [AM05], dimensionality in graphs [KL03], and gossiping protocols [KKD04].

We will use the grid dimension via the following simple corollary:

[2] In the literature this property is often defined for $x = 2$ only. This is essentially equivalent but slightly less convenient technically because in order to use this property one needs to round x up to the nearest power of two.

Lemma 2.14. *Suppose d_M is a metric with grid dimension α. Fix any two nodes u, v and let $d = d_M(u, v)$. Then for any positive r, r^* such that $\frac{d+r}{r^*} \geq 2$ we have $|B_u(r)| \leq (\frac{d+r}{r^*})^\alpha |B_v(r^*)|$.*

Proof. Since $B_u(r) \subset B_v(d+r)$, we have $|B_u(r)| \leq B_v(d+r) \leq |B_v(r^*)|\, (\frac{d+r}{r^*})^\alpha$. □

Grid dimension is a useful notion of low-dimensionality. However, it is not *robust*, in the sense that the dimension of a subset can be larger than the dimension of the entire metric. For a simple example, consider the set $[n]$ with a standard metric $d(x,y) = |x-y|$. The grid dimension of such set is 1, but for a subset $[n/2] \cup \{n-1\}$ the grid dimension is $\Omega(\log n)$.

2.3.2 Doubling metrics

Doubling metrics is a combinatorial (non-geometric) notion of low dimensionality that has recently become popular in the theoretical computer science literature [GKL03, KL04, KLMN05, Tal04, CGMZ05] in many different contexts, including metric embeddings, traveling salesman and compact data structures.

Any point set in a k-dimensional ℓ_p metric has the following property, called the *doubling property* [Ass83]: for some $\alpha = k + O(1) \in \mathbb{N}$ every set of diameter d can be covered by 2^α sets of diameter $d/2$. (The diameter of a set is the maximal distance between any two points in it.) This motivates the following definition: *doubling dimension* is the smallest α such that the above property holds. Clearly, doubling dimension of any n-node metric is at most $\log n$. Doubling metrics are defined as metrics of bounded (and, intuitively, low) doubling dimension.

By definition, doubling metrics generalize constant-dimensional ℓ_p metrics. Doubling metrics is a much wider class of metrics: in particular, there exist doubling metrics on n nodes that need distortion $\Omega(\sqrt{\log n})$ to embed into any ℓ_p, $p \geq 2$ [Sem96, Laa02, LP01, GKL03].[3] Moreover, doubling metrics subsume growth-constrained metrics:

Lemma 2.15. *For any metric, the doubling dimension is at most 4.1 times the grid dimension.*

Proof. Consider a metric with grid dimension α. Fix a set S of diameter d. We need to show that this set can be covered with $2^{8\alpha}$ sets of diameter $d/2$. Specifically, we fix some node $u \in S$, and cover the ball $B = B_u(d)$ with $2^{4.1\alpha}$ balls of radius $d/4$. We do it in a greedy fashion: while there exists a node in B which is not covered, pick any such node v and add a ball $B_v(d/4)$ to the cover.

Let $\mathcal{F}$ be the cover thus constructed; let C be the set of centers of balls in $\mathcal{F}$. Note that the distance between any two nodes in C is at least $d/4$, so the balls with centers in C and radius $d/8$ are pairwise disjoint. Furthermore, each such ball lies inside $B^* = B_u(1\frac{1}{8}\, d)$ and by Lemma 2.14 has cardinality at least $1/17^\alpha$ that of B^*. Therefore C contains at most $17^\alpha \leq 2^{4.1\,\alpha}$ nodes. □

However, doubling metrics is a much wider class of metrics: as an example of a doubling metric with high (super-constant) grid dimension, consider the set $\{1, 2, 4, \ldots, 2^n\}$ equipped with the standard distance function $d(x,y) = |x-y|$. Furthermore, unlike grid dimension, the doubling dimension is *robust*:

Lemma 2.16. *The doubling dimension of a subset is no larger than that of the entire metric.*

Proof. Let α be the doubling dimension of a metric on node set V, and let S be a subset. Then any subset $S' \subset S$ can be covered by 2^α subsets $S_1, S_2, S_3, \ldots \subset V$, each of diameter $d/2$. To obtain the desired covering by 2^α subsets of S, just intersect each of the S_i's with S. □

[3] In fact, the example in these papers – the shortest-paths metric of the *Laakso fractal* – is growth-constrained, so it yields the corresponding non-embeddability result for growth-constrained metrics.

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

Lemma 2.17. *In a metric of doubling dimension α, any set of diameter d can be covered by $2^{\alpha k}$ balls of radius $d/2^k$, for any integer $k \geq 1$. The desired cover can be efficiently constructed.*

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

In fact, for all our applications it suffices to redefine the doubling property in terms of covering a large ball with balls of half the radius. Moreover, it is slightly more convenient technically; in particular, the proof of Lemma 2.17 simplifies, and in Lemma 2.15 the constant factor is improved from 4.1 to 3.2. However, under this definition we no longer have the appealing robustness property (Lemma 2.16).

Note that the aspect ratio Δ can be arbitrarily large with respect to the number of nodes n and doubling dimension α. For instance, consider a 3-node metric space $\{1, 2, \Delta\}$, equipped with the natural distance function $d(x, y) = |x - y|$. However, it is easy to bound the aspect ratio from below:

Lemma 2.18. $1 + \log \Delta \geq \frac{1}{\alpha} \log n$, *for any metric with aspect ratio Δ and doubling dimension α.*

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

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

Theorem 2.19. *For any complete metric of doubling dimension α there exists a 2^α-doubling measure. If the metric is finite, such measure can be constructed efficiently, in time $O(2^{O(\alpha)}\, n \log n)$.*

The original existence result for finite metrics (and, in fact, for compact metrics) is due to [ALV75]. The proof has been simplified by [Wu98] and extended to complete metrics in [LS98]. The algorithmic result builds on the construction from [Wu98] and is due to [MHP05].

For $r > 0$ an *r-net* on a metric is a set S such that any point of the metric is at distance at most r from S, and any two points in S are at distance at least r. It is easy to see that for a finite metric such set exists and can be constructed greedily, starting from any (possibly empty) set of points that are at distance at least r from each other. It is often useful to consider r-nets in conjunction with doubling metrics, because of the following simple fact:

Lemma 2.20. *Any r-net has at most $(4r'/r)^\alpha$ elements in any ball of radius $r' \geq r$.*

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

For a more complete mathematical treatment of doubling metrics refer to the book by Heinonen [Hei01].

2.3.3 Decomposable metrics

Let us state the definition of a *padded decomposition* (see e.g. [GKL03, KLMN05]).

Definition 2.21. Given a finite metric space (V, d), a positive parameter $\Delta > 0$ and a mapping $\beta : V \to \mathbb{R}$, a Δ-bounded *β-padded decomposition* is a distribution Π over partitions of V such that the following conditions hold:

(a) For each partition P in the support of Π, the diameter of every cluster in P is at most Δ.
(b) If P is sampled from Π, then each ball $\mathbf{B}_x(\frac{\Delta}{\beta(x)})$ is partitioned by P with probability at most $\frac{1}{2}$.

Say that a metric *admits β-padded decompositions* (where β is a number called *decomposability parameter*) if for every $\Delta > 0$ it admits a Δ-bounded β-padded decomposition. It is known that any finite metric space admits $O(\log n)$-padded decomposition [Bar96]. Moreover, metrics of doubling dimension α admit $O(\alpha)$-padded decompositions [GKL03]. Furthermore, if a complete graph on r nodes is not a minor of a given graph G (e.g. if it has treewidth at most r) then the shortest-paths metric of G admits $O(r^2)$-padded decompositions [KPR93, Rao99, FT03].

For the sake of completeness, let us take a brief detour and overview graph minors and tree-width. A graph H is called a *minor* of an (undirected) graph G if H can be obtained from G by first deleting some vertices and edges, and then contracting some further edges. Graph minors have a rich theory, see an excellent book by Diestel [Die97] for more background. In particular, a seminal Kuratowsky theorem says that a graph can be drawn on a plane if and only if it does not have K_5 and $K_{3,3}$ as a minor; here K_r is a complete graph on r nodes, and $K_{a,b}$ is a complete bi-partite graph with a nodes in one part and b nodes in another. Moreover, for any given surface S there is an integer $r(S)$ such that if graph can be drawn on S then it cannot have $K_{r(S)}$ as a minor (and hence admits an $O(r^2(S))$-padded decomposition). This is a corollary of a deep result that for any given surface S there exists a finite list $\mathcal{L}$ of graphs such that any graph G can be drawn on S if and only if no graph in $\mathcal{L}$ is a minor of G. This result comes from a long line of work on graph minors, mainly due to Robertson and Seymour; see [Die97] for background, proof outline, and full bibliographical information.

It is known that a graph does not have K_r as a minor if it has tree-width at most r. Tree-width is a positive number which quantifies how close a graph is to being a tree; we omit the exact definition here. Tree-width is a major tool in the theory of graph minors. Besides, it has rich algorithmic applications: many problems admit much more efficient algorithms on (tree-like) graphs of low tree-width. See the books by Diestel [Die97] and Downey and Fellows [DF98] for more background. This is the end of our detour.

Let us define *decomposable metrics* as metrics that admit β-padded decomposition, for a constant β. Decomposable metrics are useful as a common generalization of doubling metrics and shortest-paths metrics of minor-excluding graphs. In particular, by [KLMN05] decomposable metrics can be embedded into ℓ_2 with distortion $O(\sqrt{\beta \log n})$, and more generally into ℓ_p, $p \geq 1$ with distortion $O(\beta^{1-1/p})\,(\log n)^{1/p}$.

In this thesis decomposable metrics appear in Chapter 4 on gracefully degrading distortion.

Chapter 3

Triangulation and Embedding using Small Sets of Beacons

In this chapter we discuss our results on network triangulation and embedding. This chapter is mainly based on [KSW04], the paper that started this line of work; it also includes relevant results from two follow-up papers, [Sli05b] and [ABC$^+$05]. Specifically, our main result on beacon-based embeddings (Section 3.2) and the lower bounds (Section 3.6) are from [ABC$^+$05, CDG$^+$05]. The last two subsections of Section 3.3, and Theorem 3.27 on fully distributed embeddings are from [Sli05b].

Our results in this chapter will generally involve showing that a large enough set of beacons sampled uniformly at random from the metric space will have a certain desired property. (For brevity, we will refer to such a sampled subset of the space as "a constant number of randomly selected beacons.") Because we will be working in many cases with constant-size samples, our properties will typically hold with a constant probability that can be made arbitrarily close to 1. Hence, in this context, we will sometimes use the phrase "with probability close to 1" as an informal short-hand for: with a probability that can be made arbitrarily close to 1 by increasing the sample size by a constant factor.

3.1 Beacon-based triangulation

We start by defining a notion of beacon-based distance estimation via triangle inequality.

Definition 3.1. Given a set S of beacons, we define lower and upper distance bounds for each pair (u, v) of points: $D^-_{uv} = \max_{b \in S} |d_{ub} - d_{vb}|$ and $D^+_{uv} = \min_{b \in S}(d_{ub} + d_{bv})$. We say that S achieves an (ϵ, δ)-*triangulation* if for all but an ϵ fraction of the pairs (u, v), we have $D^-_{uv} \leq (1+\delta) D^+_{uv}$.

As noted in the introduction, good triangulation bounds cannot be obtained for all metrics since, for example, non-trivial lower bound values D^-_{uv} cannot be achieved in the uniform metric in which all distances are 1. However, it is interesting to note that in every metric space, the upper bound D^+_{uv} actually does come within a constant factor of the true distance on all but an ϵ fraction of pairs.

Theorem 3.2. *If M is an arbitrary finite metric space, then a constant number of randomly selected beacons achieves an upper bound estimate $D^+_{uv} \leq 3 d_{uv}$ for all but an ϵ-fraction of pairs (u, v) with probability at least $1 - \gamma$, where the constant depends on ϵ and γ.*

Proof. Let B_u be the smallest ball around u containing at least $\epsilon n/2$ nodes. For each point u in M, and with enough beacons, at least one point in B_u will be selected as a beacon with probability close to 1. Suppose this happens, and let b be a beacon in B_u. Then all but at most $\epsilon n/2$ points v lie outside B_u or on its boundary; for any such point, we have $d_{vb} \le d_{ub} + d_{uv} \le 2d_{uv}$ and hence $D^+_{uv} \le d_{ub} + d_{vb} \le d_{uv} + 2d_{uv} = 3d_{uv}$. □

The upper bound of 3 in Theorem 3.2 is tight, as shown by the shortest-path metric of the complete bipartite graph $G = K_{n,n}$ with unit-distance edges. For all non-beacon pairs (u, v) on opposite sides of G, we have $D^+_{uv} = 3d_{uv}$. With a modification of this example, we can in fact show that no algorithm given access to each node's distances to all beacons can estimate d_{uv} to within a factor better than 3 for a large fraction of pairs (u, v). Specifically, we randomly generate a graph G' by deleting each edge from $G = K_{n,n}$ with probability $\frac{1}{2}$. If u and v are on opposite sides of G', then $d_{uv} = 1$ if the edge (u, v) is present, and otherwise $d_{uv} = 3$ with probability $1 - o(1)$. But if neither u nor v is a beacon, the full set of node-to-beacon distances gives no information about the presence or absence of the edge (u, v), and hence one cannot resolve whether this distance is 1 or 3.

For metrics of bounded doubling dimension, we have a much stronger result.

Theorem 3.3. *In any s-doubling metric M, a constant number of randomly selected beacons achieves an (ϵ, δ)-triangulation with probability $1 - \gamma$, where the constant depends on δ, ϵ, γ, and s.*

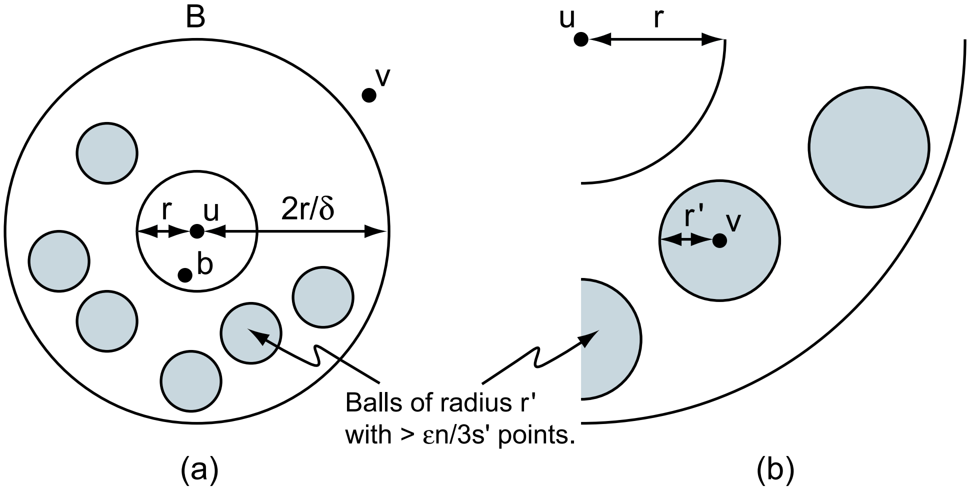

Figure 3.1: Triangulation in doubling metrics.

Proof. Fix any point u. Let $r = r_u(\epsilon/3)$, and consider a large ball $B = B_u(2r/\delta)$. By our definition of r, there are only a small number of points at distance strictly less than r from u, and we will ignore our estimated distances to these points. By selecting enough beacons, we can ensure that with probability close to 1 at least one beacon b lies in $B_u(r)$. Consider any point $v \notin B$. Since b is close to u and relatively very far from v, we can argue that the upper and lower bound provided by b on the distance from u to v will be good (see Figure 3.1a). In particular, if $d = d_{uv}$ then $d_{vb} + d_{ub} \le d + 2d_{ub} \le d + 2r = (1 + \delta)d$, and similarly $d_{vb} - d_{ub} \ge (1 - \delta)d$.

It remains to consider the possibly large set of points in the annulus $B - B_u(r)$. For these points, a beacon in $B_u(r)$ will not necessarily suffice to give the desired bound. Instead, we need to use the doubling property to show that the points in the annulus can be covered with a bounded number of very small balls, and with probability close to 1 we can ensure beacons lie in most of these. In other words, to estimate the distance d_{uv} for $v \in B - B_u(r)$, we will find a beacon close to v rather than close to u.

We would like to cover the annulus with balls of small radius $r' = \delta r/2$. By the doubling property, B (and hence $B - B_u(r)$) can be covered by $s' = (2/\delta)^{2 \log s}$ balls of radius r', as shown in Figure 3.1(b).

Disregarding balls containing fewer than $\epsilon n/3s'$ points throws out at most $\epsilon n/3$ points. Again, if we know that each of the remaining balls contains a beacon, then all points in these balls will have upper and lower bounds that are within a $1 \pm \delta$ factor of their respective distances to u.

Thus, we conclude by arguing that if we chose a sufficiently large (constant) number k of beacons, namely $k = O(s'/\epsilon)(\log \frac{1}{\epsilon})$, then with probability close to 1 a beacon will be selected in all but an $\epsilon/3$ fraction of balls containing $\epsilon n/3s'$ or more points. Combining these results shows that all but $\frac{1}{3}\epsilon n$ points have good estimated distances to all but $\frac{2}{3}\epsilon n$ points. This is the desired result. □

Remark. The above argument uses $O(\frac{1}{\epsilon}\log\frac{1}{\epsilon})\,(2/\delta)^{2\log s}$ beacons to obtain an (ϵ,δ)-triangulation with high probability. Note that a similar argument with $O(\frac{1}{\epsilon}\log n)\,(2/\delta)^{2\log s}$ beacons yields a *strong* (ϵ,δ)-triangulation. In Section 3.3 we obtain a strong (ϵ,δ)-triangulation using a number of beacons that depends only on s, ϵ and δ.

The following lemma is implicit in the proof of Theorem 3.3, and it will be very useful in our subsequent discussion of doubling metrics. To state the lemma, we introduce the following definitions. If E is a set of pairs of points in M, we say that E is an *ϵ-dense set* if it includes all but an ϵ fraction of all pairs, and we say that it is a *strong ϵ-dense set* if it includes all but an ϵ fraction of all pairs of the form (u,v) for each point u.

Lemma 3.4. *Consider an s-doubling metric (V,d), fix $\epsilon,\delta \in (0,1)$, and let $\epsilon_\delta = \frac{\epsilon}{2}\left(\frac{\delta}{2}\right)^{2\log s}$. Then for a strong ϵ-dense set of node pairs (u,v) we have $\min(r_u(\epsilon), r_v(\epsilon_\delta)) \le \delta d_{uv}$.*

Perfect triangulation As mentioned in the introduction, the stronger notion of *perfect triangulation* is sometimes achievable, when $D^-_{uv} = D^+_{uv} = d_{uv}$ for all but an ϵ-fraction of node pairs, using only a constant number of beacons. A natural example where this occurs is for the points of a finite d-dimensional lattice under the L_1 metric (this is a consequence of Theorem 3.5 below). It is natural to ask whether perfect triangulation is possible for all finite point sets in the L_1 metric, but this is too strong; consider for example the union of the points $\{(i, n-i) : i \in [n]\}$ and $\{-i, -(n-i) : i \in [n]\}$ in the plane.

As a way to understand how general this phenomenon is, we use the following notion of a *dense point set* as a generalization of the d-dimensional lattice: We say that a finite subset of $\Re^d$ under the L_1 metric is *dense* if the coordinates of all points lie in the interval $[0, kn^{1/d}]$ for a constant k, and the minimum distance between each pair of points is 1. (We will refer to k as the *density parameter*.)

Theorem 3.5. *In any dense point set M under the L_1 metric, a constant number of randomly selected beacons achieves a perfect triangulation with ϵ slack and with probability $1-\gamma$, where the constant depends on ϵ, γ, the dimension, and the density parameter.*

We start with a proof sketch and follow up with the full proof. For ease of exposition we assume that $d = 2$, but the same techniques extend naturally to any constant dimension.

Proof Sketch. Given a dense point set M in $[0, \sqrt{kn}]^2$, we divide M into square cells with width and height $\delta\sqrt{kn}$, for a small constant δ. We partition these cells into two types: *heavy* and *light*, where roughly speaking the heavy cells are those that contain at least $\Omega(\delta^2 n)$ points. We argue that with probability close to 1, each heavy cell will contain a beacon. Also, we can ignore errors on pairs that involve points in light cells, or that involve two points in the same heavy cell, since there are relatively few pairs like this. Thus, we only need to consider pairs of points that belong to distinct heavy cells.

We then argue that for most heavy cells C, there are heavy cells K_1, K_2, K_3, K_4 in each of the four "quadrants" of the square $[0, \sqrt{kn}]^2$ defined by treating C as the origin. This requires a geometric argument based on the density property; however, once the existence of K_1, K_2, K_3, K_4 is established, one beacon in each K_i is sufficient to provide a tight lower bound on any distance pair involving a point in C. Analogously, for the upper bound, we show by another application of the density property that for most pairs of heavy cells C and C', there is a heavy cell K in the rectangle with corners at C and C'; one beacon in K is sufficient to provide a tight upper bound on distances between points in C and C'. □

Proof of Theorem 3.5: Consider a dense point set M in $[0, \sqrt{kn}]^2$. Divide M into cells with width and height $\delta\sqrt{kn}$, for some δ to be chosen later. There will be $\frac{1}{\delta^2}$ cells. Let x_C and y_C denote the row and column of cell C. Define $h = \min(\delta^2 n/4k, \delta^2 n\epsilon/3)$, and call a cell C *heavy* if it contains at least h points, and *light* otherwise. The idea is that we will be able to ensure that with high probability, nearly all heavy cells will contain beacons, and that a negligible number of points fall outside of the heavy cells. We will then argue that for most pairs of points that lie in heavy cells, triangulation will give matching upper and lower bounds.

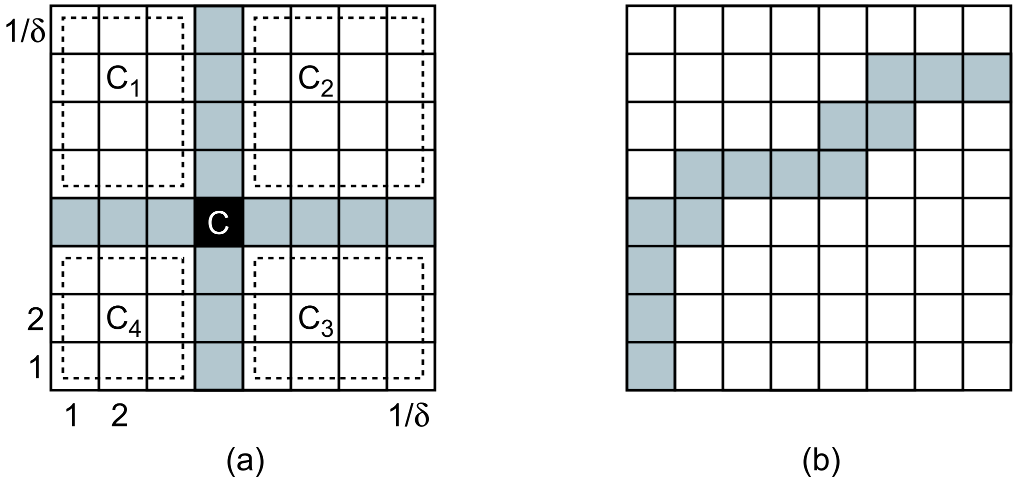

Figure 3.2: Dense point sets: (a) a cell C, $\mathcal{A}_C$ in gray, and corresp. quadrants; (b) a band of bad heavy cells.

Since no two points in M are within a distance of 1, no cell can have more than $4\delta^2 nk$ points. So if we let α be the fraction of cells that are heavy, then (omitting some easy arithmetic) $\alpha \geq 1/(4k+1)$.

We will begin by proving that the lower bound is correct for most pairs. Say two cells C, D are *aligned* if $x_C = x_D$ or $y_C = y_D$. Let $\mathcal{A}_C$ be the set of cells aligned with C. Note that the removal of $\mathcal{A}_C$ partitions the area into four quadrants, which we label C_1, C_2, C_3, and C_4, as shown in Figure 3.2(a). Say a dense cell C is *good* if each of its four quadrants contain at least one heavy cell, and *bad* otherwise. Observe that if C is good, and all dense cells contain beacons, then all points in C will have correct lower bounds to all points in $M - \mathcal{A}_C$.

We now need to show that most dense cells are good. Any dense cell that is not good can attribute its badness to one of its quadrants. Define $\mathcal{B}_i$ for $1 \leq i \leq 4$ to be the set of heavy cells lacking a heavy cell in their i^{th} quadrant. Consider cells $C, D \in \mathcal{B}_1$ and note that $x_C + y_C \neq x_D + y_D$, since otherwise one of these cells would be to the upper-left of the other, violating our assumption. Therefore $|\mathcal{B}_1| \leq \frac{2}{\delta}$ (see Figure 3.2(b) for a possible $\mathcal{B}_1$ set). The argument is symmetric for all four quadrants, so in total, there can be no more than $\frac{8}{\delta}$ bad cells. Since any cell contains at most $4\delta^2 nk$ points, the total number of points in bad cells is at most $32\delta nk$. Choosing $\delta = \frac{\epsilon}{96k}$ ensures that only $\frac{\epsilon}{3}n$ points are in bad cells.

By our definition of h, the total number of points that are in light cells is also at most $\frac{\epsilon}{3}n$. Lastly, for those points in any good cell C, we have no guarantee about the lower bound to points in $\mathcal{A}_C$. But this set contains $\frac{2}{\delta} - 2$ cells, and hence fewer than $\frac{\epsilon}{3}n$ points. Hence, by selecting a large enough number of beacons, we can ensure with high probability that all but an ϵ fraction of distances have correct lower bounds.

The same general idea works for the upper bound as well. The primary difference is we need the idea of a heavy cell D being bad *relative* to some cell C, meaning there are no heavy cells in the rectangular region bounded by C and D. It is this region that needs to contain a beacon for us to have a good upper bound on distances from C to D. As before, we can show that only a small number of cells are bad relative to any other cell, and for all other cells, the calculated upper bound will be correct. The same choice of δ used above gives the desired result.

□

3.2 Beacon-based embeddings

In this section we prove our main result on beacon-based embeddings. The result is two-fold. On one hand, we show that using a small number of beacons it is possible to embed an arbitrary metric into low-dimensional ℓ_p, $p \geq 1$ space with low distortion (and slack). On the other hand, for any $\epsilon > 0$ we achieve an embedding with ϵ-slack and distortion that depends only on ϵ, which is a novel result for traditional (non-distributed) embeddings.

Let us fix $\epsilon > 0$ and write $\rho_u = \rho_u(\epsilon)$. Recall that an edge (u, v) is *ϵ-long* if $d_{uv} \geq \min(\rho_u, \rho_v)$; call it *$\epsilon$-good* if $d_{uv} \geq 4 \min(\rho_u, \rho_v)$. We partition all the ϵ-long edges into two groups, namely those which are ϵ-good and those which are not, and use a separate embedding (i.e. a separate block of coordinates) to handle each of the groups. Specifically, we handle ϵ-good edges using a Bourgain-style embedding, and for the rest of the ϵ-long edges we use an auxiliary embedding such that for any edge (u, v), the embedded (u, v)-distance is $\Theta(\rho_u + \rho_v)$. The combined embedding has dimension $O(\log^2 \frac{1}{\epsilon})$ and achieves distortion $O(\log \frac{1}{\epsilon})$ on a set of all but an ϵ-fraction of edges.

There are several ways in which this result can be refined. First, we can ask for low *ϵ-uniform-slack* distortion, and require distortion $O(\log \frac{1}{\epsilon})$ on the set of all *ϵ-long* edges; we can indeed obtain this, but have to boost the number of dimensions to $O(\log n \log \frac{1}{\epsilon})$. As Theorem 3.10 shows, this increase is indeed required. We note that this logarithmic increase in the number of dimensions is not the case for doubling metrics: in Section 3.3.2 we embed doubling metrics into any ℓ_p, $p \geq 1$ with ϵ-uniform slack, distortion $O(\log \frac{1}{\epsilon} \log\log \frac{1}{\epsilon})$ and dimension $(\log \frac{1}{\epsilon})^{O(\log \frac{1}{\epsilon})}$

Second, this embedding can be computed in a distributed *beacon-based* framework. Here a small number of nodes are selected independently and uniformly at random, and designated as *beacons*. Then the coordinates of each node are computed as a (possibly randomized) function of its distances to the beacons.

Third, note that for the ϵ-slack result, the target dimension is independent of n, which suggests that this result can be extended to infinite metrics. We discuss this further in Section 3.3.3.

Theorem 3.6. *For any source metric (V, d), any target metric ℓ_p, $p \geq 1$ and any parameter $\epsilon > 0$, we give the following two $O(\log \frac{1}{\epsilon})$-distortion embeddings:*

(a) with ϵ-slack into $O(\log^2 \frac{1}{\epsilon})$ dimensions, and

(b) with ϵ-uniform slack into $O(\log n \log \frac{1}{\epsilon})$ dimensions.

These embeddings can be computed with high probability by randomized beacon-based algorithms that use, respectively, only $O(\frac{1}{\epsilon} \log \frac{1}{\epsilon})$ and $O(\frac{1}{\epsilon} \log n)$ beacons.

The proof will use a sampling lemma (Lemma 2.10 in Section 2.2) which is implicitly used in [LLR95] but neither proved nor explicitely stated.

Proof. Let $\delta > 0$ be the desired total failure probability. The embedding algorithm is essentially the same for both parts, with one difference: we let $k = O(\log \frac{1}{\delta} + \log \frac{1}{\epsilon})$ for part (a), and $k = O(\log \frac{1}{\delta} + \log n)$ for part (b). We describe a centralized algorithm first, and prove that it indeed constructs the desired embedding. Then we show how to make this algorithm beacon-based.

We use two blocks of coordinates, of size kt and k, respectively, where $t = \lceil \log \frac{1}{\epsilon} \rceil$. The first block comes from a Bourgain-style embedding without the smaller distance scales. For each $i \in [t]$ choose k independent random subsets of V of size 2^i each, call them S_{ij}, $j \in [k]$. The first-block coordinates of a given node u are

$$f_{ij}(u) = (kt)^{-1/p}\, d(u, S_{ij}), \text{ where } i \in [t],\, j \in [k].$$

For every node u and every $j \in [k]$, choose a number $\beta_{ui} \in \{-1, 1\}$ independently and uniformly at random. The second-block coordinates of u are $g_j(u) = k^{-1/p}\, \rho_u\, \beta_{uj}$, where $j \in [k]$. This completes the embedding.

For an edge uv, let $f(uv)$ and $g(uv)$ denote the ℓ_p-distance between u and v in the first and the second block of coordinates, respectively. By construction, $f(uv) \leq d_{uv}$ and $g(uv) \leq \rho_u + \rho_v$. Moreover:

Lemma 3.7. *For every ϵ-good edge uv, $f(uv) \geq \Omega(d_{uv}/t)$ with failure probability at most $t/2^{\Omega(k)}$.*

Let us prove this Lemma. Let us fix an ϵ-good edge uv and let $d = d_{uv}$. Let α_i be the minimum of the following three quantities: $\rho_u(2^{-i})$, $\rho_v(2^{-i})$ and $d/2$. The numbers α_i are non-increasing; $\alpha_0 = d/2$. Moreover, since edge uv is ϵ-good we have $\alpha_t \leq \min(\rho_u, \rho_v, d/2) \leq d/4$.

Claim 3.8. *For each i with failure probability at most $1/2^{\Omega(k)}$ we have the event*

$$\sum_{j \in [k]} |d(u, S_{ij}) - d(v, S_{ij})| \geq \Omega(k)(\alpha_i - \alpha_{i+1}) \tag{3.1}$$

Proof. We use a standard Bourgain-style argument. Let us fix i and let $\gamma = 2^i \epsilon$. Without loss of generality let us assume that the ball around u reaches size γn before the ball around v does: $\alpha_i = \rho_u(\gamma) \leq \rho_v(\gamma)$. A given set S_{ij} contributes at least $\frac{1}{k}(\alpha_{i+1} - \alpha_i)$ to d^*_{uv} as long as it has the following property: it hits $B = B_u(\alpha_i)$ and misses the open ball B' of radius α_{i+1} around v. By Lemma 2.10 the probability of this happening is at least a constant c (since the two balls are disjoint, $|B| \geq \gamma n$ and $|B'| \leq 2\gamma n$). Thus the expected number of S_{ij}'s with this property is ck, so by Chernoff bounds with failure probability at most $1/2^{\Omega(k)}$ it is the case that at least $ck/2$ of S_{ij}'s do have this property, thus ensuring (3.1). □

Therefore, with failure probability at most $t/2^{\Omega(k)}$, the event (3.1) happens for all $i \in [t]$ simultaneously, in which case

$$\sum_{i \in [t],\, j \in [k]} |d(u, S_{ij}) - d(v, S_{ij})| \geq \sum_{i \in [t]} \Omega(k)(\alpha_i - \alpha_{i+1}) = \Omega(k)(\alpha_0 - \alpha_t) \geq \Omega(kd), \tag{3.2}$$

which proves Lemma 3.7 for the case $p = 1$. We extend this to $p \geq 1$ using a standard inequality. Let $f_p(uv)$ be the value of $f(u, v)$ for a given $p \geq 1$. Let

$$x_{ij} = |d(u, S_{ij}) - d(v, S_{ij})|$$

be the contribution to $f(u, v)$ of the set S_{ij}. Then

$$f_p(uv) = \left(\frac{1}{tk} \sum_{ij} x_{ij}^p \right)^{1/p} \geq \left(\frac{1}{tk} \sum_{ij} x_{ij} \right) = f_1(u, v) \geq \Omega(d/t).$$

This proves Lemma 3.7.

Claim 3.9. *For each edge uv, $g(uv) = \Omega(\rho_u + \rho_v)$ with failure probability at most $1/2^{\Omega(k)}$.*

Proof. Let N_j be the indicator random variable for the event $\beta_{uj} \neq \beta_{vj}$. Since N_j's are independent and their sum N has expectation $k/2$, by Chernoff Bounds $N \geq k/4$ with the desired failure probability. □

Now fix an ϵ-long edge uv and let $d = d_{uv}$. Without loss of generality assume $\rho_u \leq \rho_v$; note that $\rho_u \leq d$. Since $B_u(\rho_u) \subset B_v(\rho_u + d)$, the cardinality of the latter ball is at least ϵn. It follows that $\rho_v \leq \rho_u + d$, so $g(uv) \leq \rho_u + \rho_v \leq 3d$. Since $f(uv) \leq d$, the embedded uv-distance is $O(d)$.

To lower-bound the embedded uv-distance, note that with failure probability at most $t/2^{\Omega(k)}$ the following happens: if edge uv is ϵ-good then this distance is $\Omega(d/t)$ due to $f(uv)$; else it is $\Omega(d)$ due to $g(uv)$. For part (a) we use Markov inequality to show that with failure probability at most δ this happens for all but an ϵ-fraction of ϵ-long edges. For part (b) we take a Union Bound to show that with failure probability at most δ this happens for *all* ϵ-long edges. This completes the proof of correctness for the centralized embedding.

It remains to provide the beacon-based version of the algorithm. Let S be the union of all sets S_{ij}. The Bourgain-style part of the algorithm depends only on distances to the $\Theta(k/\epsilon)$ nodes in S, so it can be seen as beacon-based, with all nodes in S acting as beacons. To define the second block of coordinates we need to know the ρ_u's, which we do not. However, we will estimate them using the same set S of beacons.

Fix a node u. Let B be the open ball around u of radius ρ_u, i.e. the set of all nodes v such that $d_{uv} < \rho_u$. Let B' be the smallest ball around u that contains at least $4\epsilon n$ nodes. Note that S is a set of ck/ϵ beacons chosen independently and uniformly at random, for some constant c.

In expectation at most ck beacons land in B, and at least $4ck$ beacons land in B'. By Chernoff Bounds with failure probability at most $1/2^{\Omega(k)}$ the following event E_u happens: at most $2ck$ beacons land in B, and at least $2ck$ beacons land in B'. Rank the beacons according to its distance from u, and let w be the $(2ck)$-th closest beacon. Define our estimate of ρ_u as $\rho'_u = d_{uw}$. Note that if event E_u happens, then ρ'_u lies between ρ_u and $\rho_u(4\epsilon)$.

Consider a 4ϵ-good edge uv such that both E_u and E_v happen. Then (as in the non-beacon-based proof) we can upper-bound the embedded uv-distance by $O(d_{uv})$, and lower-bound it by $\Omega(d_{uv}/t)$ with high probability. For part (a) we use Markov inequality to show that with failure probability at most δ event E_u happens for all but an ϵ-fraction of nodes. For part (b) we take a Union Bound to show that with failure probability at most δ this event happens for all nodes. □

The following theorem lower-bounds the target dimension required for ϵ-uniform slack, essentially showing that in part (b) of Theorem 3.6 the dependence of dimension on $\log n$ is indeed necessary.

Theorem 3.10. *For any $\epsilon < \frac{1}{2}$ there is a metric (V, d) such that any ϵ-uniform slack embedding into l_p, $p \geq 1$ with distortion D requires $\Omega(\log_D n)$ dimensions.*

Proof. Take a clique on $\epsilon\, n$ red and $(1-\epsilon)n$ blue nodes, assign length two to each of the blue-blue edges, and assign unit lengths to all the remaining edges. Consider the metric generated by this graph. Now all the blue-blue edges are ϵ-long, and thus any distortion-D ϵ-uniform-slack embedding must maintain all the distances between the blue vertices. But this is just a uniform metric on $(1-\epsilon)n$ nodes, and the lower bound follows by a simple volume argument. □

Theorem 3.6 suggests a trade-off between distortion and slack. It turns out that $O(\log \frac{1}{\epsilon})$, the trade-off in the theorem, is optimal up to a constant factor. This is further discussed in Section 3.6.

3.3 Beacon-based approaches: further results

In this section we discuss further results for beacon-based approaches on doubling metrics. Firstly, we obtain a beacon-based embedding with a novel *black-box* flavor (Theorem 3.17) which closely mimics the behavior of GNP: essentially, the coordinates of every non-beacon node u can be computed separately by any black-box procedure that minimizes distortion on distances from u to the beacons. Secondly, we get rid of the dependency of n in our results on strong triangulation and on embeddings with uniform slack (Theorem 3.18). Thirdly, in Section 3.3.3 we extend our results to infinite metrics and to a version of ϵ-slack defined with respect to an arbitrary underlying measure on nodes.

The subsections in this section build on one another. Moreover, the embedding technique from this section is used in the next section (Section 3.4 on fully distributed approaches), and the structural lemma on (ϵ, μ)-packings (Lemma 3.19) is used later in Chapter 7.

3.3.1 Black-box GNP-style embedding

We will use our triangulation analysis via the following definition. Consider a set $S \subset V$ of beacons. Let us call S a *(strong)* (ϵ, δ)*-base* if for a (strong) ϵ-dense set of node pairs (u, v) there is a beacon $b \in S$ which lies within distance δd_{uv} from u or v. Note that any such set achieves a (strong) $(\epsilon, 3\delta)$-triangulation. Let us restate the conclusions from the proof of Theorem 3.3 as follows:

Theorem 3.11. *Consider an s-doubling metric on n nodes. Let $k_0 = O(\frac{1}{\epsilon})\,(2/\delta)^{2\log s}$. Then:*

(a) $(k_0 \log \frac{1}{\epsilon})$ randomly selected nodes form an (ϵ, δ)-base with probability close to 1.

(b) $(k_0 \log n)$ randomly selected nodes form a strong (ϵ, δ)-base with high probability.

For a set S of beacons, let E_S be the set of all node pairs (u, v) where at least one of u or v belongs to S. We show that if beacons form an (ϵ, δ)-base, for a sufficiently small δ, then in order to guarantee a low-distortion embedding with slack it suffices to achieve low distortion on E_S.

Lemma 3.12. *Consider a metric M with an (ϵ, δ)-base S, and suppose an embedding $f : M \to X$ has non-contracting distortion Δ on E_S, where $\Delta \leq \frac{1}{4\delta}$. Then the embedding has distortion $O(\Delta)$ with slack ϵ. Furthermore, if S is a strong (ϵ, δ)-base, then the embedding has distortion $O(\Delta)$ with ϵ-uniform slack.*

Proof. This lemma is subsumed by Lemma 3.21 in Section 3.4. This is because in the terminology of that section, the edge set E_S is an (ϵ, δ)-frame. □

In fact, for any beacon set S we are be able to guarantee distortion $\Delta = O(\log |S|)$ on E_S.

Lemma 3.13. *Consider a metric (V, d) and a set $S \subset V$ of k beacons. Then there exists a constant c_0 and an embedding into ℓ_p, $p \geq 1$ with $O(k \log k)$ dimensions that achieves distortion $(c_0 \log k)$ on the edge set E_S. Moreover, in this embedding the coordinates of every given node u are defined as a function of its distances to the nodes in S, and can be efficiently computed.*

Proof Sketch. We first embed S using the algorithm of Bourgain [Bou85, LLR95]. Recall that this involves choosing, for each $i = 1, 2, \ldots, \lfloor \log k \rfloor$, a collection of x subsets of B of size 2^i, each uniformly at random. Let S_{ij} denote the j^{th} of these. We assign each node $b \in S$ a coordinate corresponding to each set S_{ij}, defined to be $d(b, S_{ij})$, the minimum distance between b and any point in S_{ij}.

Having embedded the beacons, we then embed every other node u using these same sets $\{S_{ij}\}$; for each S_{ij}, node u constructs a coordinate of value $d(u, S_{ij})$. In the approach of Linial et al., $x = O(\log k)$ sets of each size are chosen. Here, by way of contrast, we take $x = \Theta(k)$; we claim that with this choice of random sets $\{S_{ij}\}$ in the embedding, the set of node-beacon pairs is embedded with distortion $O(\log k)$ with probability close to 1.

To establish this claim, we give upper and lower bounds on the embedded distances; the calculations here differ from [Bou85, LLR95] in that we will be taking a union bound over subsets of beacons, rather than over the much larger set of all node pairs. The upper bound is straightforward, so we focus on the lower bound. Here, we fix i and let A and A' be two disjoint subsets of S of size $k/2^i$ and $2k/2^i$ respectively.

One can show there is a constant c so with probability at least c, a given S_{ij} has the property that it hits A and misses A'. Thus the expected number of S_{ij}'s with this property is ck, so applying the Chernoff bound, for large enough $x = \Theta(k)$ the probability that at most $cx/2$ of S_{ij}'s do not have this property is at most $e^{-cx/8} \leq 2^{-2k}$. Therefore with probability close to 1 for all i, for *every* pair A, A' of disjoint subsets of S of the right size, this property holds for $\Omega(k)$ sets S_{ij}. Once this is true, consider embedding any given node u, separately from all other non-beacon nodes; an analog of the telescoping-sum argument from [LLR95] gives the desired lower bound with probability close to 1. □

Let us say that a (strong) ***ϵ-base*** is a (strong) $\left(\epsilon,\, \frac{1}{4\Delta}\right)$-base S such that $\Delta = c_0 \log |S|$. Combining the previous two lemmas, we obtain a beacon-based embedding whenever the beacons form an ϵ-base.

Theorem 3.14. *Consider a metric space (V, d) and a set $S \subset V$ of k beacons. If S is a (strong) ϵ-base, then there is a $O(k \log k)$-dimensional embedding into ℓ_p which has distortion $O(\log k)$ with ϵ-(uniform) slack. In this embedding, the coordinates of every given node u are defined as a function of its distances to the beacons in S, and can be efficiently computed.*

We will use this theorem to obtain improved embeddings with uniform slack (Theorem 3.18); moreover, this embedding technique will be essential for our result on fully distributed embeddings in the next section.

In view of the above theorem, we need to make sure that a small set of beacons forms a (strong) ϵ-base. Indeed, by Theorem 3.11 such beacon sets exist and can be constructed via random node selection:

Corollary 3.15. *Consider an s-doubling metric on n nodes. Let S be the set of $k \geq 4$ randomly selected nodes. Then there exists a constant c such that:*

(a) if $k \geq (s/\epsilon)^{c \log\log(s/\epsilon)}$ then S is an ϵ-base with probability close to 1.

(b) if $k \geq x^{c \log\log x}$, $x = \frac{s}{\epsilon} \log n$, then S is a strong ϵ-base with high probability.

Proof Sketch. Let c_0 be the constant from Lemma 3.13. We start with k and define $\delta = (4 c_0 \log k)^{-1}$. Take $k_0 = O(\frac{1}{\epsilon})\,(2/\delta)^{2 \log s}$ from Theorem 3.11. Then it suffices to check that $k \geq k_0 \log \frac{1}{\epsilon}$ for part (a), and that $k \geq k_0 \log n$ for part (b). □

Theorem 3.14 does not quite capture the full power of Lemma 3.12 and Lemma 3.13. We can further exploit these two lemmas to obtain a beacon-based embedding with a novel *black-box* flavor: beacons are embedded first (by inspecting only the distances between the beacons), and then the coordinates of every non-beacon node u can be computed separately by any black-box procedure that inspects the distances from u to the beacons and minimizes distortion on these distances. This closely mimics the behavior of GNP.

Definition 3.16. Consider a metric (V, d), node set $S \subset V$, an embedding $f : S \to X$, and a node $u \notin S$. Then a ***(u, Δ)-extension*** of f is an embedding $g : S \cup \{u\} \to X$ that coincides with f on S and has distortion Δ on node pairs (u, v), $v \in S$.

Theorem 3.17. *Fix $p \geq 1$ and let c_0 be the constant from Lemma 3.13. For any metric (S, d) there exists an embedding $f_{(S,d)} : S \to \ell_p^{\Theta(|S| \log |S|)}$ with distortion $c_0 \log |S|$ and the following property (*):*

Property (*). Consider a metric (V, d) and a beacon set $S \subset V$. Let $f = f_{(S,d)}$ and let $\Delta = c_0 \log |S|$.

(a) For each node $u \notin S$ there exists a (u, Δ)-extension of f. Let g_u be any such extension.

(b) Let $g : V \to \ell_p$ be an embedding that coincides with f on S, and equals to $g_u(u)$ for every node $u \notin S$. If S is a (strong) $\left(\epsilon,\, \frac{1}{4\Delta}\right)$-base, then g achieves distortion $O(\Delta)$ with ϵ-(uniform) slack.

Remark. In part (a), in order to construct a suitable g_u it suffices to inspect only the coordinates of the beacons under f and the distances from u to the beacons. A key feature of this theorem is that it does not require neither any specific embedding g_u nor any specific procedure to compute it: any black-box procedure that computes a (u, Δ)-extension of f would work.

3.3.2 Strong triangulation and uniform slack with a constant number of beacons

Recall our results on strong triangulation and on embeddings with uniform slack required the number of beacons which was proportional to $\log n$. It turns out that for doubling metrics we can get rid of this dependency on n.

Theorem 3.18. *Consider an s-doubling metric space and fix $\epsilon > 0$.*

(a) For any $\delta > 0$ there exists a strong (ϵ, δ)-base of size $k = \frac{2}{\epsilon}\,[O(\frac{1}{\delta})]^{\log s}$. Moreover, a set of $O(k \log k)$ randomly chosen nodes forms a strong (ϵ, δ)-base with probability close to 1. Recall that using any (ϵ, δ)-base as a set of beacons leads to a strong (ϵ, δ)-triangulation.

(b) There exists an embedding into $\ell_p^{O(k \log k)}$, $p \geq 1$ with distortion $O(\log k)$ and ϵ-uniform slack, where $k = (\frac{s}{\epsilon})^{O(\log\log(s/\epsilon))}$; such embedding can be computed with high probability by a beacon-based algorithm with k beacons selected uniformly at random.

The key to this theorem is the following lemma on the structure of doubling metrics. We will also use this lemma in the next subsection and also in Chapter 7.

Lemma 3.19. *Consider a (possibly infinite) complete metric space of doubling dimension α, equipped with a probability measure μ. Let $r_u(\epsilon, \mu)$ be the radius of the smallest ball around u that has measure ϵ. Then for any $\epsilon > 0$ there exists an* **(ϵ, μ)-packing***: a family $\mathcal{F}$ of disjoint balls of measure at least $\epsilon/16^\alpha$ each, such that for any node u there exists a ball $B_v(r) \in \mathcal{F}$ such that $d_{uv} + r \leq 6r_u(\epsilon, \mu)$. Moreover, if the metric is finite then such $\mathcal{F}$ can be efficiently computed.*

It is easy to see that if μ is a doubling measure then for every node u this (ϵ, μ)-packing $\mathcal{F}$ has the two useful local properties of an r-net, $r = 6r_u(\epsilon, \mu)$: firstly, the ball $B_u(r)$ contains at least one element of $\mathcal{F}$, and secondly, for any k the ball $B_u(kr)$ contains at most $k^{O(\alpha)}$ elements of $\mathcal{F}$. The notion of (ϵ, μ)-packing allows us to state these properties in terms of the underlying doubling measure, and, moreover, to generalize them to arbitrary probability measures. In this subsection we will use (ϵ, μ)-packings such that μ is the counting probability measure (a measure μ such that $\mu(u) = 1/n$ for every node u). We will use the full generality of this lemma in the next subsection.

Proof of Lemma 3.19: Let us fix ϵ and let $r_u = r_u(\epsilon, \mu)$. For a given node u, say a ball $B_v(r)$ is *u-zooming* if it is a subset of $B_u(3r_u)$, has measure at least $\epsilon/16^\alpha$, and $B_v(4r)$ has measure at most ϵ. We claim that for every node u either there exists a u-zooming ball, or there exists a node $b_u \in B_u(2r_u)$ of measure at least ϵ.

Suppose not. Let $r = r_u$. By the doubling property of the metric (see Lemma 2.17), $B_u(r)$ can be covered by 16^α balls of radius $r/8$. At least one of these balls, say $B_v(r/8)$, has cardinality at least $\epsilon/16^\alpha$; since without loss of generality $B_v(r/8)$ overlaps with $B_u(r)$, it follows that $d_{uv} \leq \frac{9}{8}r$ and $B_v(r/2) \subset B_u(2r)$. Since there is no u-zooming ball, in particular the ball $B_v(r/8)$ is not u-zooming, so $B_v(r/2)$ has measure at least ϵ.

Iterating this argument i times, we come up with a node v_i such that $d(v_{i-1}, v_i) \leq \frac{9}{8}r/2^i$ and the ball $B_{v_i}(r/2^i)$ has cardinality at least ϵ. If the metric is finite, then for large enough i this ball consists of only one node v_i, which therefore has measure at least ϵ. This is a contradiction since $d(u, v_i) \leq \frac{9}{8}r(2 - 2^{-i})$. Now if the metric is infinite, then we have an infinite Cauchy sequence of nodes $\{v_i\}$. Since the metric is complete, this sequence has a limit, call it v; note that $v \in B_u(2r)$. Then for each i the ball $B_v(3r/2^i)$ contains ball $B_{v_i}(r/2^i)$, hence has measure at least ϵ. Therefore node v has measure at least ϵ, contradiction. Claim proved.

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

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

Suppose that for a given v the claim is false. Since by definition of a v-zooming ball $B_v \subset B_v(3r_v)$, it follows that $B_v \notin \mathcal{F}$. Since $\mathcal{F}$ is maximal, B_v overlaps with some ball $B_u \in \mathcal{F}$. If $B_u = \{b_u\}$ then it trivially lies in $B_v(3r_v)$, contradiction. So B_u is a u-zooming ball; say w is its center, and r is its radius. By definition of a u-zooming ball, $B_w(4r)$ has measure at most ϵ. If $4r \geq d_{vw} + r_v$, then ball $B_w(4r)$ contains ball $B_v(r_v)$; as the latter ball has measure at least ϵ, the two balls coincide, and thus B_u lies in $B_v(r_v)$, contradiction. Therefore $4r < d_{vw} + r_v$.

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

Proof Sketch of Theorem 3.18: For part (a), let us fix ϵ, δ and take $\epsilon_\delta = \frac{1}{2}\,\epsilon\,(\delta/2)^{2\log s}$ as in Lemma 3.4. Let μ be the counting probability measure, and let $\mathcal{F}_\delta$ be an (ϵ_δ, μ)-packing guaranteed by Lemma 3.19. Say $S \subset V$ is a δ-hitting set if it hits a ball of radius $6r_u(\epsilon_\delta)$ around every node u. Note that S is δ-hitting if it hits every ball in $\mathcal{F}_\delta$. Moreover, since the balls in $\mathcal{F}_\delta$ are disjoint and have measure at least $\epsilon^* = \epsilon_\delta / s^4$ each, it follows that $O(1/\epsilon^*)\log(1/\epsilon^*)$ randomly chosen nodes suffices to form a δ-hitting set with probability close to 1.

Let H_δ be a δ-hitting set. We claim that $H_{\delta/6}$ is a strong (ϵ, δ)-base. Indeed, recall that by Lemma 3.4 for each node u there exists a set S_u of measure at least $1-\epsilon$ which has the following property: for every $v \in S_u$ a ball around u or v of radius δd_{uv} has cardinality at least ϵ_δ. Therefore for every $v \in S_u$ some node in H_δ lies within distance $6\delta d_{uv}$ from u or v. Claim proved. It immediately follows that we can use $H_{\delta/6}$ as the beacon set to obtain the desired strong (ϵ, δ)-triangulation.

For part (b), we claim that a set S of $k = (\frac{s}{\epsilon})^{O(\log\log(s/\epsilon))}$ randomly selected beacons is an ϵ-base with probability close to 1. Indeed, we need to use part (a) to check that S is an (ϵ, δ)-base for $\delta = (4c_0 \log k)^{-1}$, where c_0 is the constant from Lemma 3.13; we omit the details. Now part (b) follows by Theorem 3.14. □

3.3.3 Infinite metrics and arbitrary measures

In the previous sections, our results for beacon-based approaches are defined for finite metrics; ϵ-dense sets are (essentially) defined with respect to the counting measure. Here we extend them to infinite metrics and arbitrary probability measures. Specifically, suppose we are given a probability measure μ on V. This measure induces a product measure on node pairs. We can define an (ϵ, δ, μ)-*triangulation* and embeddings with a (ϵ, μ)-slack, where the desired properties hold for a set of edges of measure at least $1-\epsilon$. Also, we can define a strong (ϵ, δ, μ)-*triangulation* and embeddings with a (ϵ, μ)-uniform slack; here the desired properties hold for all node pairs (u, v), $v \in S_u$ where $\mu(S_u) \geq 1-\epsilon$.

Our result on beacon-based embeddings for arbitrary metrics (Theorem 3.6) extends to the (strong) (ϵ, μ)-slack setting in a straightforward way. In the embedding algorithm, instead of selecting beacons uniformly at random (i.e. with respect to the counting measure) we select them with respect to measure μ; the proof carries over without much modification. Moreover, part (a) (the part about (ϵ, μ)-slack) extends to infinite metrics.

In order to achieve similar extensions for triangulation and for embeddings with ϵ-uniform slack, we need the machinery in this section. Specifically, for any probability measure μ on V we can prove the analog of Theorem 3.18 with (ϵ, δ, μ)-base instead of (ϵ, δ)-base, strong (ϵ, δ, μ)-triangulation instead of strong (ϵ, δ)-triangulation, and (ϵ, μ)-uniform slack instead of ϵ-uniform slack.

In a sketch, we can prove such theorem proceeds as follows. First we note that Lemma 3.12 extends to the new setting: it suffices to guarantee low distortion on distances to beacons as long as they form an (ϵ, δ, μ)-base for a sufficiently small δ. Consequently Theorem 3.14 extends to the new setting, too. Then we just mimic the proof of Theorem 3.18 using (the full generality of) Lemma 3.19.

3.4 Fully distributed approaches

Recent work in the networking literature has considered so-called 'fully distributed' approaches to triangulation and embedding problems, in which no single node has to perform a large number of measurements [DCKM04, PCW$^+$03, ST03]. Instead, for a relatively small parameter k, each node selects k virtual 'neighbors' uniformly at random and measures distances to them; let E_k denote the set of all pairs (u, v) where v is one of the selected neighbors of u. All nodes then run a distributed algorithm that uses the measured distances on the pairs E_k to embed the full metric. The distributed algorithms in these papers are based on different heuristics: *Vivaldi* [DCKM04] simulates a network of physical springs, *Lighthouse* [PCW$^+$03] uses global-local coordinates, and [ST03] claims to simulate the Big Bang explosion. They offer no proofs, but their experimental results are quite strong. In particular, *Vivaldi* [DCKM04] uses the testbed from the *GNP* algorithm [NZ02] and claims slightly better performance. Here we consider what kinds of theoretical guarantees can be obtained for algorithms of this type; as in previous sections, we focus on doubling metrics.

First, suppose we view the distributed embedding heuristic as a black box that embeds the nodes with distortion at most Δ on the pairs E_k. Is this enough to provide a guarantee for the full metric?

Definition 3.20. Given a set E of node pairs in a metric, we can consider the weighted graph $G(E)$ in which these pairs form the edges, and each edge (u, v) is labeled with the distance d_{uv}. We say that a uv-path P in $G(E)$ is *δ-skewed* if for some $e \in P$, the total edge weight of $P \setminus \{e\}$ is at most δd_{uv}, and e is incident to one of u or v — in other words, P consists of an initial "long hop" followed by a number of short ones. Finally, we say that the set of pairs E is a (strong) **$(\boldsymbol{\epsilon}, \boldsymbol{\delta})$-frame** if $G(E)$ contains a δ-skewed path for all pairs in a (strong) ϵ-dense set. We will assume throughout this section that δ is sufficiently small: $\delta < 1/4$.

Frames E as defined here have a useful "rigidity" property, as the following result shows: an embedding with bounded distortion on the pairs in E must also have bounded distortion on all but an ϵ-fraction of node pairs. In this sense, frames have a similar flavor to *spanners*, but they include a slack parameter and also require the approximately distance-preserving paths to have a particular "skewed" structure.

Lemma 3.21. *Consider a metric M with a (ϵ, δ)-frame E, and suppose an embedding $f : M \to X$ has non-contracting distortion Δ on E, where $\Delta \leq \frac{1}{4\delta}$. Then the embedding has distortion $O(\Delta)$ with slack ϵ. Furthermore, if E is a strong (ϵ, δ)-frame, then the embedding has this distortion with ϵ-uniform slack.*

Proof. Let d^X be the distance function on X; for nodes $u, v \in M$, let us write d^*_{uv} for $d^X_{f(u),f(v)}$.

Suppose the pair (u, v) has a δ-skewed path P in $G(E)$, with long edge (u, p). By the definition of a frame combined with the triangle inequality, we have $(1-\delta)d_{uv} \leq d_{up} \leq (1+\delta)d_{uv}$. Since the embedding has non-contracting distortion Δ on E, we have $(1 - \delta) \leq d^*_{up}/d_{uv} \leq \Delta(1 + \delta)$ and $d_{vp} \leq \Delta\delta d_{uv}$; hence, using the assumptions that X is a metric and that $\delta < 1/4$, we have

$$d^*_{uv} \in [d^*_{up} - d^*_{vp},\ d^*_{up} + d^*_{vp}] \subseteq d_{uv}\,[1 - \delta - \Delta\delta,\ \Delta(1 + 2\delta)] \subseteq d_{uv}\,[\tfrac{1}{2},\ \tfrac{3}{2}\Delta].$$

It follows that the distortion of f is $O(\Delta)$ on the set of all pairs that have a δ-skewed path. □

By Lemma 3.21, it suffices to show that the set of pairs E_k forms an (ϵ, δ)-frame for $\delta \leq \frac{1}{4\Delta}$; then we have an embedding of the full metric with distortion $O(\Delta)$ and slack ϵ.

Theorem 3.22. *Let M be a doubling metric. There exists $k = (2\log n)^{O(1)}$ such that for any ϵ and δ that are each at least $\Omega(1/\log^{O(1)} n)$, the set E_k of probed edges is a strong (ϵ, δ)-frame with high probability.*

Proof. Let the doubling dimension of M be 2^s. For some constant c to be defined later, set $\delta^* = \delta/(c\log^2 n)$ and $\epsilon^* = \frac{\epsilon}{2}(\delta^*/2)^{2\log s}$. By Chernoff bounds taking

$$k = O(\tfrac{1}{\epsilon^*}\log n) = O(\tfrac{1}{\epsilon})\,(\tfrac{1}{\delta})^{2\log s}\, s^{O(1)}\,(\log n)^{1+4\log s} \tag{3.3}$$

suffices to make sure that with high probability each node has at least $3\log n$ neighbors in a ball of size $\epsilon^* n$ around every other node. By Lemma 3.4, for a strong ϵ-set of node pairs uv, a ball of size $\epsilon^* n$ around one of the nodes (say v) has radius at most $\delta^* d_{uv}$. As we argued, u has a neighbor in this ball, call it w. Now, each node in this ball has at least $3\log n$ neighbors in it, chosen uniformly at random. Therefore by Theorem 2.4 the graph induced by this ball in E_k contains an $O(\log n)$-degree expander, and hence by Theorem 2.2 has diameter at most $c\log^2 n$ for some constant c. This is the c that we use in the definition of δ^* and ϵ^*. In particular, E_k contains a vw-path with at most $c\log n$ hops, each of length at most $\delta^* d_{uv}$, so the metric length of this path is at most δd_{uv}. Therefore the edge set E_k is a strong (ϵ, δ)-frame. □

Theorem 3.22 already helps provide some underpinning for the success of distributed embedding heuristics in recent networking research. But to go beyond this black-box result to concrete distributed algorithms, we need to think about techniques for triangulation and embedding that operate in a decentralized fashion on the graph $G(E_k)$. In this section, we focus on the problem of distributed triangulation in particular.

Here's a schematic description of a distributed triangulation algorithm. First, a (small) number of nodes S declare themselves to be *beacons*. Messages are then passed over the edges of the graph $G(E_k)$, at the end of which each node u has, for each beacon b, a pair of upper and lower bounds $D^-_{ub} \leq d_{ub} \leq D^+_{ub}$. This is the crux: unlike standard beacon-based algorithms, node u never actually measures its distance to beacon b (unless they happen to be neighbors in $G(E_k)$), so it must infer bounds on the distance from the distributed algorithm. Finally, the distance between two non-beacon nodes u and v can be estimated via

$$\max_{b\in S}\left(|D^+_{ub} - D^-_{vb}|,\ |D^+_{vb} - D^-_{ub}|\right) \leq d_{uv} \leq \min_{b\in S}\left(D^+_{ub} + D^+_{vb}\right).$$

We denote the left-hand and the right-hand sides by D^-_{uv} and D^+_{uv}, respectively, and say such process is a (strong) (ϵ, δ)-*triangulation* if $D^+_{uv} \leq (1+\delta)\, D^-_{uv}$ for a (strong) ϵ-dense set of node pairs. Note that this definition of triangulation generalizes the one for the beacon-based triangulation in Section 3.1: if we measure the distance between node u and beacon b, then we just set $D^+_{ub} = D^-_{ub} = d_{ub}$.

Given a set E_k of measured distances as in Theorem 3.22, our goal is to perform triangulation with only a small number of messages passed between nodes.

Theorem 3.23. *Let M be a doubling metric, and suppose that every node has $k = (2\log n)^{\Omega(1)}$ neighbors chosen independently and uniformly at random.[1] Then for any ϵ and δ that are each at least $\Omega(1/\log^{O(1)} n)$, an (ϵ, δ)-triangulation can be achieved with high probability in time polylogarithmic in n, with only a polylogarithmic load per node, taking into account the work for distance measurements, storage, and the number of bits sent and received.*

Proof. We will use the following multi-stage algorithm:

Algorithm 3.24. *Suppose each node knows (ϵ, δ, n) and chooses (ϵ^*, k, c) as in Theorem 3.22.*

[1] The value of k in terms of (ϵ, δ) and the doubling dimension 2^s is given in (3.3).

1. *Each node selects k neighbors[2] uniformly at random, measures distances to them, and decides (independently, with probability k/n) whether it is a beacon.*

2. *Beacons announce themselves to their neighbors. Specifically, each beacon b sorts its measurements from low to high and estimates $r_b(\epsilon^*)$ by the measurement ranked $2\epsilon^* k$. Call this measurement r_b. Then it sends a message $M(b, r_b, i)$ to all its neighbors, where i is the number of hops that the message has traversed, initially set to 0.*

3. *When node u receives $M(b, r_b, i)$ from v, node u updates its existing bounds on d_{ub} using the new bounds $d_{uv} \pm 2ir_b$. Say the message is* new *if u does not already store $M(b, r_b, i')$ with $i' \leq i$. If so and moreover $d_{uv} \leq 2r_b$ and $i < c \log n$, then u stores it and forwards $M(b, r_b, i+1)$ to all its neighbors but v.*

We now analyze this algorithm. Let $K = c \log n$. Each message is forwarded at most K times, yielding the claimed running time. A given node can broadcast the message from a given beacon at most K times, yielding the claimed number of messages per node. When $M(b, r_b, i)$ is forwarded, all hops but possibly the last one have length at most r_b, so the distance bounds in step 3 are valid.

By a straightforward application of Chernoff bounds, it holds with high probability for every beacon b that at most $2\epsilon^* k$ neighbors lie within distance $r_b(\epsilon^*)$ from b, and at least $2\epsilon^* k$ neighbors lie within distance $r_b(4\epsilon^*)$ from b, so $r_b(\epsilon^*) \leq r_b \leq r_b(4\epsilon^*)$.

Let b be a beacon, and let B_b be the smallest ball around b that has size at least $\epsilon^* n$. In the proof of Theorem 3.22 we saw that the graph induced by this ball in the edge set E_k has diameter at most K. Since $r_b \geq r_b(\epsilon^*)$, each $w \in B_b$ will receive a message from b via a path of at most K hops of length at most $2r_b$ each, so w will upper-bound d_{wb} by $D^+_{wb} \leq 2r_b K$. Moreover, since (by the proof of Theorem 3.22) every node u has a neighbor $w \in B_b$, node u will receive a message from beacon b via this node w, and consequently bound d_{ub} by $d_{uw} \pm D^+_{wb}$, which is (at worst) $d_{ub} \pm 3r_b K$. We have proved the following:

Claim 3.25. *With high probability for each node u and beacon b bounds $D^{\pm}_{ub}$ lie within $d_{ub} \pm O(r_b \log n)$.*

Now, by Lemma 3.4 there exists an ϵ-set of node pairs (u, v) such that the ball B around u or v of radius $r = O(\delta d_{uv}/\log n)$ has at least $4\epsilon^* n$ points. With high probability, each such ball B contains a beacon, call it b. Since $B_b(2r)$ contains B, $r_b \leq r_b(4\epsilon^*) \leq 2r$. We have proved the following:

Claim 3.26. *With high probability for each node pair (u, v) in a strong ϵ-dense edge-set, there exists a beacon b such that* $\min(d_{ub}, d_{vb}) \leq r$ *and $r_b \leq r$, for some $r = O(\delta d_{uv}/\log n)$.*

It is easy to see that such beacon b yields bounds on d_{uv} that are within $d_{uv}\,(1 \pm O(\delta))$. □

Now let us extend the above algorithm for triangulation to a fully distributed algorithm that computes a low-dimensional embedding into ℓ_p, $p \geq 1$ which has low distortion with slack. In fact, for any given $\epsilon > 0$ we compute an embedding with ϵ-slack that has dimension and distortion that depend only on the doubling dimension and the parameter ϵ, not on the number of nodes in the system.

Theorem 3.27. *Let M be a doubling metric, and suppose that every node has $k = (2\log n)^{\Omega(1)}$ neighbors chosen independently and uniformly at random. Then there exists a fully distributed algorithm that given $\epsilon \geq (\log n)^{-O(1)}$ with high probability constructs a $O(k \log k)$-dimensional embedding into ℓ_p, $p \geq 1$ which has distortion $O(\log k)$ with ϵ-uniform slack. In this algorithm the per-node load and the total completion time are at most $O(k^2 \log^3 n)$.*

[2] Neighbors are undirected, in the sense that if u selects v as a neighbor, then u becomes a neighbor of v, too.

In the remainder of this section we prove Theorem 3.27.

Let 2^s be the doubling dimension of M. Let us fix (ϵ, s) and assume that they are known to the participating nodes. Take $\delta = c/\log n$, where c is a constant to be specified later, and let k be defined by (3.3).

The high-level algorithm is simple. First the nodes compute an (ϵ, δ)-triangulation using Algorithm 3.24; note that such triangulation uses at most k and at least $\Omega(k)$ beacons. Then the beacons measure distances to one another and broadcast them to the entire network using a uniform gossip [Pit87]; in this phase each beacon broadcasts one message of size $O(k)$, the total per-node load being at most $O(k^2 \log n)$. Upon receiving this information nodes update the bounds D^+ on their distances to beacons accordingly, by running a shortest-paths algorithm on the available distances. (Note that in this step D^+ can only decrease, but not below the true distance; in particular, Claim 3.25 still holds.) Finally, nodes run the embedding algorithm in Theorem 3.14 *with the same beacon set*, using the upper bounds D^+ instead of the latent true distances to the beacons.

Our proof outline follows that of Theorem 3.14, but the details are quite different and significantly more complicated. As in Theorem 3.14, first we bound the distortion on node-to-beacon distances, then use those to bound distances between other node pairs. However, we need to compensate for the fact that D^+, the distance function that we are actually embedding, is not necessarily a metric. In particular, in our proof D^+ is more than just a function that approximately obeys the triangle inequality: it is essential that D^+ is close to a specific metric, as expressed by Claim 3.25 and Claim 3.26. We will use these two claims to reason about the embedded distances to beacons, which is why we use the same set of beacons for both triangulation and embedding.

For completeness let's restate the embedding algorithm. Let S_{beac} be the beacon set from the (ϵ, δ)-triangulation; for simplicity assume there are exactly k beacons. For each $i \in [\log k]$ choose $\Theta(k)$ random subsets of S_{beac} of size 2^i each; let S_{ij} be the j-th of those. These subsets are broadcasted to the entire network using a uniform gossip [Pit87]: one message of size $O(k^2)$ is broadcasted, incurring a per-node load at most $O(k^2 \log n)$. Then every node u embeds itself into ℓ_p so that each dimension ij is defined as $D^+(u, S_{ij})/\Theta(k)$, where $D^+(u, S)$ is the smallest D^+_{uv} such that $v \in S$. Recall that we use $\Theta(k)$ beacon sets of each size scale, not $\Theta(\log k)$ as [LLR95], in order to guarantee the following claim from the proof of Lemma 3.13:

Claim 3.28. *With high probability for any $i \in [\log k]$ and any pair of disjoint subsets $S, S' \subset S_{beac}$ of size at least $k/2^i$ and at most $2k/2^i$, respectively, it is the case that at least $\Omega(k)$ sets S_{ij} hit S and miss S'.*

Then, letting d^*_{uv} be the uv-distance in the embedding, we can bound the embedded node-to-beacon distances:

Claim 3.29. *Whp for each node u and every beacon b we have $d_{ub} \leq d^*_{ub} \leq O(\log k) D^+_{ub}$.*

Now by Claim 3.26 with high probability for an ϵ-dense set of node pairs (u, v) there is a beacon b within distance $O(r)$ from u or v (say, from v) such that $r_b \leq O(r)$, for some $r = O(\delta d_{uv}/\log n)$. Therefore Claim 3.25 for any such node pair (u, v) we have

$$(1 - O(\delta)) d_{uv} \leq d^*_{ub} \leq O(\log k) d_{uv}$$

and $d^*_{vb} \leq O(\log k) \delta d_{uv}$, so it follows that

$$d_{uv}/2 \leq d^*_{ub} - d^*_{vb} \leq d^*_{uv} \leq d^*_{ub} + d^*_{vb} \leq O(\log k) d_{uv}$$

as long as the constant c that defines δ is small enough.

To complete the proof of Theorem 3.27 it remains to prove Claim 3.29. For simplicity consider the case $p = 1$ first. For a node set S and any pair uv of nodes define $D^+_{uv}(S) = |D^+(u, S) - D^+(v, S)|$. Then the embedded uv-distance d^*_{uv} is equal to the sum $\sum D^+_{uv}(S_{ij})$ over all beacon sets S_{ij}. In order to establish the desired upper bound on d^*_{ub} it suffices to prove that if u is a node, b is a beacon and S is a set of beacons then $D^+_{ub}(S) \leq 2D^+_{ub}$. It will follow by a standard argument from the following claim: $|D^+_{ub'} - D^+_{bb'}| \leq 2D^+_{ub}$ for any two beacons b, b'.

Let's prove this claim. Consider the beacon b_u that is closest to u with respect to D^+; let $x = D^+(u, b_u)$ and $y = d(b', b_u)$. The beacons measure distances to each other, so $D^+_{bb'} = d_{bb'}$. Node u has updated $D^+_{ub'}$ according to these measurements, so it is at most $x + y$; obviously, it is at least $d_{ub'}$, which is lower-bounded by $y - x$. Therefore $|D^+_{ub'} - y| \leq x$, so, completing the proof,

$$|D^+_{ub'} - D^+_{bb'}| \leq |y - d_{bb'}| + |D^+_{ub'} - y| \leq d(b, b_u) + x \leq d_{bu} + 2x \leq 3D^+_{ub}.$$

It remains to establish the lower bound in Claim 3.29, which we will accomplish by a version of Bourgain's telescoping sum argument. Let $S_u(r)$ be the set of beacons b such that $D^+_{ub} \leq r$. For a fixed node u and beacon b, let $\rho_i = \min(\rho_u(i), \rho_v(i), d_{ub}/2)$, where $\rho_u(i)$ is the smallest r such that $S_u(r)$ contains at least $k/2^i$ beacons.

We claim that for each given i the sum $X_i = \sum_j D^+_{ub}(S_{ij})$ is at least $\Omega(k)(\rho_{i-1} - \rho_i)$. Indeed, fix i and without loss of generality assume that $\rho_u(i) \leq \rho_b(i)$. Note that the sets $S = S_u(\rho_i)$ and the interior S' of $S_b(\rho_{i-1})$ are disjoint since if a node v belongs to both S and S' then

$$d_{ub} \leq d_{uv} + d_{bv} \leq D^+_{uv} + D^+_{bv} < \rho_i + \rho_{i-1} \leq d_{ub},$$

contradiction. Therefore by Claim 3.28 with high probability for each i at least $\Omega(k)$ sets S_{ij} hit S and miss S', thus contributing at least $\rho_{i-1} - \rho_i$ each to X_i. This proves the claim.

Let $t = \lfloor \log k \rfloor$ and note that by definition $\rho_b(t) = 0$ (since $S_b(0)$ contains at one beacon, namely b itself), so $\rho_t = 0$. Summing up the X_i's we get $d^*_{ub} \geq \Omega(k)(\rho_1 - \rho_t) = \Omega(k) d_{ub}$ as desired, as long as $\rho_1 \geq d_{ub}/4$. Now suppose $\rho_1 < d_{ub}/4$ and assume that $\rho_u(1) < \rho_b(1)$ (the case $\rho_u(1) \geq \rho_b(1)$ is treated similarly). Then the sets $S = S_u(d_{ub}/4)$ and $S' = S_{\text{beac}} \setminus S$ are disjoint and have size at least $n/2$ and at most $n/2$, respectively. Therefore by Claim 3.28 with high probability at least $\Omega(k)$ sets S_{1j} hit S and miss S', thus contributing at least $D^+_{ub}/2 = \Omega(d_{ub})$ each to X_i, so that $d^*_{ub} \geq \Omega(k) d_{ub}$ as desired. This completes the proof of Claim 3.29 for $p = 1$. We can extend it to general $p \geq 1$ following [LLR95]; we omit the details. This completes the proof of Theorem 3.27.

3.5 Improved embeddings for growth-constrained metrics

We can obtain a number of improvements to our results when a given metric is growth-constrained.

Firstly, we show that the following simple *nearest-beacon* embedding is effective in growth-constrained metrics: select k beacons uniformly at random, embed the beacons with distortion $O(\log k)$ (e.g. using the Bourgain's algorithm), and then simply position each non-beacon node at the embedded location of its nearest beacon. The sufficient number of beacons is then a function of grid dimension and slack ϵ.

Theorem 3.30. *Consider a metric d with grid dimension α. Then for any $\epsilon > 0$ the nearest-beacon embedding with $k = O(4^\alpha)(\frac{1}{\epsilon}\log\frac{1}{\epsilon})$ beacons has distortion $O(\alpha + \log\frac{1}{\epsilon})$ with slack ϵ.*

We defer the proof till later in this section. Combined with the fully distributed triangulation from Section 3.4, the nearest-beacon embedding yields a fully distributed (*Vivaldi*-style) embedding for growth-constrained metrics. Specifically, fix $\epsilon > 0$, choose k as in the above theorem, set $\delta = 1/\Theta(\log k)$ and perform a fully distributed (ϵ, δ)-triangulation from Theorem 3.23. Then for each non-beacon node u, choose the nearest beacon with respect to the triangulation (say, with respect to the upper bound D^+), and position u at the embedded location of this beacon. The proof proceeds similarly.

It is worth noting, on the other hand, that there are doubling metrics in which this nearest-beacon embedding does not yield good results. Specifically, consider the *exponential line*, which is the point set $V = \{2^i : i \in [n]\}$ equipped with the one-dimensional distance function $d(x, y) = |x - y|$. Recall from Section 2.3 that this is a standard example of a doubling metric that is very far from being growth-constrained. Suppose we choose a set S of beacons in this metric. Then for any node $u \in V$ the nearest beacon is $\min B$, so all non-beacon nodes are mapped to the same beacon.

Our second result is an embedding with gracefully degrading distortion. Qualitatively this is a special case of a much more complicated embedding in Chapter 4 (the quantitative guarantees are slightly different).

Theorem 3.31. *Consider a metric with grid dimension α. Then it can be embedded into ℓ_p, $p \geq 1$ with $O(\log^2 n)$ dimensions and gracefully degrading distortion $O(\alpha + \log \frac{1}{\epsilon})$. In particular, such embedding is achieved by Bourgain's algorithm.*

A beacon-based version of the above theorem produces ”gracefully degrading distortion with slack”:

Theorem 3.32. *Consider a metric with grid dimension α. For any $\epsilon^* > 0$ there exists a beacon-based algorithm which uses $O(\frac{1}{\epsilon^*} \log n)$ beacons and computes an embedding into ℓ_p, $p \geq 1$ with $O(\log n)(\log \frac{1}{\epsilon^*})$ dimensions and the following property: for any $\epsilon \geq \epsilon^*$, distortion on all ϵ-long edges is $O(\alpha + \log \frac{1}{\epsilon})$. In particular, such embedding is achieved by Bourgain's algorithm with $O(\log \frac{1}{\epsilon^*})$ higher distance scales.*

Let us proceed to the proofs.

First we use the basic structural property of growth-constrained metrics (expressed by Lemma 2.14) to derive some further structural properties which will be essential to our results. Recall that for a node u and $\epsilon \in (0, 1]$ we let $r_u(\epsilon)$ denote the smallest radius of a ball around u that contains at least ϵn nodes. For brevity, let us denote $r^+_{uv}(\epsilon) = \max(r_u(\epsilon), r_v(\epsilon))$.

Lemma 3.33. *Consider a metric d with grid dimension α, and fix positive $\epsilon > 0$. Then:*

(a) for any $\delta \in (0, 1]$ there exists an ϵ-dense set of node pairs (u, v) such that $r^+_{uv}(\epsilon\, \delta^\alpha) \leq \delta d_{uv}$.

(b) for any $(\epsilon\, 2^\alpha)$-long edge (u, v) we have $r^+_{uv}(\epsilon) \leq d_{uv}$.

(c) for any $x \geq 1$ it is the case that $r_u(\epsilon x) \geq x^{1/\alpha}\, r_u(\epsilon)$.

Proof. For part (a), fix node u and let $r = r_u(\epsilon\, \delta^\alpha)$. Let B be the open ball around u of radius r/δ.

To prove the lemma, it suffices to prove that there exist at least $(1 - \epsilon)n$ nodes v such that $d_{uv} \geq r/\delta$. Equivalently, we show $|B| \leq \epsilon n$. Indeed, by Lemma 2.14 for any $x > 0$ we have

$$|B_u(r/\delta - x)| \leq (\tfrac{1}{\delta})^\alpha |B_u(r - x/\delta)| \leq (\tfrac{1}{\delta})^\alpha |B_u(r)| \leq \epsilon n.$$

It follows that $|B| = \lim_{x \to +0} |B_u(r/\delta - x)| \leq \epsilon n$. This proves part (a).

For part (b), let us fix a $(\epsilon\, 2^\alpha)$-long edge (u, v), let $d = d_{uv}$ and $r = r_u(\epsilon\, 2^\alpha)$. Note that without loss of generality $d \geq r$. In particular, $d \geq r_u(\epsilon)$. It remains to show that $d \geq r_v(\epsilon)$. Suppose not. Then the ball $B_v(d)$ contains less than ϵn nodes. By Lemma 2.14 it follows that

$$(\epsilon\, 2^\alpha)\, n \leq |B_u(r)| \leq |B_v(d)|\, (\tfrac{d+r}{d})^\alpha < (\epsilon\, 2^\alpha)\, n,$$

contradiction. This proves part (b).

For part (c), let $r = r_u(\epsilon)$. Note that by Lemma 2.14 for any $y > 0$ we have

$$|B_u(r\, x^{1/\alpha})| \leq |B_u(r - y)| \left(\tfrac{r\, x^{1/\alpha}}{r - y}\right)^\alpha < \epsilon x n \left(\tfrac{r}{r - y}\right)^\alpha.$$

By taking the limit $y \to +0$ we have $|B_u(r\, x^{1/\alpha})| \leq (\epsilon x)n$, which proves part (c). □

Proof of Theorem 3.30 on the nearest-beacon embedding: Let us fix $\epsilon > 0$ and set $\epsilon^* = \epsilon/4^\alpha$. Suppose we choose $k = O(\frac{1}{\epsilon^*}\log\frac{1}{\epsilon})$ beacons uniformly at random. If the constant in $O(\cdot)$ is sufficiently large, then with probability close to 1 it is the case that for at least $(1-\epsilon)n$ nodes u, there is a beacon among the $\epsilon^* n$ nodes closest to u.

Now Lemma 3.33(a) there exists an ϵ-dense set of node pairs (u,v) such that $r^+_{uv}(\epsilon^*) \leq d_{uv}/4$ and moreover there are beacons among the $\epsilon^* n$ nodes closest to u and among the $\epsilon^* n$ nodes closest to v. Let b_u and b_v be the beacons closest to u and v, respectively. It follows that $d(b_u, b_v) = \Theta(d_{uv})$. So, letting d^* be the distance in the nearest-beacon embedding, we have

$$\Omega(d_{uv}) \leq d^*(u,v) = d^*(b_u, b_v) \leq O(d_{uv}\,\log k)$$

(without loss of generality we assume that we embed the beacons using a non-contracting embedding). □

Proof of Theorem 3.31 on gracefully degrading distortion: Recall that Bourgain's algorithm uses random sets S_{ij} of size 2^i, for each $i \in [\log n]$ and $j \in [k]$, $k = O(\log n)$. Denote the contribution of the set S_{ij} by $x_{ij} = |d(u, S_{ij}) - d(v, S_{ij})|$. For normalization purposes divide all coordinates by $k^{1/p}$, so that the embedded uv-distance is

$$f_p(u,v) = \left(\tfrac{1}{k}\textstyle\sum_{ij} x^p_{ij}\right)^{1/p}.$$

For simplicity consider the case $p = 1$ first.

Fix $\epsilon \in (0,1]$ and let $x = \log\frac{1}{\epsilon}$. Let us consider an $(\epsilon\, 2^\alpha)$-long node pair (u,v); let d and d^* be the true and embedded (u,v)-distances, respectively. By the original Bourgain's proof we have $d^* \geq \Omega(d)$. Since $x_{ij} \leq d$, the Bourgain's upper bound is $d^* = O(d \log n)$. Here we improve it to $d^* = O(d)(\alpha + x)$ by showing that

$$\sum_{i>x}\sum_j x_{ij} \leq O(dk\alpha). \tag{3.4}$$

Once we show (3.4), it would follow that distortion on edge (u,v) is at most $O(\alpha + x)$, which for $\epsilon^* = \epsilon\, 2^\alpha$ is at most $O(\alpha + \log\frac{1}{\epsilon^*})$ as required by the theorem. Therefore it remains to prove (3.4).

Note that by Lemma 3.33(b) we have $d_{uv} \geq r^+_{uv}(\epsilon)$. Fix $i > x$. Let $\beta = 2^{1/\alpha}$ and $t = \frac{i-x}{2}$. Let X_{ju} be a 0-1 random variable that is equal to 1 if and only if $d(u, S_{ij}) > d\beta^{-t}$.

Claim 3.34. $\Pr[X_{ju} = 1] \leq \exp(-2^t)$.

Proof. Consider l such that $i \geq l \geq x$ and let $r = r_u(2^{-l})$. By Lemma 3.33(c) we have

$$d > r_u(2^{-x}) = r_u(2^{-l}\, 2^{l-x}) \geq \beta^{l-x}\, r.$$

So $d\,\beta^{x-l} \geq r$. Therefore

$$\Pr\left[d(u, S_{ij}) > d\,\beta^{x-l}\right] \leq \Pr[S_{ij} \text{ misses } B_u(r)] = \left(1 - 2^{-l}\right)^{2^i} < \exp\left(-2^{i-l}\right).$$

The claim follows if we take $l = \frac{i+x}{2}$. □

We would like to upper-bound $\sum_j X_{ju}$ by a constant times the expectation, but for large enough t the expectation is too small to give small enough failure probability via Chernoff bounds. However, if we give up a factor of $2^{2^t}/2^t$, then the Chernoff bound (Lemma 3.35 with $l = 2^t$) gives $\sum_j X_{ju} = O(k2^{-t})$ with a sufficiently small failure probability to make sure that this happens for all u simultaneously.

Note that $x_{ij} > 2d\beta^{-t}$ only if $Y_j = 1$, where $Y_j = X_{ju} \vee X_{jv}$. So

$$\sum_j x_{ij} \leq O(d)\sum_j(\beta^{-t} + Y_j) = O(dk)(\beta^{-t} + 2^{-t}).$$

Summing this over all $i > x$ we obtain the desired upper bound (3.4) since $\frac{1}{1-1/\beta} = O(\alpha)$.

To extend this theorem to a general $p \leq 1$ we need a more complicated calculation than the one in [LLR95]. As before, consider a fixed $i > x$. Let S be the set of all j such that $Y_j = 1$. Recall that with high probability it is the case that for all pairs (u, v) the size of S is at most $O(k2^{-t})$. Therefore

$$\begin{aligned}
\sum_j x_{ij}^p &= \sum_{j\in S} x_{ij}^p + \sum_{i\notin S} x_{ij}^p \leq |S| f_p + k(2d\beta^{-t})^{1/p} \\
&= O(k)(2d)^p(2^{-t} + \beta^{-tp}) \\
\frac{1}{k}\sum_{i>x}\sum_j x_{ij}^p &\leq (2d)^p \sum_{i>x} O(\beta^{-ip} + 2^{-i}) \leq O\left(\frac{(2d)^p}{1-\beta^{-p}}\right) \\
&\leq (2d)^p O(\alpha/p) \\
f_p(u,v) &= \left(\frac{1}{k}\sum_{i>x}\sum_j x_{ij}^p + \frac{1}{k}\sum_{i\leq x}\sum_j x_{ij}^p\right)^{1/p} \\
&\leq O(d)(x + \alpha/p)^{1/p}
\end{aligned}$$

For a lower bound, let us denote $l = x + 2s$. We claim that $r_{uv}^+(2^{-l}) \leq d/4$. Indeed, by Lemma 2.14

$$\epsilon n \leq |B_u(r_u(\epsilon))| \leq 4^\alpha\, |B_u(r_u(\epsilon)/4)| \leq 4^\alpha\, |B_u(d/4)|.$$

So ball $B_u(d/4)$ contains at least $(\epsilon/4^\alpha)n$ nodes, so $r_u(\epsilon/4^\alpha) \leq d/4$. We prove this for node v similarly. Claim proved.

Now, in the proof of Theorem 3.6 (see (3.2)) we essentially show that $\sum_{i\leq l}\sum_j x_{ij} \geq \Omega(kd)$. Therefore,

$$\begin{aligned}
f_p(u,v) &\geq \left(\frac{1}{k}\sum_{i\leq l}\sum_j x_{ij}^p\right)^{1/p} = l^{1/p}\left(\frac{1}{kl}\sum_{i\leq l}\sum_j x_{ij}^p\right)^{1/p} \geq l^{1/p}\left(\frac{1}{kl}\sum_{i\leq l}\sum_j x_{ij}\right) \\
&\geq \Omega(d)(x+\alpha)^{1/p-1}
\end{aligned}$$

So the total (two-sided) distortion is at most $O(x + \alpha)$ as required. □

In the proof of the above theorem, we have used the following version of Chernoff bounds:

Lemma 3.35. *Let X_j, $j \in [8\log n]$ be independent 0-1 random variables such that $\Pr[X_j = 1] = e^{-l}$ where $l > 16$. Then $\sum X_j < \frac{8}{l}\log n$ with probability at least $1 - n^{-4}$.*

Proof. Let $X = \sum X_j$ and $\mu = E(X)$. Let $1 + \delta = e^l/l$. Then using Chernoff Bounds we get

$$\Pr[X > 8l^{-1}\log n] = \Pr[X > (1+\delta)\mu] < e^{-\mu}\left(\frac{e}{1+\delta}\right)^{(1+\delta)\mu} < \left(\frac{(el)^{1/l}}{e}\right)^{8\log n} < \frac{1}{n^4}$$

since $(el)^{1/l} < \sqrt{e}$ for any $l > 16$. □

Proof of Theorem 3.32 proceeds similarly to that of Theorem 3.31; we omit the details.

Type of Embedding	Our Lower Bound	Original Example
All metrics into ℓ_p, $p \geq 1$	$\Omega(\frac{1}{p})(\log\frac{1}{\epsilon})$	Constant-degree expanders [Mat97]
$\mathcal{F}$ into ℓ_p, $p \in (1,2]$	$\Omega(1-p)\sqrt{\log 1/\epsilon}$	Laakso fractal [LMN04]
Growth-constrained ℓ_1-metrics into ℓ_1^d	$\Omega(\sqrt{\log_d 1/\epsilon})$	Laakso fractal [LMN04]
$\mathcal{F}$ into distributions of dominating trees	$\Omega(\log\frac{1}{\epsilon})$	$n \times n$ grid [AKPW95]
All metrics into tree metrics	$\Omega(1/\sqrt{\epsilon})$	n-cycle [RR98, Gup01]
ℓ_2^{2m+1} into ℓ_2^{2m}	$\Omega(1/\sqrt{\epsilon})^{1/m}$	[Mat90]

Here $\mathcal{F}$ is the family of doubling metrics that are shortest-paths metrics of planar graphs.
In the last two lines, bounds for ϵ-uniform slack can be obtained by replacing $\sqrt{\epsilon}$ by ϵ.

Table 3.1: Embeddings with slack ϵ: lower bounds on distortion

3.6 Lower bounds on embeddings with slack

In this section, we describe a general technique to derive lower bounds for ϵ-slack embeddings from lower bounds for ordinary embeddings. For simplicity of exposition, we will first give a concrete example proving lower bounds for ϵ-slack embeddings into ℓ_p (which will follow from a lower bound for embedding expanders into ℓ_p [Mat97]). Then we provide the general technique; the bounds obtained by this technique are given in Table 3.1. Let us mention that allowing arbitrary *expansions* is crucial to our results: if we insisted that *none of the pairwise distances should increase*, the lower bound of $\Omega(\frac{1}{p}\log n)$ distortion [Mat97] for embeddings into ℓ_p holds even with ϵ-slack (see Section 3.6.2 for more details).

Theorem 3.36. *For an arbitrarily small positive ϵ there exists a finite metric on arbitrarily many nodes that requires distortion $\Omega(\frac{1}{p}\log\frac{1}{\epsilon})$ to embed into ℓ_p, $p \geq 1$ with ϵ-slack.*

Proof. Given an ϵ such that $0 < \epsilon \leq 1/12$, let $k = 1/(3\sqrt{\epsilon})$. Suppose every finite metric can be embedded into ℓ_p with distortion D and ϵ-slack.

Consider a constant degree expander graph H on k vertices. Let (H, d) be the shortest path metric defined by H. For each vertex $s \in H$, let L_s be a path containing n/k vertices. Attach the path L_s to s at one of its endpoints. The length of each edge of L_s is small enough so that if δ is the length of path L_s, then $\delta \cdot D \leq 1/2$. Let the new graph be G and the shortest path metric defined on it be (G, d). We now prove that if (G, d) can be embedded into ℓ_p with distortion D and ϵ-slack, then H can be embedded into ℓ_p with distortion $4D$ *without any slack.*

Let $\varphi : G \rightarrow \ell_p$ be the embedding of (G, d) into ℓ_p with distortion D and ϵ-slack. Let E denote the set of *ignored* pairs, i.e. let us assume that the complement of E incurs distortion at most D. Note that ϵ-slack means that $|E| \leq \epsilon n^2 2$. We delete all the vertices that participate in more than $\sqrt{\epsilon}n$ pairs in E. By a simple counting argument, at most $\sqrt{\epsilon}n$ vertices of G can be deleted. Therefore, at least one point from each path survives. For each $s \in H$, let v_s denote a survived vertex from the path L_s. We define an embedding ψ of H into ℓ_p as $\psi(s) = \varphi(v_s)$.

We now bound the distortion of the embedding ψ by $4D$. Let x, y be two vertices in H. Then v_x and v_y are the survivors in L_x and L_y respectively. Note that v_x and v_y participate in at most $\sqrt{\epsilon}n$ pairs in E. Since $|L_y| = 3\sqrt{\epsilon}n$, it follows that there is another survivor $t \in L_y$ such that neither $\{t, v_x\}$ nor $\{t, v_y\}$ is in E. Since the distortion of the map φ is D, we can assume that for edge $(u, v) \notin E$,

$$d(u,v) \leq \|\varphi(u) - \varphi(v)\|_p \leq D \cdot d(u,v).$$

Now we can bound $\psi(xy) := \|\psi(x) - \psi(y)\|_p$ as follows:

$$\begin{aligned}
\psi(xy) &= \|\varphi(v_x) - \varphi(v_y)\|_p \\
&\leq \|\varphi(v_x) - \varphi(t)\| + \|\varphi(t) - \varphi(v_y)\| \\
&\leq D\,(d(v_x, t) + d(t, v_y)) \\
&\leq D\,(1 + 3\delta)\,d(x, y) \leq 2D\,d(x, y).
\end{aligned}$$

Similarly,

$$\begin{aligned}
\psi(xy) &\geq \|\varphi(v_x) - \varphi(t)\|_p - \|\varphi(t) - \varphi(v_y)\|_p \\
&\geq d(v_x, t) - Dd(t, v_y) \geq (1 - D\delta)d(x, y) \\
&\geq d(x, y)/2.
\end{aligned}$$

Hence $\frac{1}{2}d(u, v) \leq \psi(uv) \leq 2D \cdot d(u, v)$, and so ψ is a map from H to ℓ_p with distortion $4D$.

To finish the proof of the theorem, we note that a constant-degree expander on k vertices requires $\Omega(\log k/p)$ distortion to embed into ℓ_p [Mat97]. □

3.6.1 General lower-bounding technique

The technique used in Theorem 3.36 of starting with a $O(1)$-degree expander H_k on k vertices, replacing each vertex with a path on n/k vertices to get G, and for suitable $k \approx O(1/\sqrt{\epsilon})$ arguing that ϵ-slack embeddings of G_n give us slack-less embeddings of H_k with (roughly) the same distortion is quite general. In fact, we use it to obtain lower bounds on *both the distortion and dimensions of embeddings* into ℓ_p from similar lower bounds for slack-less embeddings; similar results can be obtained for embeddings into trees, or distributions of trees. We summarize these results in Table 3.1.

Theorem 3.37. *Suppose for each k there exists a k-node metric H_k such that any embedding of H_k into ℓ_p with $L(k)$ dimensions has distortion at least $D(k)$. Then for an arbitrarily small positive ϵ there exist finite metrics M, M^* on arbitrarily large number of nodes such that:*

(a) any embedding of M into ℓ_p with $L(\frac{1}{3\sqrt{\epsilon}})$ dimensions has ϵ-slack distortion $\Omega(D(\frac{1}{3\sqrt{\epsilon}}))$.

(b) any embedding of M^* into ℓ_p with $L(\frac{1}{3\epsilon})$ dimensions has ϵ-uniform slack distortion $\Omega(D(\frac{1}{3\epsilon}))$.

Moreover, if metrics $\{H_k\}$ are planar (resp. K_r-minor-free, doubling, ℓ_p^d) then so are M and M^*.

Note that this result can be used to translate, for instance, the lower bound for dimensionality reduction in ℓ_1 (Theorem 2.8) into the realm of ϵ-slack as well.

Similarly, we provide a lower bound theorem for (probabilistic) embeddings into trees:

Theorem 3.38. *Suppose for each k there exists a k-node metric H_k such that any (probabilistic) embedding of H_k into trees has distortion at least $D(k)$. Then for an arbitrarily small positive ϵ there exist finite metrics M, M^* on arbitrarily large number of nodes such that:*

(a) any (probabilistic) embedding of M into trees has ϵ-slack distortion $\Omega(D(\frac{1}{3\sqrt{\epsilon}}))$.

(b) any (probabilistic) embedding of M^* into trees has ϵ-uniform slack distortion $\Omega(D(\frac{1}{3\epsilon}))$.

Moreover, if metrics $\{H_k\}$ are planar (resp. K_r-minor-free, doubling, ℓ_p^d) then so are M and M^*.

For instance, we can now derive a lower bound of $\Omega(1/\sqrt{\epsilon})$ on the distortion incurred when embedding the n-cycle into a single tree.

The proofs of the two above theorems are based on the following lemma:

Lemma 3.39 (Master Lemma). *Suppose H is a metric on k points and $\mathcal{T}$ is a collection of metrics on k points, such that any embedding of H into $\mathcal{T}$ incurs a distortion at least D. Suppose $\mathcal{S}$ is a collection of metrics such that every subset of k points in each metric in $\mathcal{S}$ embeds into $\mathcal{T}$ with distortion at most ρ. Setting $\epsilon = 1/9k^2$, there exist arbitrarily large metrics that embed into $\mathcal{S}$ with ϵ-slack distortion $\Omega(\frac{D}{\rho})$.*

Remark. In order to obtain lower bounds for *ϵ-uniform slack* embeddings instead of ϵ-slack embeddings, we need to set $\epsilon = 1/3k$ instead of $\epsilon = 1/9k^2$; the rest of the proof remains essentially unchanged.

Before we prove Lemma 3.39, let us show how to derive the above results from it.

Proof of Theorem 3.37: Suppose $\{H_k\}$ is the given family of metrics. Let us fix a large enough k such that $\epsilon = 1/9k^2$ is small enough. Now in Lemma 3.39, let us set H to be H_k and $\mathcal{T}$ to be the collection of metrics with k points in ℓ_p with at most $L(k)$ dimensions. Hence, H embeds into $\mathcal{T}$ with distortion at least $D(k) = D(\frac{1}{3\sqrt{\epsilon}})$. We set $\mathcal{S}$ to be the family of metrics in ℓ_p with at most $L(k) = L(\frac{1}{3\sqrt{\epsilon}})$ dimensions. It follows that any subset of k points in any metric in $\mathcal{S}$ embeds into $\mathcal{T}$ with distortion 1. Hence, we conclude that there exists a family of metrics, each of which embeds into ℓ_p with at most $L(\frac{1}{3\sqrt{\epsilon}})$ dimensions with ϵ-slack distortion at least $\Omega(D(\frac{1}{3\sqrt{\epsilon}}))$. □

The application of Lemma 3.39 to prove the lower bounds for embeddings into trees is very similar; we sketch it here to emphasize the general patterns, as well as the slight changes required.

Proof of Theorem 3.38: Again, we large enough k, and set $\epsilon = 1/9k^2$. As before, H is set to be H_k. We set $\mathcal{T}$ to be the family of tree metrics on k points (or distribution of tree metrics on k points). Again, H embeds into $\mathcal{T}$ with distortion at least $D(k) = D(\frac{1}{3\sqrt{\epsilon}})$. We set $\mathcal{S}$ to be the family of tree metrics (or distribution of tree metrics). Note that by a result of Gupta [Gup01], any subset of k points in any metric in $\mathcal{S}$ embeds into $\mathcal{T}$ with distortion at most 8. Now the result of Theorem 3.38 follows from Lemma 3.39 as before. □

Let us now prove the Lemma 3.39: first we show how to construct a family of metrics with the desired properties. Suppose $H = (S, d)$ is a metric such that $|S| = k$. Moreover, H embeds into $\mathcal{T}$ with distortion at least D. Without loss of generality, assume that the pairwise distance in H is at least 1. For each n that is a multiple of $3k$, we define a metric $\hat{H}$ with n points in the following way. These would be the family of metrics that exhibits the lower bound for slack embeddings.

Consider a uniform line metric with point set L of size $\frac{n}{k}$ such that the two terminal points are at distance δ away from each other, where δ is small and whose value will be specified later. For each $s \in S$, we identify s with a terminal point of a copy L_s of the line metric L. We call the augmented metric $\hat{H} = (V, d)$ with point set $V = \cup_{s\in S} L_s$. If H is already in some host space X, we just need the condition that for each $s \in S$, we can embed a copy of L of length δ isomorphically into X that identifies one end point with s. Common metric spaces like ℓ_p certainly satisfy this condition. (Note that to avoid too many symbols, we use d for the various metrics.) Hence, for $u \in L_x$ and $v \in L_y$, $|d(u, v) - d(x, y)| \leq 2\delta$.

Proposition 3.40. *Let H and $\hat{H}$ be metrics defined as above. Then,* **(a)** *if H is a metric induced by a K_r-minor free graph, then so is $\hat{H}$, and* **(b)** *if H is a doubling metric, then so is $\hat{H}$.*

The next lemma states a crucial property of the edges that are ignored by any ϵ-slack embedding.

Lemma 3.41. *Suppose an ϵ-slack embedding of some metric (V, d) ignores the set of edges E. Then, there exists a subset $T \subseteq V$ of size at least $(1 - \sqrt{\epsilon})n$ such that each vertex in T intersects with at most $\sqrt{\epsilon}n$ edges in E.*

Proof. It suffices to show that it is impossible to have a subset $S \subseteq V$ of size greater than $\sqrt{\epsilon}n$ such that each vertex in S intersects more than $\sqrt{\epsilon}n$ edges in E. Otherwise, the total number of edges ignored would be greater than $(\sqrt{\epsilon}n)^2/2 > \epsilon n^2/2 > \epsilon\binom{n}{2}$. □

Note that for an ϵ-*uniform slack* embedding, the number of ignored edges incident on any node is at most ϵn by definition; this is one place in the proof which changes when considering uniform slack.

The following lemma implies Lemma 3.39:

Lemma 3.42. *Let $H = (S, d)$ be a metric on k points. Suppose $\mathcal{T}$ and $\mathcal{S}$ are families of metrics such that H embeds into $\mathcal{T}$ with distortion at least D, and every subset of k points in each metric in $\mathcal{S}$ embeds into $\mathcal{T}$ with distortion at most ρ.*

Suppose δ is small enough such that $(\frac{D}{4\rho} + 2)\delta \leq \frac{1}{2}$. Let $\hat{H} = (V, d)$ be the metric be defined as above. Let $\epsilon := 1/9k^2$. Then, $\hat{H}$ embeds into $\mathcal{S}$ with ϵ-slack distortion at least $D/4\rho$.

Proof. Suppose, on the contrary, φ is an embedding of $\hat{H}$ into $\mathcal{S}$ with ϵ-slack distortion $R < D/4\rho$ that ignores the set E of edges. Then by Lemma 3.41, there exists a subset T of V such that $|T| \geq (1 - \sqrt{\epsilon})n$ and for all $v \in T$, v intersects at most $\sqrt{\epsilon}n$ edges in E.

For each $s \in S$, the set L_s contains $\frac{n}{k} = 3\sqrt{\epsilon}n$ points and hence there exists some point in $T \cap L_s$, which we call v_s. We define an embedding ψ of H into $\mathcal{S}$ given by $\psi(s) := \varphi(v_s)$. We next bound the distortion of the embedding ψ. Let $x, y \in S$. Since v_x and v_y are in T, each of them has at most $\sqrt{\epsilon}n$ neighbors. Observing that $|L_y| = 3\sqrt{\epsilon}n$, it follows that there exists a point $t \in L_y$ such that neither $\{v_x, t\}$ nor $\{v_y, t\}$ is contained in E. We can assume that for $\{u, v\} \notin E$, $d(u, v) \leq ||\varphi(u) - \varphi(v)|| \leq Rd(u, v)$.

Hence, it follows that

$$\begin{aligned}
\|\psi(x) - \psi(y)\| =& \|\varphi(v_x) - \varphi(v_y)\| \\
\leq& \|\varphi(v_x) - \varphi(t)\| + \|\varphi(t) - \varphi(v_y)\| \\
\leq& R(d(v_x, t) + d(t, v_y)) \leq R(d(x, y) + 3\delta) \\
\leq& R(1 + 3\delta)d(x, y) \leq 2Rd(x, y),
\end{aligned}$$

and similarly,

$$\begin{aligned}
\|\psi(x) - \psi(y)\| \geq& \|\varphi(v_x) - \varphi(t)\| - \|\varphi(t) - \varphi(v_y)\| \\
\geq& d(v_x, t) - Rd(t, v_y) \geq d(x, y) - 2\delta - R\delta \\
\geq& (1 - (R + 2)\delta)d(x, y) \geq d(x, y)/2,
\end{aligned}$$

where the last inequality follows from the fact that $(R + 2)\delta \leq 1/2$. It then follows that ψ embeds H into $\mathcal{S}$ with distortion at most $4R$. However, since any metric in $\mathcal{S}$ embeds into $\mathcal{T}$ with distortion at most ρ, it follows that H embeds into $\mathcal{T}$ with distortion at most $4\rho R < D$, from which we obtain the desired contradiction. □

3.6.2 Lower bounds for contracting embeddings

Let us consider contracting embeddings with slack. Formally, a contracting embedding has distortion D with ϵ-slack if no pairwise distance expands and all but ϵ-fraction of the pairs contract by no more than D. We show that such embeddings incur an $\Omega(\log n)$ distortion in order to embed constant-degree expander graphs into ℓ_p, $p \geq 1$.

Theorem 3.43. *For the shortest-paths metric of a bounded-degree expander on n vertices, distortion of any contracting embedding into ℓ_p, $p \geq 1$ is $\Omega(\frac{1}{p}\log n)$ even if we allow slack $\epsilon < \frac{1}{2}$.*

Proof. Let $G = (V, E)$ be a bounded-degree expander on n vertices, and let ρ denote its shortest path metric. Let φ be a contracting embedding of this metric to ℓ_p, $p \geq 1$ with distortion D and slack $\epsilon < \frac{1}{2}$. Let σ denote the metric on ℓ_p; to simplify the notation, we will denote $\varphi(V) \subseteq \ell_p$ by V. Define

$$\begin{aligned} R(\sigma) &= \sqrt{\sigma^2(V \times V)/\sigma^2(E)}\ , \text{ where} \\ \sigma^2(S) &= \textstyle\sum_{(x,y)\in S} \sigma(x,y)^2 \text{ for any set } S \subseteq V \times V. \end{aligned}$$

First we show that $R(\sigma) \leq O(\sqrt{n})$. The proof is exactly the same as that of Theorem 15.5.1 in Matousek [Mat02b] and works despite the fact that we allow $\varepsilon \cdot n^2$ pairwise distances to be as low as 0. Note that

$$\sigma^2(E) = \textstyle\sum_{(x,y)\in E} \sigma(x,y)^2 \leq \sum_{(x,y)\in E} \rho(x,y)^2 = O(n)$$

Now, we bound $\sigma^2(V \times V)$ from below. If all n^2 pairs were contracted by at most D, then we would get

$$\sigma^2(V \times V) \geq \textstyle\sum_{(u,v)} \left(\frac{\rho(u,v)}{D}\right)^2 \geq \frac{n^2 \log^2 n}{D^2}$$

However, we need to take into account the fact that $\varepsilon \cdot n^2$ pairs of vertices could have distance 0 between them. Therefore, $\sigma^2(V \times V)$ is at least $(n/D)^2(\log^2 n)$ minus the loss due to the slack. To upper-bound this loss, consider a pair (x, y) of nodes for which the distortion is bigger than D. The pair will contribute 0 instead of $\rho(x, y)/D$. Thus the loss due to the pair (x, y) is at most $(\log n)/D$. Therefore, the total loss due to the slack is at most $\epsilon(n/D)^2(\log^2 n)$. Therefore, since $R(\sigma) \leq O(\sqrt{n})$, it follows that $D = \Omega(\log n)$. □

Chapter 4

Gracefully Degrading Distortion for Decomposable Metrics

In this chapter we prove our result from [ABC$^+$05, CDG$^+$05] on low-dimensional embeddings into ℓ_p, $p \geq 1$ with gracefully degrading distortion:

Theorem 4.1. *Consider a metric (V, d) which admits β-padded decompositions. Then it can be embedded into ℓ_p, $p \geq 1$ with $O(\log^2 n)$ dimensions and gracefully degrading distortion $O(\beta)(\log \frac{1}{\epsilon})^{1/p}$. The embedding procedure is given as a randomized algorithm which succeeds with high probability.*

Furthermore, in Section 4.8 we discuss the extensions to arbitrary metrics (via probabilistic embeddings into trees).

The proof of Theorem 4.1 builds on the well-known embedding algorithms of Bourgain [Bou85] and Linial et al. [LLR95], and combines ideas from Chapter 3 and [Rao99, GKL03, KLMN05] with some novel ones. To the best of our understanding, the embeddings given in the previous papers do not directly give us gracefully degrading distortion, and hence the additional machinery indeed seems to be required.

Let us fix $k = O(\log n)$, where the constant will be specified later. We will construct an embedding $\varphi : V \to \ell_p$ with $7k^2$ dimensions; the coordinates of φ will be indexed by triples $(i, j, l) \in [k] \times [k] \times [7]$.

We will show how to construct the map φ in rest of this section, which has the following conceptual steps. We first define a concrete notion of "distance scales" in Section 4.1, in terms of which we can cast many previous embeddings, and specify the desired properties for the distance scales in our embedding. We then show how to construct the distance scales as well as the claimed embedding φ in Section 4.2, and show that it has gracefully degrading distortion in Section 4.3.

4.1 Distance scales and scale bundles

Our algorithm, just like the algorithms in [Bou85, LLR95, Rao99, GKL03, KSW04, KLMN05, Lee05], operates on distance scales that start around the diameter of the metric, and go all the way down to the smallest distance in the metric. Informally, the embedding φ has block of coordinates for each distance scale, such that if the true uv-distance for some edge uv is within this scale, then the uv-distance in these coordinates of φ is roughly equal to the true distance. These blocks of coordinates are then combined into an embedding that works for all scales simultaneously.

Different embeddings use very different notions of distance scales; in cases like the Rao-style embeddings, there are clear coordinates for each distance that is a power of 2—but in Bourgain-style embeddings, this is not the case. To be able to give a unified picture, let us formally define a *distance scale* f to be a coordinate map $f : V \to \mathbb{R}$. A *scale bundle* $\{f_{ij}\}$ is then a collection of coordinate maps f_{ij}, such that for every fixed index j and node u, the values $f_{ij}(u)$ are *decreasing* with i.

We can now cast and interpret previous embeddings in this language: in the Bourgain-style embeddings [Bou85, LLR95], $f_{ij}(u)$ is the radius of the smallest ball around u containing 2^{n-i} nodes, and hence the cardinality of $\mathbf{B}_u(f_{ij}(u))$ halves as we increase i. In the Rao-style embeddings [Rao99, GKL03], the scales are $f_{ij}(u) = \text{diameter}(V)/2^i$, and hence the distance scales halve as we increase i. The measured descent embedding in [KLMN05] essentially ensures a judicious mixture of the above two properties: as we increase i, the ball $\mathbf{B}_u(f_{ij}(u))$ either halves in radius, or halves in cardinality, whichever comes first.

For our embedding, we need *both* the radius and the cardinality of $\mathbf{B}_u(f_{ij}(u))$ to halve—and hence have to define the scale-bundles accordingly. This would be easy to achieve by itself; however, to give good upper bounds on the embedded distance, we also need each distance scale to be sufficiently smooth, by which we mean that all the distance scales f_{ij} must themselves be 1-Lipschitz. In other words, we want that $|f_{ij}(u) - f_{ij}(v)| \leq d(u, v)$. The construction of the scale bundle $\{f_{ij}\}$ with both halving and smoothness properties turns out to be a bit non-trivial, the details of which are given in the next section.

4.2 The embedding algorithm

Let us construct the embedding for Theorem 4.1. We have not attempted to optimize the multiplicative constant for distortion, having chosen the constants for ease of exposition whilst ensuring that the proofs work.

First we will construct a *scale bundle* $\{f_{ij} : i, j \in [k]\}$. For a fixed j, the maps f_{ij} are constructed by an independent random process, inductively from $i = 0$ to $i = k - 1$. We start with $f_{(0,j)}(\cdot)$ equal to the diameter Φ_d of the metric. Given f_{ij}, we construct $f_{(i+1,j)}$ as follows. Let U_{ij} be a random set such that each node u is included independently with probability $1/|\mathbf{B}_u(4f_{ij}(u))|$. Define $f_{(i+1,j)}(u)$ as the minimum of $d(u, U_{ij})$ and $f_{ij}(u)/2$. This completes the construction of the scale bundle.

To proceed, let us state a lemma that captures, for our purposes, the structure of the metric.

Lemma 4.2. *Consider a metric (V, d) which admits β-padded decompositions. Then for any 1-Lipschitz coordinate map f there is a randomized embedding g into ℓ_p, $p \geq 1$ with $t = 6$ dimensions so that*

(a) each coordinate of g is 1-Lipschitz and upper-bounded by f; and

(b) if $f(u)/d_{uv} \in [\frac{1}{4}; 4]$ for some edge uv then, with probability $\Omega(1)$, $\|g(u)-g(v)\|_p \geq \Omega(d_{uv}\, t^{1/p}/\beta)$.

Section 4.4 and Appendix 4.7 contain two different proofs of this lemma; the first one uses padded decomposition techniques from [GKL03, KLMN05], and the other uses some Bourgain-style ideas [Bou85, LLR95] which we believe are novel and possibly of independent interest.[1]

Fix a pair $i, j \in [k]$. Apply Lemma 4.2 to the map f_{ij} and obtain a 6-dimensional embedding; denote these 6 coordinates as $g_{(i,j,l)}$, $1 \leq l \leq 6$. Let W_{ij} be a random set such that each node u is included independently with probability $1/|\mathbf{B}_u(f_{ij}(u)/2)|$. Define $g_{(i,j,0)}(u)$ as the minimum of $f_{ij}(u)$ and $d(u, W_{ij})$. Finally, we set $\varphi_{(i,j,l)} = k^{-1/p}\, g_{(i,j,l)}$.

Lemma 4.3. *The maps f_{ij}, g_{ij} and $\varphi_{(i,j,l)}$ are 1-Lipschitz.*

Proof. Indeed, $f_{(0,j)}$ is 1-Lipschitz by definition, and the inductive step follows since the $\min$ of two 1-Lipschitz maps is 1-Lipschitz. For the same reason, the maps $g_{(i,j,l)}$ are 1-Lipschitz as well, and therefore so are the maps $\varphi_{(i,j,l)}$. □

[1] More precisely, the second proof is for the important special case when β is the doubling dimension. In this proof the target dimension becomes $t = O(\beta \log \beta)$, which results in target dimension $O(\log^2 n)(\beta \log \beta)$ in Theorem 4.1.

Since $k = O(\log n)$, it immediately follows that the embedded distance is at most $O(\log n)$ times the true distance. In the next section, we will prove a sharper upper bound of $O(d_{uv})(\log \frac{1}{\epsilon})^{1/p}$ for any ϵ-long edge uv, and a lower bound $\Omega(d_{uv}/\beta)$ for any edge.

4.3 Analysis

Let us prove Theorem 4.1 by giving bounds on the stretch and contraction of the embedding φ. The following definition will be useful: for a node u, an interval $[a, b]$ is *u-broad* if a or b is equal to d_{uv} for some v, $a \le b/4$ and $|\mathbf{B}_u(a)| \le \frac{1}{32}|\mathbf{B}_u(b)|$.

Let us state two lemmas that capture the useful properties of the maps f_{ij} and $g_{(i,j,0)}$, respectively; note that these properties hold for an arbitrary input metric. The proofs are deferred to Section 4.6.

Lemma 4.4. *With high probability it is the case that:*

(a) for any 1-Lipschitz maps $f'_{ij} \le f_{ij}$ and any ϵ-long edge uv $\sum_{ij} f'_{ij}(uv) \le O(kd_{uv} \log \frac{1}{\epsilon})$.

(b) for each node u, each u-broad interval contains values f_{ij} for $\Omega(k)$ different j's.

Lemma 4.5. *Fix edge uv and indices ij; let $R = f_{ij}(u)$ and $d = d_{uv}$. Given that $R \ge 4d$ and $|\mathbf{B}_u(d/4)| = c\,|\mathbf{B}_u(R)|$, the event $g_{(i,j,0)}(uv) \ge \Omega(d)$ happens with conditional probability $\Omega(c)$.*

Proof of Theorem 4.1: Fix an ϵ-long edge uv and let $d = d_{uv}$. Since $g_{(i,j,l)} \le f_{ij}$ for each l, by Lemma 4.4a the embedded uv-distance is upper-bounded by $O(d \log \frac{1}{\epsilon})$ for $p = 1$; the same argument gives an upper bound of $O(d)(\log \frac{1}{\epsilon})^{1/p}$ for $p > 1$.

It remains to lower-bound the embedded uv-distance by $\Omega(d/\beta)$, where β is the parameter in Theorem 4.1 and Lemma 4.2. Denote by $g_{ij}(uv)$ the total ℓ_p-distance between u and v in the coordinates $g_{(i,j,l)}$, $l \ge 1$. Denote by $\mathcal{E}_{ij}$ the event that $g_{(i,j,0)}(uv)$ or $g_{ij}(uv)$ is at least $\Omega(d/\beta)$. It suffices to prove that with high probability events $\mathcal{E}_{ij}$ happen for at least $\Omega(k)$ (i,j)-pairs. We consider two cases, depending on whether $\rho_u(\epsilon/32) \ge d/4$.

Case (a). If $\rho_u(\epsilon/32) \ge d/4$ then the interval $I = [d/4; d]$ is u-broad, so by Lemma 4.4b there are $\Omega(k)$ different j's such that $f_{ij}(u) \in I$ for some i. By Lemma 4.2 and Chernoff bounds (Lemma 2.1a) for $\Omega(k)$ of these ij pairs we have $g_{ij}(uv) \ge \Omega(d/\beta)$, case (a) complete.

Case (b). Assume $\rho_u(\epsilon/32) < d/4$; consider the interval $I = [d; \max[4d, \rho_u(32\epsilon)]]$. We claim that

$$\Pr\left[\mathcal{E}_{ij} \mid f_{ij}(u) \in I\right] \ge \Omega(1), \text{ for each } (i,j)\text{-pair.} \tag{4.1}$$

Indeed, fix ij and suppose $f = f_{ij}(u) \in I$. There are two cases, $f \in [d;\, 4d]$ and $f \in (4d;\, \rho_u(32\epsilon)]$. In the first case by Lemma 4.2 $g_{ij}(uv) \ge \Omega(d/\beta)$ with conditional probability at least $\Omega(1)$. In the second case

$$|\mathbf{B}_u(d/4)| \ge \epsilon n/32 \ge 2^{-10}\,(32\epsilon n) \ge 2^{-10}\,|\mathbf{B}_u(f)|,$$

so by Lemma 4.5 $g_{(i,j,0)}(uv) \ge \Omega(d)$ with conditional probability $\Omega(1)$. This proves (4.1). Since the interval I is u-broad, by Lemma 4.4b there are $\Omega(k)$ different j's such that $f_{ij}(u) \in I$ for some i. Since for different j's the events in (4.1) are independent, case (b) follows by Chernoff bounds (Lemma 2.1a). □

4.4 Analysis: proof of Lemma 4.2

In this section we use padded decomposition techniques from [GKL03, KLMN05] to prove Lemma 4.2. Let us recall the definitions of a *padded decomposition* and a *decomposition bundle* from [GKL03, KLMN05].

Definition 4.6. Given a finite metric space (V, d), a positive parameter $\Delta > 0$ and a mapping $\beta : V \to \mathbb{R}$, a Δ-bounded *β-padded decomposition* is a distribution Π over partitions of V such that the following conditions hold:

(a) For each partition P in the support of Π, the diameter of every cluster in P is at most Δ.

(b) If P is sampled from Π, then each ball $\mathbf{B}_x(\Delta/\beta(x))$ is partitioned by P with probability $< \frac{1}{2}$.

Given a function $\beta : V \times \mathbb{Z} \to \mathbb{R}$, a *$\beta$-padded decomposition bundle* on V is a set of padded decompositions $\{\eta(i) : i \in \mathbb{Z}\}$ such that each $\eta(i)$ is a 2^i-bounded $\beta(\cdot, i)$-padded decomposition of V.

If a metric admits a β-padded decomposition bundle such that β is constant, we simply say that this metric *admits β-padded decompositions.*

The randomized construction. Let η be a β-padded decomposition bundle. For each $u \in \mathbb{Z}$, let the decomposition P_u be chosen according to the distribution $\eta(u)$. We denote $P_u(x)$ to be the unique cluster in P_u containing x.

Moreover, for $u \in \mathbb{Z}$, let $\{\sigma_u(C) : C \subseteq V\}$ be i.i.d. unbiased $\{0,1\}$-random variables. Let $T = \{0, 1, \ldots, 5\}$. Let $u(x) := \lceil \log_2 f(x) \rceil$. For each $t \in T$, we define a (random) subset

$$W^t := \{x \in V : \sigma_{u(x)-t}(P_{u(x)-t}(x)) = 0\}, \tag{4.2}$$

from which we obtain $g_t(\cdot) = \min\{d(\cdot, W^t), f(\cdot)\}$.

Bounding the contraction of the embedding. We fix vertices $x, y \in V$ and let $d = d(x, y)$. Consider the embedded distance between them. The aim is to show that under some condition, there exists t such that $|g_t(x) - g_t(y)| \geq \rho d$ happens with constant probability, where ρ depends on the β-padded decomposition bundle.

Lemma 4.7. *Suppose $f(x) \in [\frac{d}{4}, 4d]$ and $t \in T$ is the integer such that $\hat{u} := u(x) - t$ satisfies $2^{\hat{u}} \in [d/8, d/4)$. Let $J := \{-1, 0, 1\}$ and $\rho := \min\{\frac{1}{32\beta(x,u)} : u \in \hat{u} + J\}$. Then the event $|g_t(x) - g_t(y)| \geq \rho d$ happens with probability at least 1/64.*

Proof. Consider the random process that determine the coordinate g_t. We like to show that the union of the following two disjoint events happens with constant probability, which implies our goal. There are two cases:

Case 1 The set W^t contains x but is disjoint with $B_y(\rho d)$.

Case 2 The set W^t contains no points from $B_x(2\rho d)$ but at least one point from $B_y(\rho d)$.

Let us define the following auxiliary events.

- Event $\mathcal{E}_1$ occurs when x is contained in W^t.
- Event $\mathcal{E}_2$ occurs when W^t is disjoint with $B_y(\rho d)$.
- Event $\mathcal{E}_3$ occurs when for all $z \in B_x(2\rho d)$ and $u \in \hat{u} + J$, x and z are in the same cluster in $\eta(u)$.
- Event $\mathcal{E}_4$ occurs if for all $u \in \hat{u} + J$, $\sigma_u(P_u(x)) = 1$.

Observe that the event $\mathcal{E}_1 \cap \mathcal{E}_2$ implies the event in Case 1. Note that given a decomposition $\eta(\hat{u})$, the point x lies in a cluster different from those intersecting $B_y(\rho d)$, because $2^{\hat{u}} < \frac{d}{4} < (1 - \rho)d$. Hence the events $\mathcal{E}_1$ and $\mathcal{E}_2$ are conditionally independent, given $\eta(\hat{u})$; this in turn implies that

$$Pr\left[\mathcal{E}_1 \cap \mathcal{E}_2 | \, \eta(\hat{u})\right] = Pr\left[\mathcal{E}_1 | \, \eta(\hat{u})\right] \; Pr\left[\mathcal{E}_2 | \, \eta(\hat{u})\right] \; = \tfrac{1}{2} \, Pr\left[\mathcal{E}_2 | \, \eta(\hat{u})\right].$$

Since this fact holds for all decompositions $\eta(\hat{u})$, it follows that $Pr[\mathcal{E}_1 \cap \mathcal{E}_2] = \frac{1}{2} Pr[\mathcal{E}_2]$

Observe that the event $\mathcal{E}_3 \cap \mathcal{E}_4 \cap \overline{\mathcal{E}_2}$ implies the event in Case 2. This follows from the fact that $|u(x) - u(z)| \in J$. Since $f(x) \geq \frac{d}{4}$, f is 1-Lipschitz and $d(x,z) \leq 2\rho d \leq \frac{d}{8}$, it follows $f(x)$ and $f(z)$ are within a multiplicative factor of 2 from each other. Hence $u(x)$ and $u(z)$ differ by at most one. Again, given the decompositions $\eta(u)$, $u \in \hat{u} + J$, the event $\mathcal{E}_4$ is independent of the event $\mathcal{E}_3 \cap \overline{\mathcal{E}_2}$. Hence,

$$Pr\left[\mathcal{E}_3 \cap \mathcal{E}_4 \cap \overline{\mathcal{E}_2}\right] = Pr\left[\mathcal{E}_4\right]\ Pr\left[\mathcal{E}_3 \cap \overline{\mathcal{E}_2}\right] = \tfrac{1}{8} Pr\left[\mathcal{E}_3 \cap \overline{\mathcal{E}_2}\right].$$

Finally, it follows that the union of the events in cases 1 and 2 happens with probability at least

$$\tfrac{1}{2}Pr[\mathcal{E}_2] + \tfrac{1}{8}Pr[\mathcal{E}_3 \cap \overline{\mathcal{E}_2}] \geq \tfrac{1}{8}Pr[\mathcal{E}_3 \cap \mathcal{E}_2] + \tfrac{1}{8}Pr[\mathcal{E}_3 \cap \overline{\mathcal{E}_2}] = \tfrac{1}{8}Pr[\mathcal{E}_3].$$

In order to show that $\mathcal{E}_3$ happens with constant probability, we make use of the properties of β-padded decomposition bundle. Since for all $u \in \hat{u} + J$ we have

$$2\rho d \leq 2/32\beta(x,u) \cdot d \leq 2^u/\beta(x,u),$$

it follows that $\mathcal{E}_3$ happens with probability at least 1/8. Therefore, it follows the desired event happens with probability at least 1/64. □

4.5 Analysis: tools from Probability

Here we state some tools from Probability Theory that we will use to prove Lemma 4.4 and Lemma 4.5.

For a random variable X define the *distribution function* $F_X(t) = \Pr[X < t]$. For two random variables X and Y, say Y *stochastically dominates* X (written as $Y \succeq X$, or $X \preceq Y$) if $F_Y(t) \leq F_X(t)$ for all $t \in \mathbb{R}$.

Lemma 4.8. *Consider two sequences of independent random variables, $\{X_i\}$ and $\{Y_i\}$, such that all X_i and Y_i have finite domains and $X_i \preceq Y_i$ for each i. Then for each k we have $\sum_{i=1}^k X_i \preceq \sum_{i=1}^k Y_i$.*

Lemma 4.9. *Consider two sequences of Bernoulli random variables, $\{X_i\}$ and $\{Y_i\}$, such that variables $\{Y_i\}$ are independent and*

$$\Pr[X_i = 1 \mid X_j, j < i] \geq \Pr[Y_i = 1]$$

for each i. Then $\sum_{i=1}^k X_i \succeq \sum_{i=1}^k Y_i$ for each k.

Proof. We first show that for all $t \in [T]$,

$$\Pr\left[\sum_{r=1}^{t} X_r + \sum_{r=t+1}^{T} Y_r \leq m\right] \leq \Pr\left[\sum_{r=1}^{t-1} X_r + \sum_{r=t}^{T} Y_r \leq m\right], \tag{4.3}$$

which would immediately imply the lemma. Observe that for any fixed number a (or in general any random variable that is measurable in the σ-field generated by the random variables $\{X_r : r < t\}$), we have

$$\Pr\left[X_t \leq a | X_r, r < t\right] \leq Pr[Y_t \leq a] = \Pr\left[Y_t \leq a | X_r, r < t\right].$$

Note that the interesting case is when $a \in [0,1)$. The inequality comes from the assumption concerning the conditional probabilities of the sequence $\{X_r\}$, and the equality comes from the fact that Y_t is independent of the sequence $\{X_r\}$.

Since both X_t and Y_t are independent of $\{Y_r : r > t\}$, the above inequality would still hold if we further condition on the random variables $\{Y_r : r > t\}$. Finally, setting $a = m - \sum_{i<t} X_r - \sum_{i>t} Y_r$, which is measurable in the σ-field generated by $J := \{X_r : r < t\} \cup \{Y_r : r > t\}$, we obtain

$$\Pr\left[\sum_{r=1}^{t} X_r + \sum_{r=t+1}^{T} Y_r \leq m \mid J\right] \leq \Pr\left[\sum_{r=1}^{t-1} X_r + \sum_{r=t}^{T} Y_r \leq m \mid J\right].$$

Taking expectation on both sides gives (4.3). □

Lemma 4.10. *Consider a sequence of i.i.d. Bernoulli random variables $\{Y_i\}$ with success probability q. Let Z_r be the number of trials between the $(r-1)$-th success and the r-th success. Then*

$$\Pr\left[\sum_{r=1}^{k} Z_r > 2k/q\right] \leq (0.782)^k. \tag{4.4}$$

Proof. Each Z_r has a geometric distribution with parameter q, so its moment generating function is

$$E\left[e^{tZ_r}\right] = \frac{qe^t}{q-(1-q)e^t}.$$

Let $Z = \sum_{r=1}^{k} Z_r$. Since Z_r's are i.i.d, it follows that $E\left[e^{tZ}\right] = E\left[\prod_r e^{tZ_r}\right] = \left(E\left[e^{tZ_1}\right]\right)^k$.

By Markov inequality for any $t > 0$ we have

$$Pr[Z > 2k/q] = Pr\left[e^{tZ} > e^{2tk/q}\right] \leq E\left[e^{tZ}\right]e^{-2tk/q} \leq \left(\frac{qe^t}{(1-(1-q)e^t)e^{2t/q}}\right)^k.$$

Plugging in $q = 1 - 1/\sqrt{e}$ and $t = q$ we have (4.4). □

4.6 Analysis: maps f_{ij} and $g_{(i,j,0)}$

Here we prove Lemma 4.4 and Lemma 4.5. First we prove part (a) of Lemma 4.4, which is essentially the upper bound on the embedded distance for the case $p = 1$. We start with a local smoothness property of the sets U_{ij}.

Claim 4.11. *Fix $i, j \in [k]$ and an edge uv. Condition on the map f_{ij}, i.e. pause our embedding algorithm right after f_{ij} is constructed; let $r = f_{ij}(u)$. If $d_{uv} \leq r/4$ then*

$$\Pr[v \in U_{ij}] \leq 1/|\mathbf{B}_u(r)| \leq \Pr\left[v \in U_{(i+3,j)}\right].$$

Proof. Let $B = \mathbf{B}_u(r)$. For the RHS inequality, letting $r' = f_{(i+3,j)}(v)$ we have

$$4r' \leq f_{ij}(v)/2 \leq (r + d_{uv})/2 \leq 17r/32,$$

so $d_{uv} + 4r' < r$. It follows that $\mathbf{B}_v(r') \subset B$, so $v \in U_{(i+3,j)}$ with probability $1/|\mathbf{B}_v(4r')| \geq |B|$.

For the LHS inequality, letting $r' = f_{ij}(v)$ we have

$$4r' \geq 4(r - d_{uv}) \geq r + d_{uv},$$

so $B \subset \mathbf{B}_v(4r')$. Therefore $v \in U_{ij}$ with probability $1/|\mathbf{B}_v(4r')| \leq 1/|B|$. □

Fix a node u; for simplicity assume $k = 4k_0 + 1$. Let $B_{ij} = \mathbf{B}_u(f_{ij})$ and let X_{ij} be the indicator random variable for the event that $|B_{(4i+4,j)}| \leq |B_{(4i,j)}|/2$. Note that for a fixed j, the random variables X_{ij} are not independent. However, we can show that given all previous history, the ij-th event happens with at least a constant probability.

Claim 4.12. *For each $i \in [k_0]$, $j \in [k]$ and $q = 1 - e^{-1/2}$ we have $Pr[X_{ij} = 1 \mid f_{lj}, l < i] \geq q$.*

Proof. Indeed, fix ij, let $f = f_{(4i,j)}(u)$ and $f' = f_{(4i+4,j)}(u)$, and let $B = \mathbf{B}_u(r)$ be the smallest ball around u that contains at least $|B_{(4i,j)}|/2$ nodes. Clearly, $X_{ij} = 1$ if and only if $f' \leq r$. By definition of f_{ij}'s we have $f' \leq f/16$, so we are done if $r \geq f/16$. Else by Claim 4.11 any node $v \in B$ included into the set $U_{(4i+3,j)}$ with probability at least $1/2|B|$, so the probability of including at least one node in B into this set (in which case $f' \leq r$) is at least $1 - (1 - 1/2|B|)^{|B|} \geq q$. □

For a random variable X define the *distribution function* $F_X(t) = \Pr[X < t]$. For two random variables X and Y, say Y *stochastically dominates* X (written as $Y \succeq X$, or $X \preceq Y$) if $F_Y(t) \leq F_X(t)$ for all $t \in \mathbb{R}$. Note that if $X \geq Y$ then $X \succeq Y$. Consider a sequence of i.i.d. Bernoulli random variables $\{Y_i\}$ with success probability q. By Claim 4.12 and Lemma 4.9 we have the following:

$$\sum_{i=0}^{t} X_{ij} \succeq \sum_{i=0}^{t} Y_i, \text{ for any } t \in [k_0] \text{ and each } j \in [k]. \tag{4.5}$$

We will use (4.5) to prove the following crucial claim:

Claim 4.13. *Fix $\epsilon > 0$; for each j let T_j be the smallest i such that $f_{ij}(u) \leq \rho_u(\epsilon)$, or k if no such i exists. Then $\sum_j T_j = O(k \log \frac{1}{\epsilon})$ with high probability.*

Proof. Let $\alpha = \lceil \log \frac{1}{\epsilon} \rceil$. Let L_j be the smallest t such that $\sum_{i=0}^{t} X_{ij} \geq \alpha$, or k_0 if such t does not exist; note that $T_j \leq 4L_j$. For the sequence $\{Y_i\}$, let Z_r be the number of trials between the $(r-1)$-th success and the r-th success. Let $A_j = \sum_{r=(j-1)\alpha+1}^{j\alpha} Z_r$ and $Z = \sum_{r=1}^{k\alpha} Z_r$. By (4.5) for any integer $t \in [k_0]$

$$\Pr[L_j > t] = \Pr\left[\sum_{i=0}^{t} X_{ij} < \alpha\right] \leq \Pr\left[\sum_{i=0}^{t} Y_i < \alpha\right] = \Pr\left[\sum_{r=1}^{\alpha} Z_r > t\right] = \Pr[A_1 > t] \tag{4.6}$$

Since $\{A_j\}$ are i.i.d., by (4.6) and Lemma 4.8 it follows that $\sum_j L_j \succeq \sum_j A_j = Z$. Therefore by Lemma 4.10

$$\Pr\left[\sum T_j > 8k\alpha/q\right] \leq \Pr\left[\sum L_j > 2k\alpha/q\right] \leq \Pr[Z > 2k\alpha/q] < (0.782)^{k\alpha},$$

which is at most $1/n^3$ when $k = O(\log n)$ with large enough constant. □

Now we have all tools to prove Lemma 4.4a.

Proof of Lemma 4.4a: Use $T_j = T_j(u)$ from Claim 4.13. Fix some ϵ-long edge uv and let $d = d_{uv}$. Let $t_j = \max(T_j(u), T_j(v))$. Then since by the 1-Lipschitz property $f'_{ij}(uv) \leq d$ for all ij; moreover, for any ij such that $i \geq t_j$ both $f_{ij}(u)$ and $f_{ij}(v)$ are at most $d/2^{i-t_j}$. Then $f'_{ij}(uv)$ is at most twice that much (since $f'_{ij} \leq f_{ij}$), so taking the sum of the geometric series we see that

$$\sum_{ij} f'_{ij}(uv) \leq \textstyle\sum_j \left(dt_j + \sum_{i \geq t_j} d/2^{i-t_j}\right) \leq \sum_j O(dt_j) = O\left(kd \log \frac{1}{\epsilon}\right),$$

where the last inequality follows by Claim 4.13. □

To prove part (b) Lemma 4.4, let us recall the definition of a u-broad interval: for a node u, an interval $[a, b]$ is *u-broad* if a or b is equal to d_{uv} for some v, $a \leq b/4$ and $|\mathbf{B}_u(a)| \leq \frac{1}{32}|\mathbf{B}_u(b)|$.

Proof of Lemma 4.4b: It suffices to consider the u-broad intervals $[a, b]$ such that one of the endpoints is equal to d_{uv} for some v, and the other is the largest b or the smallest a, respectively, such that the interval is u-broad. Call these intervals *u-interesting*; note that there are at most $2n$ such intervals for each u.

Fix node u and a u-broad interval $I = [a, b]$, fix j and let $r_i = f_{ij}(u)$. It suffices to show that with constant probability some r_i lands in I. Indeed, then we can use Chernoff bounds (Lemma 2.1a), and then we can take the Union Bound over all nodes u and all u-interesting intervals.

Denote by $\mathcal{E}_i$ the event that $r_i > b$ and $r_{i+1} < a$; note that these events are disjoint. Since some r_i lands in I if and only if none of the $\mathcal{E}_i$'s happen, we need to bound the probability of $\cup\mathcal{E}_i$ away from 1.

For each integer $l \geq 0$ define the interval

$$I_l = \left[\rho_u\left(\epsilon\, 2^l\right),\, \rho_u\left(\epsilon\, 2^{l+1}\right)\right),\ \text{where } \epsilon n = |\mathbf{B}_u(b)|.$$

For each $\alpha \in \{0, 1, 2, 3\}$ let $N_{(l,\alpha)}$ be the number of i's such that $r_{4i+\alpha} \in I_l$. We claim that $E[N_{(l,\alpha)}] \leq 1/q$.

Consider the case $\alpha = 0$; other cases are similar. Let $N_l = N_{(l,\alpha)}$ and suppose $N_l \geq 1$. Let i_0 be the smallest i such that $r_{4i} \leq I_l$. Then $N_l \geq t$ implies $X_{ij} = 0$ for each $i \in [i_0;\, i_0 + t - 2]$. Recall that the construction of the maps f_{ij} starts with $f_{(0,j)}$. Given the specific map $f = f_{(i_0,j)}$, the construction of the maps f_{ij}, $i > i_0$ is equivalent to a similarly defined construction that starts with $f_{(i_0,j)} = f$. Therefore, by (4.5) (applied to this modified construction) we have

$$\begin{aligned}
\Pr[N_l \geq t] &\leq \Pr\left[\textstyle\sum_{\beta=0}^{t-2} X_{(i_0+\beta,\, j)} = 0\right] \leq \Pr\left[\textstyle\sum_{\beta=0}^{t-2} Y_\beta = 0\right] = (1-q)^{t-1};\\
E[N_l] &= \textstyle\sum_{t=1}^{\infty} \Pr\left[N_l \geq t\right] \leq \sum_{t=1}^{\infty}(1-q)^{t-1} = \frac{1}{q},
\end{aligned}$$

claim proved. For simplicity assume $k = 4k_0 + 1$; it follows that

$$\textstyle\sum_{i=0}^{k-1} \Pr[r_i \in I_l] = \sum_{\alpha=0}^{3} \sum_{i=0}^{k_0-1} \Pr[r_{4i+\alpha} \in I_l] = \sum_{\alpha=0}^{3} E\left[N_{(l,\alpha)}\right] \leq 4/q. \tag{4.7}$$

By Claim 4.11 if $r_i \in I_l$ then $r_{i+1} \leq a$ with conditional probability at most $|\mathbf{B}_u(a)|/|\mathbf{B}_u(r_u)| \leq 2^{-l}/32$. Therefore, $\Pr[\mathcal{E}_i \mid r_i \in I_l] \leq 2^{-l}/32$. By (4.7) it follows that

$$\begin{aligned}
\Pr[\cup\mathcal{E}_i] &= \sum_{i=0}^{k-1} \Pr[\mathcal{E}_i] = \sum_{i=0}^{k-1}\sum_{l=0}^{\infty} \Pr\left[r_i \in I_l \text{ and } \mathcal{E}_i\right] \leq \sum_{i=0}^{k-1}\sum_{l=0}^{\infty} \Pr[r_i \in I_l] \times 2^{-l}/32\\
&= \frac{1}{32}\sum_{l=0}^{\infty} 2^{-l} \sum_{i=0}^{k-1} \Pr[r_i \in I_l] \leq \frac{1}{8q}\sum_{l=0}^{\infty} 2^{-l} = \frac{1}{4q} < 1,
\end{aligned}$$

so some r_i lands in I with at least a constant probability. □

It remains to prove Lemma 4.5 about the maps $g_{(i,\,j,\,0)}$.

Proof of Lemma 4.5: Let's pause our embedding algorithm right after the map f_{ij} is chosen, and consider the probability space induced by the forthcoming random choices. Let $X_w = f_{ij}(w)$. First we claim that

$$\Pr\left[g_{(i,\,j,\,0)}(u) \leq r \mid r \leq X/8\right] \geq \Omega(\beta_r), \tag{4.8}$$

where $\beta_r = |\mathbf{B}_u(r)|/|\mathbf{B}_u(X)|$. Indeed, suppose $r \leq X/8$, let $B = \mathbf{B}_u(r)$ and consider any $w \in B$. Then by (4.11):

$$\begin{aligned}
\Pr[w \in W_{ij}] &= 1/|\mathbf{B}_w(X_w/2)| \geq 1/|\mathbf{B}_u(X)| \geq \beta_r |B|\\
\Pr\left[\, g_{(i,\,j,\,0)}(u) \leq r\,\right] &= \Pr[W_{ij} \text{ hits } B] \geq 1 - (1 - \beta_r|B|)^{|B|} \geq 1 - e^{-\beta_r} \geq \Omega(\beta_r),
\end{aligned}$$

proving (4.8). Now let $B = \mathbf{B}_v(X_v/8)$; then by (4.11) any $w \in B$ is included into the set W_{ij} with probability at most $1/B$, so

$$\Pr\left[\, g_{(i,\,j,\,0)}(v) \geq X_v/8\right] = \Pr[W_{ij} \text{ misses } B] \geq (1 - 1/|B|)^{|B|} \geq 1/4. \tag{4.9}$$

Finally, let's combine (4.8) and (4.9) to prove the claim. Let $r = d/4$ and suppose $X \geq 4d$. Since $X_v \geq X - d_{uv} \geq 3d$, by (4.9) event $g_{(i,\,j,\,0)}(v) \geq 3d/8$ happens with probability at least $1/4$. This event and the one in (4.8) are independent since they depend only on what happens in the balls $\mathbf{B}_u(d/4)$ and $\mathbf{B}_v(3d/8)$, respectively, which are disjoint. Therefore with probability at least $\Omega(\beta_r)$ both events happen, in which case $g_{(i,j,0)}(uv) \geq d/8$. □

4.7 A Bourgain-style proof of Lemma 4.2 for doubling metrics.

In this section we use the ideas of [Bou85, LLR95] to derive an alternative proof of Lemma 4.2 for the important special case when β is the doubling dimension. In this proof the target dimension becomes $t = O(\beta \log \beta)$, which results in target dimension $O(\log^2 n)(\beta \log \beta)$ in Theorem 4.1.

Let us note that in the well-known embedding algorithms of Bourgain [Bou85] and Linial et al. [LLR95] any two nodes are sampled with the same probability, i.e. with respect to the counting measure. Here use a non-trivial extension of the Bourgain's technique where we sample with respect to a doubling measure transformed with respect to a given 1-Lipschitz map.

We state our result as follows:

Lemma 4.14. *Consider a finite metric (V, d) equipped with a non-degenerate measure μ and a 1-Lipschitz coordinate map f; write $f_u = f(u)$. For every node u let*

$$\beta_\mu(u) = 2\mu[\,\mathbf{B}_u(f_u)\,] \;/\; \mu[\,\mathbf{B}_u(f_u/16)\,].$$

Then for any $k, t \in \mathbb{N}$ there is a randomized embedding g into ℓ_p, $p \geq 1$ with dimension kt so that:

(a) each coordinate map of g is 1-Lipschitz and upper-bounded by f; and

(b) $\|g(u) - g(v)\|_p \geq \Omega(d_{uv}/t)(kt)^{1/p}$ with failure probability at most $< t/2^{\Omega(k)}$ for any edge uv such that

$$f(u)/d_{uv} \in [1/4;\, 4] \text{ and } \max_{w \in \{u,v\}} \beta_\mu(w) \leq 2^t. \tag{4.10}$$

To prove Lemma 4.2 for a metric of doubling dimension β, recall that for any such metric there exists a 2^β-doubling measure μ. Plug this measure in Lemma 4.14, with $t = 4\beta + 1$ and $k = O(\log \beta)$; note that $\beta_\mu(u) \leq 2^t$ for every node u. We get the embedding in ℓ_p with $O(\beta \log \beta)$ dimensions that satisfies the conditions in Lemma 4.2.

We will need the following simple fact:

$$\text{If } d_{uv} \leq f(u)/8 \text{ for some edge } uv\text{, then } \mathbf{B}_u(f(u)/8) \subset \mathbf{B}_v(f(v)/2) \subset \mathbf{B}_u(f(u)) \tag{4.11}$$

Indeed, letting $f_u = f(u)$ the first inclusion follows since $f_v/2 \geq (f_u - d_{uv})/2 \geq f_u/8 + d_{uv}$, and the second one holds since $d_{uv} + f_v/2 \leq d_{uv} + (f_u + d_{uv})/2 < f_u$.

Proof of Lemma 4.14: Define the transformation of μ with respect to f as $\mu_f(u) = \mu(u)/2\mu(B)$, where $B = \mathbf{B}_u(f_u/2)$. Fix $k = c \log n$ where c is an absolute constants to be specified later. The coordinates are indexed by ij, where $i \in [t]$ and $j \in [k]$. For each (i,j)-pair construct a random set U_{ij} by selecting $\lceil 2^i \mu_f(V) \rceil$ nodes independently according to the probability distribution $\mu_f(\cdot)/\mu_f(V)$. Let us define the ij-th coordinate of u as $g_{ij}(u) = \min(f_u, d(u, U_{ij}))$.

Note that each map g_{ij} is 1-Lipschitz as the minimum of two 1-Lipschitz maps. Therefore part (a) holds trivially. The hard part is part (b). Fix an edge uv; let $d = d_{uv}$. For any node w let $\alpha_w(\epsilon)$ be the smallest radius r such that $\mu_f[\mathbf{B}_w(r)] \geq \epsilon$, and let

$$\rho_i = \max[\psi_u(2^{-i}), \psi_v(2^{-i})], \text{ where } \psi_w(\epsilon) = \min[\alpha_w(\epsilon), d/2, f_w].$$

Claim 4.15. *For each $i \geq 1$ and each $j \in [k]$ with probability $\Omega(1)$ we have*

$$g_{ij}(uv) := |g_{ij}(u) - g_{ij}(v)| \geq \rho_i - \rho_{i+1}.$$

Then by Chernoff bounds (Lemma 2.1(a)) w.h.p. we have

$$\sum_{ij} g_{ij}(uv) \geq \sum_{i=1}^{t} \Omega(k)(\rho_i - \rho_{i+1}) = \Omega(k)(\rho_1 - \rho_t). \tag{4.12}$$

Proof of Claim 4.15: Fix $i \geq 1$ and j, and note that if $\rho_{i+1} = d/2$ then $\rho_i = d/2$, in which case the claim is trivial. So let's assume $\rho_{i+1} < d/2$ and without loss of generality suppose $\psi_u(2^{-i}) \geq \psi_v(2^{-i})$. Consider the open ball B of radius ρ_i around u. Since $\rho_i = \psi_u(2^{-i}) \leq \alpha_u(2^{-i})$, it follows that $\mu_f(B) \leq 2^{-i}$. Now there are two cases:

- If $\rho_{i+1} = f_v$ then the desired event $g_{ij}(uv) \geq \rho_i - \rho_{i+1}$ happens whenever U_{ij} misses B, which happens with at least a constant probability since $\mu_f(B) \leq 2^{-i}$.
- If $\rho_{i+1} < f_v$ then the desired event happens whenever U_{ij} misses B and hits $B' = \mathbf{B}_v(\rho_{i+1})$. This happens with at least a constant probability by Claim 4.17 since $\rho_{i+1} \geq \psi_v(1/2^{i+1}) \geq \alpha_v(1/2^{i+1})$ and therefore $\mu_f(B') \geq 1/2^{i+1}$, and the two balls B and B' are disjoint. $\square$

Claim 4.16. *For any node w we have $\alpha_w(\frac{1}{2}) \geq f_w/8$ and $\alpha_w(1/\beta_\mu(w)) \leq f_w/16$.*

Proof. Let $B = \mathbf{B}_w(f_w/8)$. By (4.11) for any $w' \in B$

$$\mu(w)\,/\,2\mu[\,\mathbf{B}_w(f_w)\,] \;\leq\; \mu_f(w') \;\leq\; \mu(w)/2\mu(B),$$

so $\mu_f(B) \leq \frac{1}{2}$ and $\mu_f[\,\mathbf{B}_w(f_w/16)\,] \geq 1/\beta_\mu(w)$. $\square$

Suppose (4.10) holds; let $x = \max(f_u, f_v)$. Then by Claim 4.16 and the definitions of ρ_i and ψ_w we have:

$$\begin{aligned} \rho_1 &\geq \max_{w\in\{u,v\}} \min(f_w/8, d/2) \geq \min(x/8, d/2), \\ \rho_t &\leq \max_{w\in\{u,v\}} \alpha_w(2^{-t}) \leq \max_{w\in\{u,v\}} \alpha_w\left(1/\beta_\mu(w)\right) \leq \max_{w\in\{u,v\}} f_w/16 \leq x/16. \end{aligned}$$

By (4.12) for $p = 1$ it remains to show that $\rho_1 - \rho_t \geq \Omega(d)$. There are two cases:

- if $f_v \leq 4d$ then $\rho_1 \geq x/8$, so $\rho_1 - \rho_t \geq x/16 \geq \Omega(d)$.
- if $f_v > 4d$ then $\rho_1 \geq d/2$ and (since f is 1-Lipschitz)

  $$\rho_t \;\leq\; f_v/16 \;\leq\; (f_u + d)/16 \;\leq\; 5d/16,$$

  so $\rho_1 - \rho_t \geq 3d/16$.

This completes the proof for the case $p = 1$. To extend it to $p > 1$, note that the embedded uv-distance is

$$\left(\textstyle\sum_{ij} g_{ij}(uv)^p\right)^{1/p} = (kt)^{1/p}\left(\tfrac{1}{kt}\textstyle\sum_{ij} g_{ij}(uv)^p\right)^{1/p} \geq (kt)^{1/p}\left(\tfrac{1}{kt}\textstyle\sum_{ij} g_{ij}(uv)\right) \geq \Omega(d/t)\,(kt)^{1/p}. \quad \square$$

In the above proof we used the following claim which is implicit in [LLR95] and also stated in [KSW04]; we prove it here for the sake of completeness.

Claim 4.17. *Let μ be a probability measure on a finite set V. Consider disjoint events $E, E' \subset V$ such that $\mu(E) \geq q$ and $\mu(E') \leq 2q < 1/2$ for some number $q > 0$. Let S be a set of $\lceil 1/q \rceil$ points sampled independently from V according to μ. Then S hits E and misses E' with at least a constant probability.*

Proof. Obviously, the probability that S hits E and misses E' can only increase if we set $\Pr[E] = q$ and $\Pr[E'] = 2q$. Treat sampling a given point as two independent random events: first it misses E' with probability $1 - 2q$, and then (if it indeed misses) it hits E with probability $q' = \frac{q}{1-2q} \leq 2q$. Without loss of generality rearrange the order of events: first we choose whether all points miss E', and then upon success choose whether at least one point hits E. These two events happen independently with probabilities, respectively, $(1-2q)^{1/q} \geq 2^{-1/2}$ and

$$1 - (1-q')^{1/q} \geq 1 - (1-2q)^{1/q} \geq 1 - e^{-2}.$$

So the total success probability is at least $c = (1 - e^{-2})/\sqrt{2}$, which is an absolute constant as required. $\square$

4.8 An extension to arbitrary metrics

In this section we consider probabilistic embeddings of arbitrary metrics into trees, which extend to (high-dimensional) embeddings into ℓ_1. We exploit the technique of [FRT04] to obtain embeddings with slack. First we show that it gives a probabilistic embedding of arbitrary metrics into tree metrics with *expected* gracefully degrading distortion $D(\epsilon) = O(\log 1/\epsilon)$. For technical convenience, we will treat n-point metrics as functions from $[n] \times [n]$ to reals. Note that all metrics d_T generated by the algorithm in [FRT04] are *dominating*, i.e. for any edge uv we have $d(u, v) \leq d_T(u, v)$.

Theorem 4.18. *For any input metric (V, d), let d_T be the dominating HST metric on V constructed by the randomized algorithm in Fakcharoenphol et al. [FRT04]. Then the embedding from (V, d) to (V, d_T) has* expected *gracefully degrading distortion $D(\epsilon) = O(\log 1/\epsilon)$. Specifically, for any parameter $\epsilon > 0$ and any ϵ-long edge uv we have*

$$d_{uv} \leq E_\varphi[d_T(u, v)] \leq O(\log 1/\epsilon)\, d_{uv}. \tag{4.13}$$

Since tree metrics are isometrically embeddable into L_1, it follows that we can embed any metric into L_1 with gracefully degrading distortion $D(\epsilon) = O(\log \frac{1}{\epsilon})$.

Proof. For simplicity let us assume that all distances in (V, d) are distinct; otherwise we can perturb them a little bit and make them distinct, without violating the triangle inequality; see the full version of this paper for details. In what follows we will assume a working knowledge of the decomposition scheme in [FRT04].

Let us fix the parameter $\epsilon > 0$ and an ϵ-long edge uv, and let $d = d(u, v)$. Let us assume without loss of generality that $\rho_u(\epsilon) \leq \rho_v(\epsilon)$. Then $\rho_u(\epsilon) \leq d$, so $|\mathbf{B}_u(d)| \leq \epsilon n$.

Run the randomized algorithm of [FRT04] to build a tree T and the associated tree metric d_T. The decomposition scheme will separate u and v at some distance scale $2^i \geq d/2$. Let Δ be the maximum distance in the input metric. Under the distribution over tree metrics d_T that is induced by the algorithm, the expected distance $E[d_T(u, v)]$ between u and v in tree T is equal to the sum

$$\textstyle\sum_{i \geq \log d - 1}^{\log \Delta} 4 \cdot 2^i \times Pr[(u, v) \text{ first separated at level } 2^i].$$

Look at the sum for i such that $d/2 \leq 2^i < 4d$: this is at most $48d$. By the analysis of [FRT04], the rest of the sum, i.e. the sum for $i \geq \log 4d$, is at most

$$\textstyle\sum_{i \geq \log 4d}^{\log \Delta} 4 \cdot 2^i \times \frac{2d}{2^i} \log \frac{|\mathbf{B}_u, 2^i)|}{|\mathbf{B}_u, 2^{i-2})|}$$

Since the above sum telescopes, it is at most

$$8d \cdot 2 \log\left(n / |\mathbf{B}_u(d)|\right) \leq O(d \log 1/\epsilon),$$

which proves the second inequality in (4.13). The first inequality in (4.13) holds trivially because all metrics d_T generated by the algorithm in [FRT04] are dominating. □

The above embedding into ℓ_1 can be made algorithmic by sampling from the distribution and embedding each sampled tree into ℓ_1 using a fresh set of coordinates; however, the number of trees now needed to give a small distortion may be as large as $\Omega(n \log n)$. We also note in passing that a slightly modified analysis yields an embedding into a *single tree*; we omit the details.

Theorem 4.19. *For any source metric (V, d) and any parameter $\epsilon > 0$ there exists an embedding into a dominating HST metric with ϵ-uniform slack and distortion $O(\frac{1}{\epsilon} \log \frac{1}{\epsilon})$.*

Chapter 5

Network Triangulation via Rings of Neighbors

In this chapter we consider growth-constrained metrics and construct triangulation with guarantees for all node pairs. This is our main result:

Theorem 5.1. *Consider a metric with grid dimension α and polynomially bounded aspect ratio. Suppose each node has links to 3 nodes sampled independently at random in the network. Then for any $\delta > 0$ there exists a fully distributed algorithm that computes a $(0, \delta)$-triangulation of degree $(O(1/\delta))^{\alpha}(\log^2 n)$. The running time and per-node load is $(O(1/\delta))^{\alpha}(\log^7 n)$.*

A crucial element of our construction is *rings of neighbors*, a sparse distributed data structure which captures the distance information in the network. This is essentially the data structure that underlies *Meridian*, a network positioning system described in Chapter 6.

The results in this chapter represent a very recent work [Sli06] that has not yet been published.

5.1 Framework and results

Let us properly formulate our results. Since our construction is a fairly involved distributed algorithm, we need to put forward a precise model of distributed computation. Then we give describe and define rings of neighbors, and then we state our results.

Model of distributed computation. Nodes do not share data and communicate via messages. Specifically, each node follows the following cycle: receive a message, do local computation, (possibly) send messages to other nodes, wait for the next message received. We assume that some nodes initiate the algorithm by sending messages without receiving someone else's message first. These nodes are called a *starting* nodes; we assume that they start sending all at the same time.

Each node has an *address* that other nodes need to know in order to contact this node. These addresses cannot be guessed; they can either be given in advance or passed from one node to another. In particular, initially every node is given a (possibly empty) list of addresses. Later in the algorithm, a node may delete some addresses from its list, or may add some new addresses received from other nodes. The nodes whose addresses are currently in the list of node u are called the *neighbors* of u. For simplicity we assume that each address takes $O(1)$ space.

Let V be the set of all nodes; let $S \subset V$ and let $G = (V, E)$ be a graph, directed or undirected, possibly with multiple edges and self-loops. Say an algorithm is (S, G)-*distributed* if it conforms to the above model so that S is the set of starting nodes, and initially every node u is given the addresses of all its G-out-neighbors (if G is directed) or of all its G-neighbors (if G is undirected), together with the corresponding multiplicities. If $S = \{u\}$, say the algorithm is (u, G)-distributed. If $S = V$, say the algorithm is G-distributed.

The *load* on a given node includes computation, storage, and communication. For simplicity the load is defined as a sum $x + y + z$, where x is the number of CPU cycles, y is the number of bytes used for storage, and z is the number of bits sent and received; we will use the $O(\cdot)$ notation, so the exact units do not matter. The load of an algorithm is the maximal load on a node.

We also interested in the per-node space requirement. For clarity, we do not count the size of the message queue towards this requirement (but note that the number of messages sent and received does count towards the load). We assume that each node processes messages sequentially, and that the message is in the network until the node is ready to process it.

If the algorithm starts at time 0, and terminates at time τ_u on each node u, then the *total running time* is defined as $\max \tau_u$. Note that it can be very different from the load, since the latter in general does not include the idle time.

Node selection in a metric space. Say a set of nodes *supports* a metric d_{M} if any two nodes u, v in this set can measure $d_{\mathrm{M}}(u, v)$ at a unit cost once they communicate. Intuitively, any such d_{M} is a notion of distance between the nodes.

In what follows, let us fix a metric d_{M} which is supported on V. Let $B_u(r)$ be the closed ball of radius r around node u, i.e. $B_u(r) = \{v \in V : d_{\mathrm{M}}(u, v) \leq r\}$. Without loss of generality assume that the minimal distance is 1; let Δ be the diameter of the metric. Denote $B_{ui} = B_u(\Delta/2^i)$.

We seek to construct a distributed data structure that we call *rings of neighbors*. In this data structure, for each $i \in [\log \Delta]$ each node u stores addresses of k other nodes in B_{ui}. We denote these k nodes as $X_u^{(i)} = \left\{ X_{uj}^{(i)} : j \in [k] \right\}$ and call $X_u^{(i)}$ the *i-th ring of neighbors* of node u. Here k is a small number, e.g. $k = \Theta(\log n)$, which we call *ring cardinality*.

Suppose we have a randomized algorithm which constructs the rings of neighbors and, consequently, induces a joint probability distribution on random variables $\left\{ X_{uj}^{(i)} \right\}$. Intuitively, we would like these random variables to be independent and uniformly distributed on the respective balls B_{ui}. We will construct a slightly weaker version.

Let n be the cardinality of V, and let σ_{unif} be the uniform distribution on V. Say a distribution τ on V is *near-uniform* if $\|\sigma_{\text{unif}} - \tau\|_\infty \leq \frac{1}{2n}$. We can define near-uniform distributions on any given subset of nodes in a similar fashion. Let $\mathcal{F}_i$, $i \in [\log \Delta]$ be the collection of random variables from the i-th rings of neighbors of all nodes:

$$\mathcal{F}_i = \left\{ X_{uj}^{(i)} : u \in V, j \in [k] \right\}. \tag{5.1}$$

For notational convenience, define $\mathcal{F}_{-1}$ to be empty. We will construct rings of neighbors such that given $\cup_{l<i} \mathcal{F}_l$ all random variables in $\mathcal{F}_i$ are conditionally independent and near-uniformly distributed. More formally:

Definition 5.2. Consider a metric space (V, d_{M}) on n nodes, with aspect ratio Δ. *Randomized Rings of Neighbors* (RRN) on this metric space is a joint distribution on V-valued random variables

$$\left\{ X_{uj}^{(i)} : u \in V, i \in [\log \Delta], j \in [k] \right\}$$

such that with high probability the following two properties hold for each $i < \lceil \log \Delta \rceil$:

(P1) given $\cup_{l<i} \mathcal{F}_l$, random variables in $\mathcal{F}_i$ are conditionally independent.

(P2) given $\cup_{l<i} \mathcal{F}_l$, each random variable $X_{uj}^{(i)} \in \mathcal{F}_i$ has a near-uniform distribution on B_{ui}.

Here k is a fixed parameter called *ring cardinality*, and $\mathcal{F}_i$ is defined by (5.1).

Our results We provide a distributed and load-balanced construction for Randomized Rings of Neighbors on a growth-constrained metric. We need the initial communication graph to have low degree-expansion ratio.

Theorem 5.3. *Let $G = (V, E)$ be an undirected graph on n nodes. Suppose that G is an $O(\log n)$-degree expander and that V supports a metric d_M of polynomially-bounded aspect ratio and grid dimension α. Assume that each node knows (the same) constant-factor approximate upper bounds on the following:* $\log n$, *degree and expansion of G, aspect ratio and grid dimension of d_M.*

Then for any given constant $c \geq 1$ there exists a randomized G-distributed algorithm with running time and load $O(c^4)\, 2^{O(\alpha)}(\log^7 n)$ which with high probability constructs Randomized Rings of Neighbors on (V, d_M) with ring cardinality $\Omega(c\, 4^{\alpha}\, \log n)$.

The quantitative assumption that G is a $O(\log n)$-degree expander is there for numerical convenience only. It can be replaced by a low degree-expansion ratio, using the following relatively simple pre-processing step:

Lemma 5.4. *Let $G = (V, E)$ be a connected undirected graph on n nodes, of expansion γ. Suppose the numbers $d \geq \deg(G)$ and $t \geq (d/\gamma)^2\,(\log n)$ are known to all nodes.*

Then there exists a randomized G-distributed algorithm whereby every node acquires the addresses of three nodes in V so that with high probability the induced undirected graph on V is an $O(\log n)$-degree expander. The running time and (with high probability) the per-node load are $O(t \log n)$; the per-node storage is $O(d)$.

We use the construction in Theorem 5.3 to construct the desired triangulation for Theorem 5.1. Let us restate the latter theorem in a slightly more general form.

Theorem 5.5. *In the setting of Theorem 5.3, for any given $\delta \in (0, 1)$ there exists a randomized G-distributed algorithm with running time and load $O(1/\delta)^{\alpha}\; O(\log^7 n)$ which with high probability constructs a $(1+\delta)$-approximate network triangulation of degree $O(1/\delta)^{\alpha}\; O(\log^2 n)$.*

Off-line network measurements. The above results are also meaningful in a setting where the network measurements are reported to and processed in a central location. In this setting treat the network as an oracle which for any given node pair uv returns $d_{\mathrm{M}}(u, v)$ at unit cost to both u and v. We (essentially) do not need to worry about communication and processing, yet it is still desirable, and quite non-trivial, to reduce the measurement load on nodes.

Notation. Say random variables $X_1 \ldots X_k$ are *Q-nice* if their joint distribution is that of k independent random variables with a near-uniform distribution on Q.

5.2 Tools: distributed random walks

In this section we discuss load-balanced random node selection via distributed random walks. Note that we do not introduce the metric space until the next section.

Consider an undirected graph $G = (V, E)$. Let λ_{uv} be the multiplicity of edge uv, and let $d_u = \sum_v \lambda_{uv}$ be the degree of node u. For any $d \geq \deg(G)$ let us define the Markov chain $M_{(G,d)}$ as follows:

$$M_{(G,d)}(u,v) = \begin{cases} \lambda_{uv}/2d & \text{if } u \neq v \\ 1 - (d_u - \lambda_{uv})/2d & \text{otherwise} \end{cases} \tag{5.2}$$

It is easy to see that this Markov chain has a uniform stationary distribution. Moreover, by Theorem 2.5 for graphs of low expansion-degree ratio it has a rapid mixing property:

Lemma 5.6. *Let G be a connected undirected graph (possibly with loops and parallel edges) with expansion γ. Then for any $d \geq \deg(G)$ and $k \geq O(d/\gamma)^2(\log n)$ the k-step distribution of $M_{(G,d)}$ is near-uniform for any initial distribution.*

Proof. Let $M = M_{(G,d)}$. Note that M is irreducible since G is connected, and M is aperiodic since every node has a positive stalling probability. Therefore M is ergodic. M is time-reversible since $M(u,v) = M(v,u)$ holds for all node pairs. Since $M(u,v) \geq \lambda_{uv}/2d$ for all node pairs, the expansion of M (as an edge-weighted graph) is γ/d. Now the Lemma follows from Theorem 2.5. □

Using Theorem 2.3, Lemma 5.4 is a simple corollary of a more general fact:

Lemma 5.7. *In the setting of Lemma 5.4, for any $k \in \mathbb{N}$ there exists a randomized G-distributed algorithm whereby every node u acquires k addresses X_{uj}, $j \in [k]$, such that $\{X_{uj} : u \in V, j \in [k]\}$ are V-nice random variables in the probability space induced by the algorithm. The running time and (with high probability) the load are $O(t \times \max(k, \log n))$; the per-node storage is $O(k+d)$.*

Proof. By abuse of notation, let us fix some enumeration f of V and treat each node u as a unit vector in the $f(u)$-th dimension. Let σ_{unif} be the uniform distribution on V.

For a node v, let $\mathcal{A}_v = \mathcal{A}_v(G,d)$ be a v-distributed algorithm that starts at v and simulates the Markov chain $M_{(G,d)}$ for t steps. Specifically, at every step i the Markov chain visits some node X_i, which means the following: node X_i selects one of its G-neighbors (or itself) according to the distribution (5.2) and forwards the Markov chain to this node. The process starts at $X_0 = v$, and terminates at step t by returning X_t to node v.

Note that by Lemma 5.6 X_t is a random variable with a near-uniform distribution. For simplicity let us assume that at each step i nodes v and X_i experience a unit load each. It follows that for a given node w the expected load induced by algorithm $\mathcal{A}_v$, $v \neq w$ is equal to

$$\Pr[X_i = w] = \left(M^{(i)}_{(G,d)} v\right) \cdot w. \tag{5.3}$$

The overall algorithm is simple: every node u initiates k independent copies of algorithm $\mathcal{A}_u$. In the course of this algorithm, each message processed by a given node u is related to a certain step of some $\mathcal{A}_v$. To simplify the analysis of the total running time, let us assume that whenever there is contention, messages from earlier steps are given higher priority.

First, note that the per-node storage requirement is $O(k+d)$, since at any point in time a given node u needs to store only the addresses of all his G-neighbors, the current step for each of the k copies of algorithm $\mathcal{A}_u$.

Let us fix a node w and a step $i \in [t]$. Let Y_{vj} be the load induced on w by the j-th copy of algorithm $\mathcal{A}_v$. Then by (5.3) we have

$$\sum_{v \neq w} E\left(Y_{vj}\right) = \sum_{v \neq w} \left(M^{(i)}_{(G,d)}\, v\right) \cdot w \leq n \left(M^{(i)}_{(G,d)}\, \sigma_{\text{unif}}\right) \cdot w = O(n\, \sigma_{\text{unif}} \cdot w) = O(1).$$

Since $\{Y_{vj}\}$ is a family of bounded independent random variables, by Chernoff Bounds (Theorem 2.1b) with $\mu = \Theta(\max(k, \log n))$ it follows that $\sum_{\text{all }(v,j)} Y_{vj} \leq 2\mu$ with high probability. In particular, the total load on any given node (over all steps) is $O(t\mu)$ with high probability.

To bound the total running time, we claim that the processing of each step i completes, for all nodes, by time $O(i\mu)$ with high probability. Indeed, suppose a given step i is complete by time $O(i\mu)$. Since with high probability every given node u needs to process at most $O(\mu)$ messages for step $i+1$, and these messages have priority over those from later steps, processing them will take at most $O(\mu)$ time. Claim proved. □

The above construction is also used as the first step for the algorithm in Theorem 5.3, to construct the out-most rings of neighbors. To construct all other rings, we need a more general version of Lemma 5.7, where each node u wants to gather several independent near-uniform samples of some subset Q_u. The crucial difference is that in this case the subsets Q_u are different, e.g. $Q_u = B_{ui}$ for some fixed i. We state this result for a single node, call it u. To make this result useful, we need to bound the expected load on all nodes in $Q_u \setminus \{u\}$ by a small multiple of $1/|Q_u|$.

Lemma 5.8. *Let $G = (V, E)$ be an undirected graph on n nodes. Fix node u and consider a subset $Q \subset V$ such that the graph $G|Q$ has expansion γ. Suppose that:*

- *after pinging any node $v \in V$, node u can, at unit cost, determine whether $v \in Q$.*
- *node u knows numbers $d \geq \deg(G)$, $d_Q \geq \deg(G|Q)$ and $t \geq (d_Q/\gamma)^2 (\log n)$,*
- *node u is given a* random seed*: an address of some node.*

Then for any $k \in \mathbb{N}$ there exists a randomized (u, G)-distributed algorithm such that:

(a) node u acquires addresses of k nodes $X_i \in Q$, where the X_i's are Q-nice random variables. The running time and the load on node u are $O(kdt)$.

(b) The load on every other node w is at most $O\left(\sum_{wv \in G} Z_v\right)$, where Z_v is the number of times node v is "visited" by the algorithm,[1] which is at most kt for all $v \in Q$, and 0 otherwise. If the random seed was selected independently from a near-uniform distribution τ on Q, then in the probability space induced by the algorithm and τ, $E(Z_v) = O(kt/|Q|)$ for each $v \in Q$.

Proof. We use algorithm $\mathcal{A}_v(G|Q,\, d)$ defined in the proof of Lemma 5.7, in a slightly modified form. Specifically, at each step i of this algorithm node u communicates with some node $X_i \in Q$, asks this node for a list of its G- neighbors, determines which of these neighbors lie in Q, and chooses the next node X_{i+1} among those according to the distribution $M_{(G|Q,\, d)}$, see (5.2). The process starts at $X_0 = v$, and terminates at step t by returning X_t to node u. During each step node u incurs load $O(d)$, and node X_i incurs load $O(1)$.[2]

The overall algorithm is simple: node u initiates k independent copies of algorithm $\mathcal{A}_w(G|Q,\, d)$, where w is the given random seed.

Parts (a) is trivial. For part (b), we define Z_v to be the number of times some copy of algorithm $\mathcal{A}_v(G|Q,\, d)$ selects node v as the next step. Let us fix some node $v \in Q \setminus \{u\}$, and let Y_{ij} be the number of times node v is visited by the i-th step of the j-th copy of the random walk. Let σ_{unif} be the uniform distribution on Q, and let $M = M^{(i)}_{(G|Q,\, d)}$ be the i-th power of the corresponding transition matrix. Note that $M\sigma_{\text{unif}} = \sigma_{\text{unif}}$, so rows of M have unit sums, so $\|M\tau\|_\infty \leq \|\tau\|_\infty = O(1/|Q|)$. Consider the probability space induced by the algorithm and τ. Then

$$E[Y_{ij}] = O\left(\Pr_{\tau}[X_i = v]\right) = O\left((M\,\tau) \cdot v\right) = O(1/|Q|).$$

[1] For each node v, the algorithm either does not touch the list of G-neighbors of v, or reads the entire list at once. In the latter case we say that the algorithm *visits* node v.

[2] Node X_i sends a list of d addresses. However, in practice this list should fit in a very small number of packets.

We get $E(Z_v) = O(kt/|Q|)$ by summing over all $i \in [t]$ and $j \in [k]$. □

Lemma 5.7 and Lemma 5.8 are of independent interest in the context where for each node it is desirable to have a small random sample of the entire network or an ”interesting” subset thereof.

5.3 Randomized Rings of Neighbors: Proof of Theorem 5.3

In this section we bring in the metric space d_{M} and use Lemma 5.7 and Lemma 5.8 in a black-box way in order to construct the Randomized Rings of Neighbors.

Overview of the algorithm. Our construction proceeds in $\lceil \log \Delta \rceil$ stages. A given stage $i \in [\log \Delta]$ handles distances on the scale of $\Delta/2^i$: this is when all $(i+1)$-th ring neighbors are constructed. Specifically, at the beginning of stage i, the i-th ring neighbors of all nodes have already been constructed. For each node u, they induce a low-degree expander Q on the ball $B_{(u,i+1)}$; this is essentially because they are conditionally independent and distributed near-uniformly on the corresponding balls of radius $\Delta/2^i$. Node u selects its $(i+1)$-th ring neighbors by executing independent random walks on Q; by the expansion property of Q in order to guarantee near-uniformity it suffices to run these random walks for poly-log many steps. In fact, we use Lemma 5.8 for the random walks. Note that we need to be careful to separate the random seeds used in this lemma from the graph on which we do the random walk. For every node a large portion of the load comes from helping other nodes choose their neighbors; one needs to be very careful to guarantee that no node is overloaded by helping others.

Recall that V supports a metric d_{M} of grid dimension α. For simplicity, let us assume that each node knows exact values (not just constant-factor approximate upper bounds) for the following quantities: degree of G, expansion of G, aspect ratio of d_{M}, grid dimension of d_{M}, and $\log n$.

Let us set $k = c\, 4^{\alpha} \lceil \log n \rceil$ to be the ring cardinality, where the constant c is chosen at least large enough so that the following property holds:

(P3) Consider any ball B_{ui} and any ball $B_{(v,\, i+1)} \subset B_{ui}$. Suppose k nodes are chosen independently from a near-uniform distribution on B_{ui}. Then at least $3 \log n$ of them land in $B_{(v,\, i+1)}$, with failure probability at most $1/n^4$.

Note that in (P3) $|B_{ui}| \leq 4^{\alpha} |B_{(v,\, i+1)}|$ by Lemma 2.14, so $c = O(1)$ does indeed suffice.

Recall that the rings of neighbors are described by random variables $X^{(i)}_{ui}$, which are grouped in collections $\mathcal{F}_i$, see (5.1). We start by constructing $\mathcal{F}_0$ using Lemma 5.7 applied to the original connectivity graph G. Such $\mathcal{F}_0$ clearly satisfies conditions (P1) and (P2). Since G is a $O(\log n)$-degree expander, in Lemma 5.7 we take $t = O(\log^3 n)$, so for stage 0 the running time and (with high probability) the load are $2^{O(\alpha)}(\log^4 n)$, and the storage requirement is $O(k)$.

The rest of the construction proceeds in stages, so that in stage $i \geq 0$ we construct $\mathcal{F}_{i+1}$ assuming that we have already constructed $\mathcal{F}_i$ that satisfies (P1) and (P2). Let us partition the family $\mathcal{F}_i$ of random variables into two subfamilies:

$$\mathcal{F}_i^{\text{walk}} = \left\{ X^{(i)}_{uj} \,:\, u \in V,\, j \in [k/2] \right\} \;\text{ and }\; \mathcal{F}_i^{\text{seed}} = \mathcal{F}_i \setminus \mathcal{F}_i^{\text{walk}}.$$

We will invoke Lemma 5.8, independently for every node u. The underlying graph for the random walks will come from $\mathcal{F}_i^{\text{walk}}$, and the random seeds will come from $\mathcal{F}_i^{\text{seed}}$. It is important that the random seed is independent of $\mathcal{F}_i^{\text{walk}}$ (conditionally, given $\mathcal{F}_{i-1}$).

Let us define G^*_i to be the directed graph induced by $\mathcal{F}_i^{\text{walk}}$, namely a directed graph on V, possibly with self-loops and multiple edges, which contains an edge uv whenever $v = X^{(i)}_{uj}$ for some $j \in [k/2]$.

Let G_i be the undirected version of G_i^*. In proactive, to construct G_i each node u just contacts all of its $\mathcal{F}_i^{\text{walk}}$-neighbors to let them know that they should store a link to u. Note that G_i has a low degree:

Claim 5.9. $\deg(G_i) \leq O(k\, 2^\alpha)$ *with high probability.*

Proof. Condition on $\mathcal{F}_{i-1}$ and consider the probability space induced by $\mathcal{F}_i^{\text{walk}}$. For a given node u, it suffices to bound its in-degree in G_i^*. Note that $vu \in G_i^*$ only if $u \in B_{vi}$ or, equivalently, $v \in B_{ui}$. Each node $v \in B_{ui}$ has k links distributed near-uniformly on B_{vi}. Each of these links lands in u with probability at most $2/|B_{vi}|$, which is at most $2^\alpha/|B_{ui}|$ by Lemma 2.14. The expected in-degree of u in G_i^* is thus at most $k\, 2^\alpha$. The claim follows by Chernoff Bounds since by (P1) all links in G_i^* are independent given $\mathcal{F}_{i-1}$. □

For a given node u, let us define $Q_u = B_{(u,\, i+1)}$. We analyze the induced graph $G_i|Q_u$:

Claim 5.10. *The induced graph* $G_i|Q_u$ *is an* $O(k)$*-degree expander with high probability.*

Proof. Condition on $\mathcal{F}_{i-1}$ and consider the probability space induced by $\mathcal{F}_i^{\text{walk}}$. Each node $v \in Q_u$ has k out-links in G_i^*. Since $Q_u \subset B_{vi}$, by (P2) each of these links lands into a given node $w \in Q_u$ with probability at most $2/|Q_u|$. The expected in-degree of w in $G_i^*|Q_u$ is thus $O(k)$. Since by (P1) all links in G_i^* are independent given $\mathcal{F}_{i-1}$, by Chernoff Bounds the in-degree of $G_i^*|Q_u$ is at most $O(k)$ with high probability, and consequently so is the degree of $G_i|Q_u$. Moreover, by (P3) with high probability the out-degree of $G_i^*|Q_u$ is at least $3\log n$, so by Theorem 2.4 with high probability graph $G_i|Q_u$ is an expander. □

By (P3) with high probability for each node u at least $3\log n$ nodes in $\mathcal{F}_i^{\text{seed}}$ lie inside Q_u. Pick one such node at random, denote it Y_u. For a given node u, let $\mathcal{A}_u$ be the construction in Lemma 5.8 whereby node u acquires the addresses of k near-random nodes. Specifically, we invoke this construction for subset $Q = Q_u$, underlying graph G_i, random seed Y_u, and (by Claims 5.9 and 5.10) upper bounds

$$d = O(k\, 2^\alpha) \text{ and } d_Q = O(k) \text{ and } t = O(k^2 \log n).$$

The overall construction for stage i is simple: each node u invokes algorithm $\mathcal{A}_u$ and thereby acquires the addresses of k nodes in Q_u, not necessarily distinct. Define $X_{uj}^{(i+1)}$ to be the j-th of these nodes. Clearly properties (P1) and (P2) are satisfied. It remains to bound the per-node load.

Let Z_{vu} be the quantity from Lemma 5.8(b), the number of times node v is ”visited” by algorithm $\mathcal{A}_u$. Recall that $Z_{vu} = 0$ whenever $v \notin Q_u$ or, equivalently, when $u \notin Q_v$. Let us define $Z_v = \sum_{u \in V} Z_{vu}$, the total number of times node v is visited by some $\mathcal{A}_u$. Let us bound Z_v:

Claim 5.11. Z_v *is at most* $O(kt\, 2^\alpha)$ *in expectation, and at most* $O(kt\, 2^\alpha \log n)$ *with high probability.*[3]

Proof. Let us condition on $\mathcal{F}_{i-1}$ and $\mathcal{F}_i^{\text{walk}}$ (i.e. let us assume that those are fixed), and let us consider the probability space induced by the random choices in $\mathcal{F}_i^{\text{seed}}$ and in algorithms $\{\mathcal{A}_u : u \in V\}$. By Lemmas 5.8(b) and 2.14, for each $u \in Q_v$ we have

$$E(Z_{vu}) \leq O(kt/|Q_u|) \leq O(kt\, 2^\alpha)/|Q_v|,$$

so $E(Z_v) \leq O(kt\, 2^\alpha)$. Since the random variables $\{Z_{vu} : u \in Q_v\}$ are independent, the claim follows by Chernoff Bounds. □

[3] It seems we can shave off this factor of $\log n$ via a more careful analysis using the variance of Z_{vu}.

Let us fix some node w and partition the total load experienced by node w in a given stage into *direct load* induced on w by algorithms $\mathcal{A}_w$, and *indirect load* induced on w by algorithms $\mathcal{A}_u$, $u \neq w$. By Lemma 5.8(b) the direct load on node w is $O(kdt)$, and the indirect load on w is $O\left(\sum_{wv \in G_i} Z_v\right)$. By Claims 5.9 and 5.11, the latter is at most $O(k^2 t\, 4^\alpha)$ in expectation, and at most

$$T = O(k^2 t\, 4^\alpha \, \log n) = O(2^{10\,\alpha} \, \log^6 n)$$

with high probability. Summing over all stages, the total load is $O(T \log n)$ with high probability.

Let us bound the running time for a given stage. Recall that each message belongs to a particular step of one of the random walks. To simplify the analysis, let us assume that whenever there is contention, messages from earlier steps are given higher priority, and among messages from the same step, a given node u gives higher priority to messages related to algorithm $\mathcal{A}_u$. Via the same analysis as above we can show that during each step a given node receives at most T/t messages. It follows that in time $O(T/t)$ a given node u receives "answers" to all messages sent by a given step of algorithm $\mathcal{A}_u$. Therefore the total running time for a given stage is at most $O(T)$, as required. This completes the proof of Theorem 5.3.

5.4 Network Triangulation: Proof of Theorem 5.5

In this section we prove Theorem 5.5: we will use RRN to provide load-balanced constructions for $(0, \delta)$-triangulation on growth-constrained metrics.

Overview of the algorithm. As in the previous section, our construction proceeds in $\lceil \log \Delta \rceil$ stages so that each stage $i \in [\log \Delta]$ handles distances on the scale of $\Delta/2^i$. First each node selects itself (independently at random) as a *level-i beacon*; we make sure that level-i beacons are sufficiently dense on the scale of δr, and yet sufficiently sparse on the scale of r. Then level-i beacons declare themselves to other nodes via a special broadcast, so that each node (a) finds out about the nearby level-i beacons, (b) forms upper and lower bounds on distances to these beacons. These bounds are not necessarily precise enough to guarantee a sufficiently accurate triangulation. Thus we need an essential *update step* where each level-i beacon b updates the distance estimates to all level $j \leq i-2$ beacons that it knows; this is accomplished by querying all level-$(i-1)$ beacons that b is aware of. For every node a large portion of the load comes from helping other nodes form their estimates. As in the previous section, one needs to be very careful to guarantee that no node is overloaded by helping others.

For this section, we need the ring cardinality k to satisfy the following additional properties:

(P4) Consider some radius r and two nodes u, v at distance at most $\frac{4}{3}\, r$ from each other. Suppose k nodes are chosen independently from a near-uniform distribution on $B_u(2r)$. Then at least one of them lands in $B_v(\frac{2}{3}\, r)$, with failure probability at most $1/n^4$.

(P5) Consider node u and radius r. Suppose each node $v \in B_u(r)$ is selected k times, independently, with probability at least $1/2|B_v(r)|$. Then at least one node in $B_u(r)$ is selected, with failure probability at most $1/n^4$.

These properties is very similar to (P3), but are slightly different quantitatively. By Lemma 2.14 in (P5) we have $|B_v(r)| \leq 2^\alpha |B_u(r)|$, so by Chernoff bounds the minimal k such that (P5) holds is $k_{\text{b}} = O(2^\alpha \, \log n)$; we will use this quantity k_{b} later in the proof. Similarly, in (P4) we have $|B_v(\frac{2}{3}\, r)| \leq 5^\alpha |B_u(\frac{4}{3}\, r)|$, so $k = O(5^\alpha \, \log n)$ suffices.

Consider the construction of RRN in Theorem 5.3, for the ring cardinality $k = c\, 5^\alpha \lceil \log n \rceil$, where the constant c is chosen at least large enough so that all three properties (P3), (P4) and (P5) hold. We describe the RRN by a collection of directed graphs G^*_i, $i \in [\log\Delta]$: we define G^*_i to be the directed graph induced by $\mathcal{F}_i$, namely a directed graph on V, possibly with self-loops and multiple edges, which contains an edge uv, of length $d_{\mathrm{M}}(u,v)$, whenever $v = X^{(i)}_{uj}$ for some $j \in [k]$.

For two nodes u, v and $i \in [\log\Delta]$, a uv-path is *i-telescoping* if it consists of at most $\lceil\log\Delta\rceil$ edges such that (for every j) the j-th edge of this path is in graph G^*_{i+j} and takes us within distance $\frac{4}{3}\Delta/2^{i+j}$ from v. The reason we introduced (P4) is the following simple corollary:

Claim 5.12. *For any $r = \Delta/2^i$, $i \in [\log\Delta]$ and any two nodes u, v at distance at most $\frac{4}{3}\, r$ from each other, there exists an i-telescoping uv-path with high probability.*

The algorithm proceeds in stages $i = 0, 1, 2\,, \ldots, \lceil\log\Delta\rceil$. Informally, a given stage i handles distance scale $r = \Delta/2^i$. without loss of generality assume $\delta \leq \frac{2}{9}$, let δ be an integer power of two, and let $i_0 = i - \log\delta$. Each stage consists of three steps.

First step. In the first step, beacons are selected: each node u selects itself as an *level-i beacon* independently with probability close to $k_{\mathrm{b}}/|B_u(\delta r)|$. Selection is implemented via random walks: we piggy-back on the construction of RRN. Specifically, we set aside k_{b} neighbors $X^{(i_0)}_{uj}$ in the i_0-th ring of u, and we 'select' if and only if one of these neighbor is u itself.

The sole objective of beacon selection is to ensure that level-i beacons provide a good coverage on the scale of δr, and yet are relatively sparse on the scale of r; the former is used to prove accuracy, and the latter is used to bound load.

Claim 5.13. *For each node u, with high probability (a) there is at least one level-i beacon in $B_u(\delta r)$, and (b) there are at most $O(k_{\mathrm{b}})\;(8/\delta)^\alpha$ level-i beacons in $B_u(4r)$.*

Proof. Part (a) follows by (P5). Part (b) follows by Chernoff bounds since by Lemma 2.14 for any node $v \in B_u(2r)$ we have $|B_u(2r)| \leq (4/\delta)^\alpha\, |B_v(\delta r)|$. □

Second step. In the second step, level-i beacons declare themselves to other nodes via a special broadcast. This broadcast will involve at most $1 + \lceil\log\Delta\rceil$ types of packets, numbered from 0 to $\lceil\log\Delta\rceil$. Let $P_b(j)$ be a type-j broadcast packet from the special broadcast initiated by beacon b. Each beacon b initiates his broadcast by sending packet $P_b(i)$ to all its ring-$(i-1)$ neighbors. Each node stores a list of received broadcast packets (without duplicates). Suppose a given node receives a type-j packet $P_b(j)$. If $j < \lceil\log\Delta\rceil$ and this node has not seen such packet before, it sends packet $P_b(j+1)$ to all its ring-j neighbors; else it does nothing.

As a result of these broadcasts, each node u acquires the list S_{ui} of i-level beacons whose broadcasts it has received. For each beacon $b \in S_{ui}$ node u maintains upper and lower bounds on $d_{\mathrm{M}}(u,b)$, denoted $D^+(u,b)$ and $D^-(u,b)$, and initialized to, respectively, ∞ and 0. They are updated in the third step using distances between beacons. We will show that eventually we construct good estimates on distances to all i-level beacons within distance $(1+\delta)\, r$ from u.

The special broadcast described above has the following properties:

Claim 5.14. *Consider the broadcast started by a level-i beacon b. This broadcast (a) reaches all nodes in $B_b(\frac{4}{3}\, r)$, (b) stays inside the ball $B_b(4r)$, (c) induces per-node load at most $O(k\, 2^\alpha\, \log n)$.*

Proof. For part (a) note that for each node $u \in B_b(\frac{4}{3}\, r)$ the broadcast from beacon b follows each i-telescoping bu-path; by Claim 5.12 at least one such path exists. For part (b), we prove by induction on j that packet $P_b(j)$ stays within distance $4r(1 - 2^{i-j-1})$ from beacon b. For part (c), recall that the in-degree of G^*_j is $O(k\, 2^\alpha)$ (see the proof of Claim 5.9), and note that a given node can receive a given packet $P_b(j)$ only from its in-neighbors in G^*_{j-1}, and only once from each. □

Claim 5.15. *Consider the special broadcasts from all level-i beacons. The per-node load is at most $O(\log^4 n)(80/\delta)^\alpha$, and the total running time is at most $O(\log^5 n)(80/\delta)^\alpha$.*

Proof. By Claim 5.14(b) a given node u receives broadcasts only from beacons within distance $4r$ from u. By Claim 5.13 there are at most $O(k_{\mathrm{b}})\,(8/\delta)^\alpha$ such beacons, so by Claim 5.14(c) the load on node u is at most

$$L = O(k\,k_{\mathrm{b}}\,\log n)(16/\delta)^\alpha = O(\log^4 n)(80/\delta)^\alpha.$$

Let us bound the running time by $O(L \log n)$ via the following rather crude argument. Recall that a given node sends a given packet $P_b(j)$ at most once. Say the special broadcasts start at time 0. By induction on j, we claim that by time $t_j = O(L)(j-i+1)$ all packets $P_b(j)$ are sent and received. Indeed, if this is true for some j, then after time t_j a given node knows all packets $P_b(j+1)$ that it needs to send out, so it keeps sending them unless it needs to pause and receive some other packet. It will be done by time $t_j + O(L)$ since it can receive at most L packets. □

Remark. Assuming a minimal synchronization, namely that first all packets $P_b(i)$ are sent and received, then all packets $P_b(i+1)$, then all packets $P_b(i+2)$ and so on, a given node can aggregate all packets $P_b(j)$ that it sends (for a given j) into a very small number of packets. Then both the load and the running time become $O(k\,2^\alpha\,\log n) = O(10^\alpha\,\log^2 n)$.

Third step. In the third step (for $i > 0$), each level-i beacon b measures distances to all level-$(i-1)$ beacons that it knows, and for each $j \leq i-2$ updates distance estimates to all level-j beacons that it knows. Specifically, each beacon $b' \in S_{(b,\,i-1)}$ beacon b measures distance to b', and receives from b' its distance estimates $D^\pm(b', b^*)$ for each level-j beacon $b^* \in S_{(b',\,j)}$. Then whenever this level-j beacon b^* also lies in $S_{(b,\,j)}$, beacon b updates its distance estimates $D^\pm(b, b^*)$:

$$\begin{cases} D^+(b,b^*) & \leftarrow \quad \min\left(D^+(b,b^*),\ D^+(b',b^*) + d_{\mathrm{M}}(b,b')\right) \\ D^-(b,b^*) & \leftarrow \quad \max\left(D^-(b,b^*),\ D^-(b',b^*) - d_{\mathrm{M}}(b,b')\right). \end{cases} \tag{5.4}$$

This completes the description of the algorithm.

A straightforward corollary of the update rule (5.4) is that $D^\pm$ are indeed upper/lower bounds:

Claim 5.16. *For any two nodes u, v we have $D^-(u,v) \leq d_M(u,v) \leq D^+(u,v)$ at any point in the execution of the algorithm.*

Note that in the last stage, every node is a level-$\lceil \log \Delta \rceil$ beacon with high probability. In particular, each node u forms bounds $D^\pm(u,b)$ for every beacon $b \in \cup_j S_{uj}$, which form the node label in triangulation. By Claim 5.14(b) and 5.13 the degree of this triangulation is

$$|\cup_j\, S_{uj}| \leq O(k_{\mathrm{b}}\,\log n)(8/\delta)^\alpha = O(\log^2 n)\,(16/\delta)^\alpha.$$

It remains to show that our triangulation obtains the desired precision. To this end, let us first consider the distances to beacons:

Lemma 5.17. *Fix level $i \in [\log \Delta]$ and let $r = \Delta/2^i$. Then for each node u and each level-i beacon $b \in B_u((1+\delta)\,r)$ the bounds $D^\pm(u,b)$ are off from $d_M(u,b)$ by at most the additive factor of $2\delta r$.*

Proof. By Claim 5.13(a) for every level $j \in [\log \Delta]$ there exists a level-j beacon $b_j \in B_u(\delta r/2^{j-i})$.

First, recall that we assume $\delta \leq 2/9$, and so for each $j > i$ we have

$$d_{\mathrm{M}}(b, b_j) \leq d_{\mathrm{M}}(u,b) + d_{\mathrm{M}}(u,b_j) = r(1+\delta) + \delta r/2 \leq 4r/3.$$

By Claim 5.14(a) it follows that each b_j, $j > i$ receives the broadcast from beacon b. In particular, beacon b_{i+1} measures the distance to b.

Second, note that for each j

$$d_{\mathrm{M}}(b_j, b_{j+1}) \leq d_{\mathrm{M}}(u, b_j) + d_{\mathrm{M}}(u, b_{j+1}) \leq 3\delta r/2^{j+1-i} \leq r/2^{j-i},$$

so by Claim 5.14(a) beacon b_{j+1} receives the broadcast from b_j and, consequently, measures the distance to b_j. Now by induction on j we can show that each beacon b_j, $j \geq i+1$ forms bounds $D^{\pm}(b_j, b)$ that are at least as good as

$$D^{\pm}(b_j, b) = d_{\mathrm{M}}(b, b_{i+1}) \pm \sum_{l=i+1}^{j-1} d_{\mathrm{M}}(b_l, b_{l+1}).$$

Finally, recall that with high probability node u is a level-j beacon for $j = \lceil \log \Delta \rceil$. By the above equation,

$$D^{+}(u, b) \leq d_{\mathrm{M}}(u, b) + 2 \sum_{l=i+1}^{j} d_{\mathrm{M}}(u, b_l) \leq d_{\mathrm{M}}(u, b) + 2 \sum_{l=i+1}^{j} \delta\Delta/2^l \leq d_{\mathrm{M}}(u, b) + 2\delta r,$$

and similarly $D^{-}(u, b) \geq d_{\mathrm{M}}(u, b) - 2\delta r$ as required. □

Now we use Claim 5.14(a) and Lemma 5.17 to prove the desired accuracy.

Lemma 5.18. *For any two nodes* u, v *we have* $D^{+}(u, v)/D^{-}(u, v) \leq 1 + O(\delta)$.

Proof. Let us consider the distance scale i defined as the smallest i such that $r := \Delta/2^i \geq d_{\mathrm{M}}(u, v)$. By Claim 5.14(a) there exists an i-level beacon $b \in B_v(\delta r)$. Then $d_{\mathrm{M}}(u, b) \leq r(1+\delta)$, so by Lemma 5.17 both $D^{\pm}(u, b)$ and $D^{\pm}(v, b)$ are off from their respective true values by no more than the additive factor of $2\delta r$. It follows that

$$D^{+}(u, v) \leq D^{+}(u, b) + D^{+}(v, b) \leq d_{\mathrm{M}}(u, b) + d_{\mathrm{M}}(v, b) + 4\delta r \leq d_{\mathrm{M}}(u, v) + 6\delta r,$$

and similarly $D^{-}(u, v) \geq d_{\mathrm{M}}(u, v) - 6\delta r$. □

This completes the proof of Theorem 5.5.

Chapter 6

Location-aware node selection via Rings of Neighbors

In this chapter we discuss our work on *Meridian* [WSS05], a framework for performing node selection based on network location.[1] *Meridian* is a lightweight, scalable, and accurate system for keeping track of location information for participating nodes that does not require computing virtual coordinates a la [NZ02, DCKM04]. The system is simple, loosely-structured, and entails modest resources for maintenance. It can efficiently find the closest node to a target, the latency minimizing node to a given set of nodes, and the set of nodes that lie in a region defined by latency constraints, which are frequently encountered building block operations in many location-sensitive distributed systems. Although less general than virtual coordinates, *Meridian* is significantly more accurate.

In Section 6.1 we outline the system, and then proceed with the analysis that shows that Meridian provides robust performance, high scalability and good load balance. We include several figures describing the functioning of the system and some of the experimental results, but for a much more comprehensive description of the system and experimental results, see the original paper [WSS05] and the forthcoming thesis work of Bernard Wong.

6.1 Meridian: a framework for location-aware node selection

Selecting nodes based on their location in the network is a basic building block for many high-performance distributed systems. In small systems, it is possible to perform extensive measurements and make decisions based on global information. For instance, in an online game with few servers, a client can simply measure its latency to all servers and bind to the closest one for minimal response time. However, collecting global information is infeasible for a significant set of recently emerging large-scale distributed applications, where global information is unwieldy and lack of centralized servers makes it difficult to find nodes that fit selection criteria. Yet many distributed applications, such as filesharing networks, content distribution networks, backup systems, anonymous communication networks, pub-sub systems, discovery services, and multiplayer online games could benefit substantially from selecting nodes based on their location in the network.

A general technique for finding nodes that optimize a given network metric is to perform a *network embedding*, that is, to map high-dimensional network measurements into a location in a smaller Euclidean

[1] We use the term "location" to refer to a node's placement in the Internet as defined by its round-trip latency to other nodes. While we do not assume that there is a well-defined location for any node, our illustrations depict a single point in a two-dimensional space for clarity.

space. For instance, recent work in network positioning [NZ02, NZ04, DCKM04, LHC03, TC03, ST03, PCW$^+$03, CCRK04, LL04] uses large vectors of node-to-node latency measurements on the Internet to determine a corresponding single point in a d-dimensional space for each node. The resulting embedded address, a *virtual coordinate*, can be used to select nodes.

While the network embedding approach is applicable for a wide range of applications, it is neither accurate nor complete. The embedding process typically introduces significant errors. Selection of parameters, such as the constant d, the set of measurements taken to perform the embedding, the landmarks used for measurement, and the timing interval in which measurements are taken, is nontrivial and has a significant impact on the accuracy of the approach. Further, coordinates need to be recomputed as network latencies fluctuate. In addition, complex mechanisms besides virtual coordinates are required to support large-scale applications. Simple schemes, such as centralized servers that retain $O(N)$ state or naive algorithms with $O(N)$ running time, are unsuitable for large-scale networks. Peer-to-peer substrates that can naturally work with Euclidean coordinates and support range queries, such as CAN [RFH$^+$01], Mercury [BAS04] and P-Trees [CLGS04], can reduce the state requirements per node; however, these systems introduce substantial complexity and bandwidth overhead in addition to the overhead of network embedding. The simulation results in [WSS05] show that, even with a P2P substrate that always finds the best node based on virtual coordinates, the embedding error leads to a suboptimal choice.

In [WSS05] we introduce a lightweight, scalable and accurate framework, called Meridian, for performing node selection based on network location. Meridian forms a loosely-structured overlay network, uses direct latency measurements instead of latency estimates from virtual coordinates, and can solve spatial queries without an absolute coordinate space.

The basic Meridian framework is based around three mechanisms: a loose routing system based on multi-resolution rings on each node, an adaptive ring membership replacement scheme that maximizes the usefulness of the nodes populating each ring, and a gossip protocol for node discovery and dissemination.

Multi-resolution rings. Each Meridian node keeps track of a small, fixed number of other nodes in the system, and organizes this list of peers into concentric, non-overlapping rings. The ith ring has inner radius $r_i = \alpha s^{i-1}$ and outer radius $R_i = \alpha s^i$, for $i > 0$, where α is a constant, s is the multiplicative increase factor, and $r_0 = 0$, $R_0 = \alpha$ for the innermost ring. Each node keeps track of a finite number of rings; all rings $i > i^*$ for a system-wide constant i^* are collapsed into a single, outermost ring that spans the range $[\alpha s^{i^*}, \infty]$.

Meridian nodes measure the distance d_j to a peer j, and place that peer in the corresponding ring i such that $r_i < d_j \leq R_i$. This sorting of neighbors into concentric rings is performed independently at each node and requires no fixed landmarks or distributed coordination. Each node keeps track of at most k nodes in each ring and drops peers from overpopulated rings. Consequently, Meridian's space requirement per node is proportional to k. We later show in the analysis that a choice of $k = O(\log N)$ can resolve queries in $O(\log N)$ lookups; the simulations verify that a small k suffices. We assume that every participating node has a rough estimate of $\log N$.

The ring structure with its exponentially increasing ring radii favors nearby neighbors, enabling each node to retain a relatively large number of pointers to nodes in their immediate vicinity. This allows a node to authoritatively answer geographic queries for its region of the network. At the same time, the ring structure ensures that each node retains a sufficient number of pointers to remote regions, and can therefore dispatch queries towards nodes that specialize in those regions. An exponentially increasing radius also makes the total number of rings per node manageably small.

Ring membership management. The number of nodes per ring, k, represents an inherent tradeoff between accuracy and overhead. A large k increases a node's information about its peers and helps it make

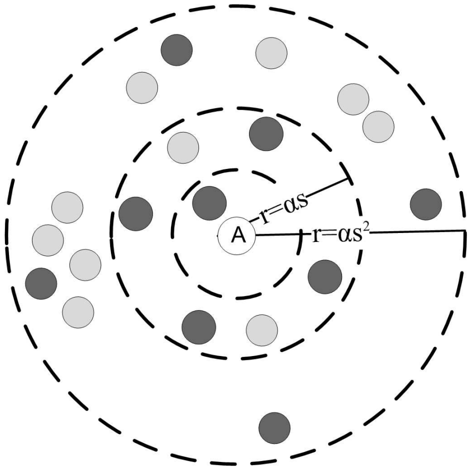

Figure 6.1: Each Meridian node keeps track of a fixed number of other nodes and organizes these nodes into concentric, non-overlapping rings of exponentially increasing radii.

better choices when routing queries. On the other hand, a large k also entails more state, more memory and more bandwidth at each node.

Within a given ring, node choice can have a significant effect on the performance of the system. A set of ring members that are geographically distributed provides much greater utility than a set of ring members that are clustered together, as shown in Figure 6.1. Intuitively, nodes that are geographically diverse instead of clustered together enable a node to forward a query to a greater region. Consequently, Meridian strives to promote geographic diversity within each ring.

Meridian achieves geographic diversity by periodically reassessing ring membership decisions and replacing ring members with alternatives that provide greater diversity. Within each ring, a Meridian node not only keeps track of the k primary ring members, but also a constant number l of secondary ring members, which serve as a FIFO pool of candidates for primary ring membership.

We quantify geographic diversity through the hypervolume of the k-polytope formed by the selected nodes. To compute the hypervolume, each node defines a local, non-exported coordinate space. A node i will periodically measure its distance d^i_j to another node j in the same ring, for all $0 \leq i, j \leq k + l$. The coordinates of node i consist of the tuple $\langle d^i_1, d^i_2, ..., d^i_{k+l} \rangle$, where $d^i_i = 0$. This embedding is trivial to construct and does not require a potentially error-introducing mapping from high-dimensional data to a lower number of dimensions.

Having computed the coordinates for all of its members in a ring, Meridian nodes then determine the subset of k nodes that provide the polytope with the largest hypervolume. For small k, it is possible to determine the maximal hypervolume polytope by considering all possible polytopes from the set of $k + l$ nodes. For large $k + l$, evaluating all subsets is infeasible. Instead, Meridian uses a greedy algorithm: A node starts out with the $k+l$ polytope, and iteratively drops the vertex (and corresponding dimension) whose absence leads to the smallest reduction in hypervolume until k vertices remain. The remaining vertices are designated the new primary members for that ring, while the remaining l nodes become secondaries. This computation can be performed in linear time using standard computational geometry tools [BDH96]. The ring membership management occurs in the background and its latency is not critical to the correct operation of Meridian. Note that the coordinates computed for ring member selection are used only to select a diverse set of ring members; they are not exported by Meridian nodes and play no role in query routing.

Churn in the system can be handled gracefully by the ring membership management system due to the loose structure of the Meridian overlay. If a node is discovered to be unreachable during the replacement process, it is dropped from the ring and removed as a secondary candidate. If a peer node is discovered to be unreachable during gossip or the actual query routing, it is removed from the ring, and replaced with a random secondary candidate node. The quality of the ring set may suffer temporarily, but will be corrected by the next ring replacement. Discovering a peer node failure during a routing query can reduce query performance; k can be increased to compensate for this expected rate of failure.

Gossip-based node discovery. The use of a gossip protocol to perform node discovery allows the Meridian overlay to be loosely connected, highly robust and inexpensively kept up-to-date of membership changes. Our gossip protocol is based on an anti-entropy push protocol [DGH$^+$87] that implements a membership service. The central goal of our gossip protocol is for each node to discover and maintain a small set of pointers to a sufficiently diverse set of nodes in the network. Our gossip protocol works as follows:

1. Each node A randomly picks a node B from each of its rings and sends a gossip packet to B containing a randomly chosen node from each of its rings.

2. On receiving the packet, node B determines through direct probes its latency to A and to each of the nodes contained in the gossip packet from A.

3. After sending a gossip packet to a node in each of its rings, node A waits until the start of its next gossip period and then begins again from step 1.

In step 2, node B sends probes to A and to the nodes in the gossip packet from A regardless of whether B has already discovered these nodes. This re-pinging ensures that stale latency information is updated, as latency between nodes on the Internet can change dynamically. The newly discovered nodes are placed on B's rings as secondary members.

For a node to initially join the system, it needs to know the IP address of one of the nodes in the Meridian overlay. The newly joining node contacts the Meridian node and acquires its entire list of ring members. It then measures its latency to these nodes and places them on its own rings; these nodes will likely be binned into different rings on the newly joining node. From there, the new node participates in the gossip protocol as usual.

The period between gossip cycles is initially set to a small value in order for new nodes to quickly propagate their arrival to the existing nodes. The new nodes gradually increase their gossip period to the same length as the existing nodes. The choice of a gossip period depends on the expected rate of latency change between nodes and expected churn in the system.

Maintenance overhead. The average bandwidth overhead to maintain the multi-resolution rings of a Meridian node is modest. The number of gossip packets a node receives is equal to the number of neighbors $(m \log N)$ multiplied by the probability of being chosen as a gossip target by one of the neighbors $(\frac{1}{\log N})$, where m is the number of rings in the ring-set. A node should therefore expect to send and receive m gossip packets and to initiate m^2 probes per gossip period. A node is also the recipient of probes from neighbors of its neighbors. Since it has $m \log N$ neighbors, each of which sends m gossip packets, there are $m^2 \log N$ gossip packets with a $\frac{1}{\log N}$ probability of containing a reference to it. Therefore, a node expects to receive m^2 probes from neighbors of its neighbors. Assuming $m = 9$, a probe packet size of 50 bytes, two packets per probe, and a gossip packet size of 100 bytes, membership dissemination consumes an average of 20.7 KB/period of bandwidth per node. For a gossip period of 60 seconds, the average overhead associated with gossip is 345 B/s, and is independent of system size.

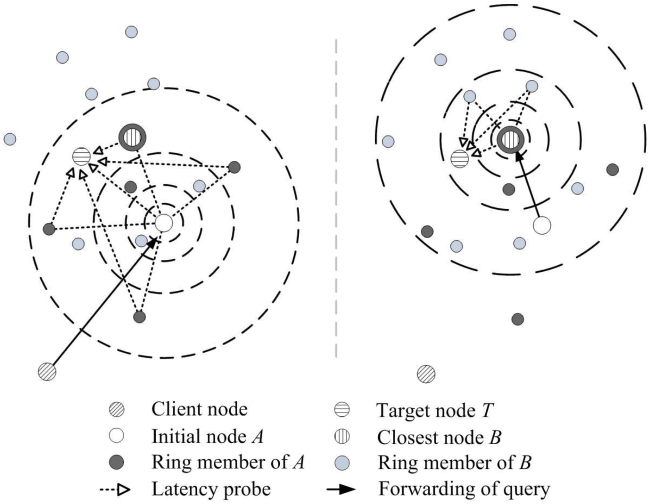

Figure 6.2: A client sends a "closest node discovery to target T" request to a Meridian node A, which determines its latency d to T and probes its ring members between $(1-\beta)\cdot d$ and $(1+\beta)\cdot d$ to determine their distances to the target. The request is forwarded to the closest node thus discovered, and the process continues until no closer node is detected.

There is also maintenance overhead for performing ring management. In every ring management period where the membership of one ring is re-evaluated, $2\log N$ requests are sent, $2\log N$ are received, $4\log^2 N$ probes are sent, and $4\log^2 N$ are received. Assuming two packets are necessary per request and per probe, the size of a probe request packet is 100 bytes and a probe packet is 50 bytes, and a 2000 node system with 16 nodes per ring, ring management consumes an average of 218 KB/period. For a ring management period of 5 minutes, the average overhead associated with ring management is 727 B/s. This analysis conservatively assumes that all primary and secondary rings of all nodes are full, which is unlikely in practice.

Application: closest node discovery. Meridian locates the closest node by performing a multi-hop search where each hop exponentially reduces the distance to the target. This is similar to searching in structured peer-to-peer networks such as Chord [SMK$^+$01], Pastry [RD01] and Tapestry [ZHR$^+$04], where each hop brings the query exponentially closer to the destination, though in Meridian the routing is performed using physical latencies instead of numerical distances in a virtual identifier space. Another important distinction that Meridian holds over the structured peer-to-peer networks is the target node need not be part of the Meridian overlay. The only requirement is that the latencies between the nodes in the overlay and the target node are measurable. This enables applications such as finding the closest node to a public web server, where the web server is not directly controlled by the distributed application and only responds to HTTP queries.

When a Meridian node receives a request to find the closest node to a target, it determines the latency d between itself and the target. Once this latency is determined, the Meridian node simultaneously queries all of its ring members whose distances are within $(1-\beta)\cdot d$ to $(1+\beta)\cdot d$. These nodes measure their distance to the target and report the result back to the Meridian node. Nodes that take more than $(2\beta+1)\cdot d$ to provide an answer are ignored, as they are more than βd away from the target.

Meridian uses an acceptance threshold β, which determines the reduction in distance at each hop. The route acceptance threshold is met if one or more of the queried peers is closer than β times the distance to the target, and the client request is forwarded to the closest node. If no peers meet the acceptance threshold,

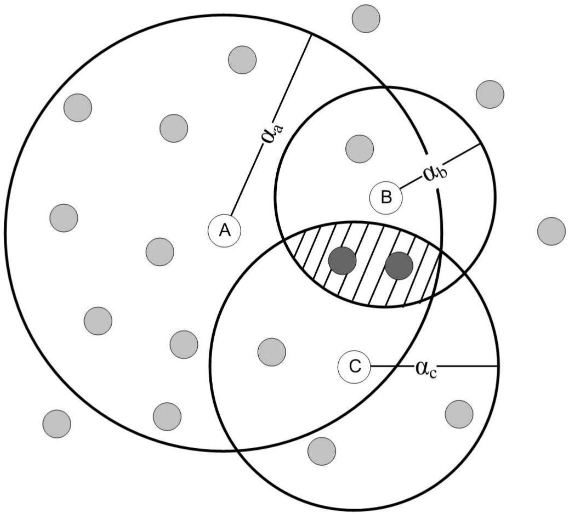

Figure 6.3: A multi-constraint query consisting of targets A, B, C with respective latency constraints of $\alpha_a, \alpha_b, \alpha_C$. The shaded area represents the solution space.

then routing stops and the closest node currently known is chosen. Figure 6.2 illustrates the process.

Meridian is agnostic to the choice of a route acceptance threshold β, where $0 \leq \beta < 1$. A small β value reduces the total number of hops, as fewer peers can satisfy the requirement, but introduces additional error as the route may be prematurely stopped before converging to the closest node. A large β reduces error at the expense of increased hop count.

Application: central leader election. Another frequently encountered problem in distributed systems is to locate a node that is "centrally situated" with respect to a set of other nodes. Typically, such a node plays a specialized role in the network that requires frequent communication with the other members of the set; selecting a centrally located node minimizes both latency and network load. An example application is leader election, which itself is a building block for higher level applications such as clustering and low latency multicast trees.

The central leader election application can be implemented by extending the closest node discovery protocol. We replace d in the single target closest node selection protocol with d_{avg} for central leader election. When a Meridian node receives a client request to find the closest node to the target set T, it determines the latency set $\{d_1, ..., d_{|T|}\}$ between itself and the targets through direct measurements, and computes the average latency $d_{avg} = (\sum_{i=1}^{|T|} d_i)/|T|$. It selects ring members that have latency between $(1 - \beta) * \min\{d_1, ..., d_{|T|}\}$ and $(1 + \beta) * \max\{d_1, ..., d_{|T|}\}$ to itself, and requests these peers to.determine their respective average latency to the targets. The remaining part of the central leader election application follows exactly from the closest node discovery protocol.

Changing the latency aggregation function from taking the average of the latencies to the highest latency target is a useful variation to the protocol, as it reduces the difference in latency between the targets to the chosen node. This is useful in multi-player online games, as a player with a significantly lower latency to the game server than the others has an unfair advantage because it is the first to receive and react on game events.

Application: multi-constraint queries. Another frequent operation in distributed systems is to find a set of nodes satisfying constraints on the network geography. For instance, an ISP or a web hosting service is typically bound by a service level agreement (SLA) to satisfy latency requirements to well-known peering locations when hosting services for clients. A geographically distributed ISP may have thousands of nodes at its disposal, and finding the right set of nodes that satisfy the given constraints may be necessary for fulfilling an SLA. Latency constraints are also important for grid based distributed computation applications, where the latency between nodes working together on a problem is often the main efficiency bottleneck. A customer may want to specify that $\forall q, p \in P$ where P is the set of grid nodes, $d_{q,p} < \gamma$ for some desired latency γ.

Finding a node that satisfies multiple constraints can be viewed as a node selection problem, where the constraints define the boundaries of a region in space (the solution space), as illustrated in Figure 6.3. A constraint is specified as a target and a latency bound around that target. When a Meridian node receives a multi-constraint query with u constraints specified as $\langle \mathit{target}_i, \mathit{range}_i \rangle$, for all $0 < i \leq u$, it measures its latency d_i to the target nodes and calculates its distance to the solution space as

$$s = \sum_{i=1}^{u} \max(0, d_i - \mathit{range}_i)^2$$

If s is 0, then the current node satisfies all the constraints, and it returns itself as the solution to the client. Otherwise, it iterates through all its peers, and simultaneously queries all peers j that are within $\max(0, (1 - \beta) \cdot (d_i - \mathit{range}_i))$ to $(1 + \beta) \cdot (d_i + \mathit{range}_i)$ from itself, for all $0 < i \leq u$. These nodes include all the peers that lie within the range of at least one of the constraints, and possibly other peers that do not satisfy any of the constraints, but are nevertheless close to the solution space. These peer nodes measure their distance to the u targets and report the results back to the source. Nodes that take longer than $\max_{0<i\leq u}((2\beta + 1) \cdot (d_i + \mathit{range}_i))$ to provide an answer are ignored.

The distance s_j of each node j to the solution space is calculated using the metric s defined above. If s_j is 0, then node j satisfies all the constraints and is returned as a solution to the client. If no zero valued s_j is returned, the client determines whether there is an $s_j < \beta \cdot s$, where β is the route acceptance threshold. If the route acceptance threshold is met, the client request is forwarded to the peer closest to the solution space. A larger β may increase the success rate, at the expense of increased hops.

6.2 Analysis of scalability

In this section we argue analytically that Meridian scales well with the size of the system. Our contributions are three-fold. First, we put forward a rigorous definition that captures the quality of Meridian ring sets, and prove that under certain reasonable assumptions small ring cardinalities suffice to ensure good quality. Second, we show that with these good-quality rings, our algorithms for nearest neighbor selection and central leader election work well, returning near-exact neighbors and central leaders respectively. We provide further results on *exact* nearest neighbors. Finally, we argue that if the ring sets of different nodes are stochastically independent then the system is load-balanced.

We model the matrix of Internet latencies as a metric, i.e. a symmetric function obeying the triangle inequality. We should not hope to achieve theoretical guarantees for *arbitrary* metrics; we need some reasonable assumptions to capture the properties of real-life latencies. We avoid assumptions on the *geometry* of the metric such as assuming it is Euclidean for two reasons. Firstly, recent experimental results suggest that approximating Internet latencies by Euclidean metrics, although a useful heuristic in some cases, incurs significant relative errors [NZ02, DCKM04, LHC03, TC03, ST03, PCW$^+$03, CCRK04, NZ04, LL04]. Secondly, and perhaps more importantly, even if we assume that the metric is Euclidean our algorithm is not allowed to use the coordinates since one of the goals of this work is precisely to avoid heavy-weight embedding-based approaches.

We will consider two families of metrics that have been popular in the recent theoretical literature as non-geometric notions of low-dimensionality: *growth-constrained* metrics and *doubling* metrics.

We focus on the case when the rate of churn and fluctuations in Internet latencies is sufficiently low so that Meridian has ample time to adjust. So for the purposes of this analysis we assume that the node set and the latency matrix do not change with time.

This section is organized as follows. We start with a formal description of the Meridian framework (Section 6.2.1). We proceed with a section on the quality of Meridian rings (Section 6.2.2). Then we analyze the performance our search algorithms (Section 6.2.3), with extensions to exact nearest neighbors (Section 6.2.4) and load-balancing (Section 6.2.5). We conclude with some directions in which our results can be fine-tuned (Section 6.2.6). To improve the flow of the section, some of the more involved proofs are moved to the next three sections.

6.2.1 Formal description of the Meridian framework

Let V be the set of all nodes in the system. Nodes running Meridian are called *Meridian nodes*. Let $S_M \subset V$ be the set of Meridian nodes, of size N. Let d be the distance function on V induced by the node-to-node latencies: $d(u,v)$ is the uv-distance, i.e. the latency between nodes u and v. Sometimes, when this is typographically convenient, we may also denote it as d_{uv}.

Let $B_u(r)$ denote the closed ball in S_M of radius r around node u, i.e. the set of all Meridian nodes within distance r from u. Define $B_{ui} = B_u(2^i)$ and $R_{ui} = B_{ui} \setminus B_{(u,i-1)}$. Then R_{ui}'s are disjoint concentric rings around u. Without loss of generality let the smallest distance be 1; denote the maximal distance by Δ.

Throughout this section we will denote the maximal number of nodes in a Meridian ring by k. Formally, for some fixed k every node u maintains $\log(\Delta)$ sets $S_{ui} \subset B_{ui}$, $0 \leq i \leq \lceil \log \Delta \rceil$ of at most k nodes each. These sets are called *m-rings* of u ('m' stands for 'Meridian'), and the nodes in these sets are called *Meridian neighbors* of u. If $|R_{ui}| \geq k$ then the corresponding m-ring S_{ui} consists of exactly k nodes that lie in ring R_{ui}. If $|R_{ui}| < k < |B_{ui}|$ then S_{ui} consists of all nodes in R_{ui}. Finally, if $|B_{ui}| \leq k$ then S_{ui} consists of all nodes in ball B_{ui}.

Let us make some remarks about the above definition. Note that each m-ring S_{ui} contains all Meridian neighbors of u that lie in ring R_{ui}. For a fixed Meridian node u, let i_0 be the largest i such that $B_{ui} \leq k$, and let i_1 be the largest i such that $R_{ui} \leq k$. Then the m-rings S_{ui}, $i \leq i_1$ are fixed by the above definition, whereas the m-rings S_{ui}, $i > i_1$ are not. Also, in the implementation we do not need to maintain m-rings S_{ui}, $i \leq i_0$ explicitely; we define them here for the convenience of the analysis.

Let us formally define the nearest-neighbor search algorithm used in Meridian. Suppose a node u receives a query to a target node t. Then u measures the distance d_{ut} and looks at the three m-rings $S_{(u,i-1)}$, S_{ui} and $S_{(u,i+1)}$, where $i = \lceil \log d_{ut} \rceil$; let S be the union of these rings. All nodes in S measure their distance to t and report their measurements to u. Then u forwards the query to the node $w \in S$ that is closest to the target t subject to the constraint that $d_{ut}/d_{wt} \leq \beta_0$. This constitutes one step of the algorithm.

If such w does not exist, the algorithm chooses the node in $S \cup \{u\}$ that is closest to t, call it w', reports this node to the node that initiated the query, and stops; in this case we say that as a result of the query, our algorithm *finds* w'. Here $\beta_0 > 1$ is a parameter that is the same for all nodes that handle a given query. We denote this algorithm by $\mathcal{A}(\beta_0)$.

For the sake of the analysis we will also consider a version of $\mathcal{A}(\beta_0)$ where instead of looking at three m-rings we look at all m-rings S_{ui}, $i \leq 1 + \lceil \log d_{ut} \rceil$. We denote this version by $\mathcal{A}^*(\beta_0)$.

Let us define the approximation ratio γ for nearest neighbor selection algorithms. Consider a node t and let v be its nearest neighbor. Say node u is a γ*-approximate* nearest neighbor of t if $d_{ut}/d_{vt} \leq \gamma$. An

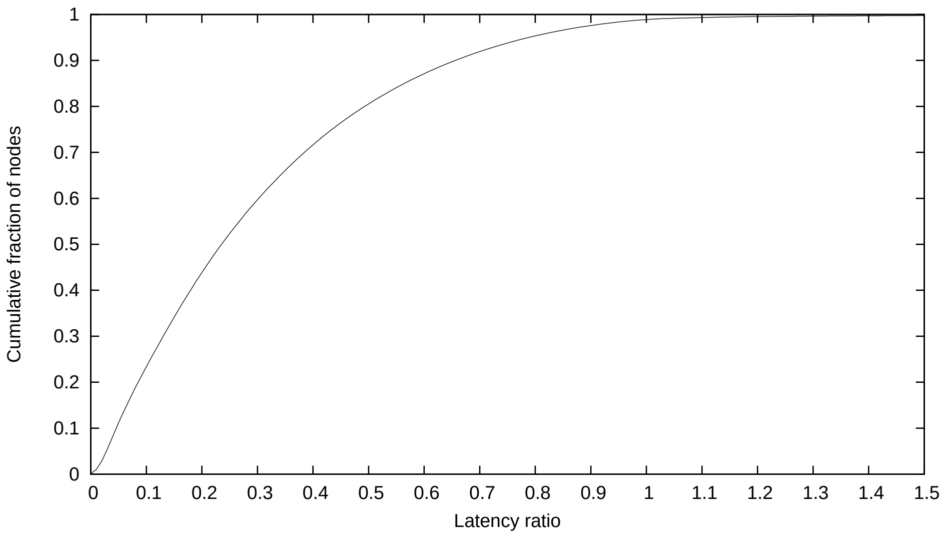

Figure 6.4: The efficiency of ring member selection can be measured through *latency ratio*: for a given ordered pair (u, v) of Meridian nodes, this ratio is defined as $d(w, v)/d(u, v)$, where w is the Meridian neighbor of u that is closest to v. In particular, for 80% node pairs the latency ratio is less than .5.

algorithm is γ-*approximate* if for any target it finds a γ-approximate nearest neighbor.

It is straightforward to generalize the algorithms $\mathcal{A}(\cdot)$ and $\mathcal{A}^*(\cdot)$ to the central leader election problem. Namely, given a set T of targets, we simply replace d_{ut}, the distance to from the current node u to target t, by the average distance from u to targets in T. Note that we are back to the nearest neighbor selection problem if $|T| = 1$. The concept of approximation ratio generalizes similarly.

6.2.2 Quality of the Meridian rings

Intuitively, we want each m-ring S_{ui} to cover the corresponding ring R_{ui} reasonably well: we want each node in R_{ui} to be within a small distance from some node in S_{ui}. For technical reasons in order to cover R_{ui} we might also need some Meridian neighbors from $S_{(u,i-1)}$ or $S_{(u,i+1)}$. We formalize the 'goodness' of m-rings is as follows:

Definition 6.1. Say the Meridian rings are ϵ-*nice*, $\epsilon < 1$, if for any two Meridian nodes $u, v \in S_M$ node u has a Meridian neighbor w such that $d(w, v) \leq \epsilon\, d(u, v)$.

In the above definition $v \in R_{ui}$ for $i = \lceil \log d_{uv} \rceil$. Since $2^{i-2} < d_{uw} < 2^{i+1}$, node w is indeed contained in one of the three m-rings $S_{(u,i-1)}$, S_{ui}, $S_{(u,i+1)}$ that are considered by algorithm $\mathcal{A}(\cdot)$.

In Section 6.2.3 we will how that under Definition 6.1, the Meridian search algorithm achieves good approximation guarantees. Later in this section we show that even for small cardinalities of m-rings it is possible to make them ϵ-nice; this is confirmed by the empirical evidence (see Figure 6.4).

Probabilistic interpretation. To show that the m-rings are indeed ϵ-nice, recall that the m-rings are constructed by an underlying randomized gossiping protocol. For each m-ring S_{ui}, this protocol induces a probability distribution over subsets of S_M, so we can treat S_{ui} as a random variable (whose values are subsets of S_M). In particular, we can talk about the distribution of a given m-ring. A natural and intuitively appealing distribution for an m-ring S_{ui} is that of a random k-node subset of the corresponding ring R_{ui}. Let us formalize this:

Definition 6.2. S_{ui} is *well-formed* if its distribution is that of a random k-node subset of R_{ui}, or if $|R_{ui}| \leq k$.

We proceed to show that if the m-rings are well-formed then even for a small value of k they are ϵ-nice; we model Internet latencies by growth-constrained metrics. Furthermore, we achieve a similar conclusion for a much more general family of doubling metrics.

Growth-constrained metrics. We show that even with small ring cardinalities it is possible to make the rings ϵ-nice. We consider a model where the metric on the Meridian nodes is growth-constrained, but we make no such assumption about the non-Meridian nodes. This is important because even in a quite unfriendly metric we might be able to choose a relatively well-behaved subset of (Meridian) nodes. We will also assume that the rings are well-formed. Intuitively, this is desirable since in a growth-constrained metric the density is approximately uniform.

Theorem 6.3. *Let the metric on S_M have grid dimension α. Fix $\delta \in (0,1)$ and $\epsilon \leq 1$; let the cardinality of a Meridian ring be $k = O(\frac{1}{\epsilon})^{\alpha} \log(N/\delta)$. Suppose the Meridian rings are created by a random process and are well-formed (but not necessarily independent). Then with probability at least $1-\delta$ they are ϵ-nice.*

Proof. Fix two Meridian nodes u, v. Recall that we are looking for a Meridian neighbor w of node u such that $d_{vw} \leq d_{uv}$. Let $r = \epsilon d_{uv}$ and pick the smallest i such that $d_{uv} + r \leq 2^i$. Then

$$B_{ui} \subset B_v(2^i + d_{uv}) \subset B_v(2^{i+1} - r) = B_v(\gamma r), \tag{6.1}$$

where $\gamma = 4 + 3/\epsilon$. By definition of the grid dimension

$$|B_{ui}| \leq |B_v(\gamma r)| \leq \gamma^{\alpha} |B_v(r)|. \tag{6.2}$$

Since $B_u(r)$ lies in $R_{ui} \cup R_{(u,i-1)}$, and the corresponding m-rings S_{ui} and $S_{(u,i-1)}$ are well-formed, at least one node from these two m-rings lands in $B_v(r)$ with some (small) failure probability p. We claim that p is *very* small, namely $p < \delta/N^2$. Indeed, note that p is upper-bounded by the probability of not hitting $B_v(r)$ if we select k nodes uniformly at random from a larger set B_{ui}. By (6.2) and the Chernoff Bounds the latter probability is at most δ/N^2, claim proved.

Recall that p is a failure probability for a given ordered node pair. By Union Bound, the probability that *any* node pair fails is at most $p \cdot N^2 < \delta$, as required. □

Doubling metrics. For doubling metrics the notion of well-formed rings is no longer adequate, since we might need to boost the probability of selecting a node from a sparser region. In fact, this is precisely the goal of our ring-membership management in Section 6.1. Fortunately, mathematical literature provides a natural way to formalize this intuition.

Recall that a measure is *s-doubling* [Hei01] if for any ball B, the measure of B is at most s times larger than that of a ball with the same center and half the radius. Intuitively, a doubling measure is an assignment of weights to nodes that makes a metric look growth-constrained; for instance, for an N-node exponential line the node with coordinate 2^i will have weight 2^{i-N}. It is known [Hei01] that for any metric of doubling dimension α there exists a $2^{O(\alpha)}$-doubling measure μ.

With a doubling measure in mind, we extend Definition 6.2 (of well-formed m-rings) as follows:

Definition 6.4. Consider a measure μ on nodes that assigns a finite non-zero probability to every node. Say than an m-ring S_{ui} is *μ-well-formed* if its distribution is that of a random k-node subset of R_{ui} drawn according to the measure $\mu(\cdot)/\mu(R_{ui})$, or if $|R_{ui}| \leq k$.

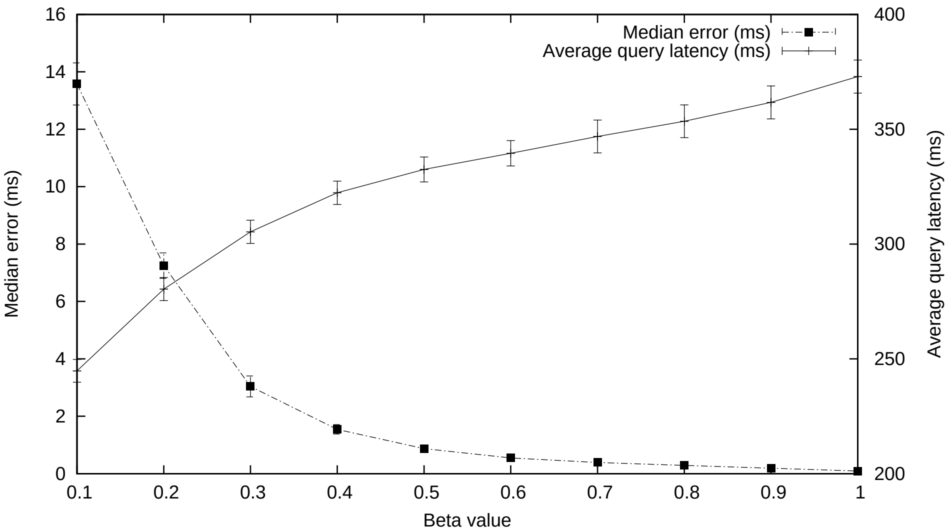

Figure 6.5: An increase in $\beta := 1/\beta_0$ significantly improves accuracy for $\beta \leq 0.5$. The average query latency increases with increasing β, as a bigger β increases the average number of hops taken in a query.

Now we obtain the guarantee in Theorem 6.3 (via a similar proof technique), where instead of well-formed m-rings we use μ-well-formed m-rings, and instead of the grid dimension we plug in a potentially much smaller doubling dimension of S_M.

Theorem 6.5. *Suppose the metric on S_M has doubling dimension α, and let μ be a 2^α-doubling measure on S_M. Fix $\delta \in (0,1)$ and $\epsilon \leq 1$; let the cardinality of a Meridian ring be $k = (\frac{1}{\epsilon})^{O(\alpha)} \log(N/\delta)$. Suppose the Meridian rings are created by a random process and are μ-well-formed (but not necessarily independent). Then with probability at least $1 - \delta$ they are ϵ-nice.*

Proof. Fix two Meridian nodes uv and let $r = \epsilon d_{uv}$. Pick the smallest i such that $d_{uv} + r \leq 2^i$. By (6.1), applying the definition of a doubling measure $\log \gamma$ times gives

$$\mu[B_{ui}]/\mu[B_v(r)] \leq 2^{O(\alpha \log \gamma)} = \gamma^{O(\alpha)}. \tag{6.3}$$

In the proof of Theorem 6.3, we essentially consider the special case when μ is the uniform measure, and use (6.2) to show that at least one node from S_{ui} or $S_{(u,i-1)}$ lands in $B_v(r)$, with failure probability at most δ/N^2 (and then the theorem follows by the Union Bound). Using (6.3) instead of (6.2), this proof trivially generalizes to any μ. □

6.2.3 Nearest neighbors and central leaders

We prove that the Meridian algorithm for nearest neighbor selection and (more generally) for central leader election achieves good approximation ratios, under the assumption that the Meridian rings are ϵ-nice. Specifically, algorithm $\mathcal{A}(2)$ is 3-approximate, for any $\epsilon \leq \frac{1}{8}$. A better approximation ratio can be proved for algorithm $\mathcal{A}^*(\beta_0)$; the provable accuracy of this algorithm tends to improve as β_0 and ϵ get smaller. The tradeoff between β_0 and the approximation ratio matches our simulation results (see Figure 6.5). We summarize these results as follows:

Theorem 6.6. *Suppose the Meridian rings are ϵ-nice, for some $\epsilon \leq \frac{1}{4}$. Consider Meridian algorithms $\mathcal{A}(2)$ and $\mathcal{A}^*(\cdot)$ for nearest-neighbor search and, more generally, for central leader election. Then:*

(a) *algorithm $\mathcal{A}(2)$ is 3-approximate, for any $\epsilon \leq \frac{1}{8}$; completes in $\lceil \log \Delta \rceil$ steps.*

(b) *algorithm $\mathcal{A}^*(1+\epsilon^2)$ is $(1+3\epsilon)$-approximate; completes in $\lceil \log(\Delta/\epsilon^2) \rceil$ steps.*

(c) *algorithm $\mathcal{A}^*(1+\gamma)$ is $(1+3\epsilon+\gamma)$-approximate, for any $\gamma \in \left[\epsilon^2;\ \frac{2}{5}\right]$; completes in $\lceil \log(\Delta/\gamma) \rceil$ steps.*

Proof Sketch. Let T be the set of targets, and let $d_T(u)$ be the average distance from node u to the targets in T. Let v^* be the central leader, i.e. the Meridian node that minimizes d_T. For a node u, let $r(u) = d_T(u)/d_T(v^*)$ be the approximation ratio. If the query is forwarded from node u to node v, we say that the *progress* at u is $d_T(u)/d_T(v)$.

For part (a) we show that the progress is at least 2 at every node u such that $r(u) \geq 3$, so in at most $\log \Delta$ steps we reach some node v such that $r(v) < 3$.

For parts (bc) we define a function $f(x)$ which is continuously increasing from $f(1) < 1+3\epsilon$ to infinity, and show that algorithm $\mathcal{A}(\beta_0)$ achieves progress $x \geq \beta_0$ at any node u such that $r(u) = f(x)$. The query is thus forwarded from node u to some node v such that $d_T(v) \leq d_T(u)/x$; it follows that $r(v) \leq f(x)/x$.

The query proceeds in two stages. In the first stage the progress at each node is $x \geq 2$; in at most $\log \Delta$ steps we reach some node u such that $r(u) < f(2)$. For the second stage, the progress can be less that 2. The crucial observation is that $f(1+y)/(1+y) \leq f(1+y/2)$ for any $y \leq 1$. Therefore if for the current node $r(\cdot)$ is $f(1+y)$, then for the next node it is at most $f(1+y/2)$.

If $\beta_0 = 1+\gamma$ then iterating this $\log \frac{1}{\gamma}$ times we reach a node such that $r(\cdot) \leq f(1+\gamma/2)$. For part (c) we note that $f(1+\gamma/2) < 1+3\epsilon+\gamma$. For part (b) we take $\gamma = \epsilon^2$ and note that $f(1+\epsilon^2/2) \leq 1+3\epsilon$. □

6.2.4 Extensions: exact nearest neighbors

We extend our result on growth-constrained metrics (Theorem 6.3 in conjunction with Theorem 6.6) to show that a version of algorithm $\mathcal{A}(2)$ finds *exact* nearest neighbors.

We will use a somewhat more restrictive model: in addition to assuming that the metric on the set S_M of Meridian nodes is growth-constrained, we will need a similar assumption about the set $Q \subset V$ of potential targets. Specifically, we consider two settings. In one setting, we assume that the metric on Q is growth-constrained, and that the set S_M of Meridian nodes is chosen uniformly at random from Q. In the other setting we make a more fine-grained assumption: we assume that the metric on $S_M \cup \{q\}$ is growth-constrained, for any target $q \in Q$. Note that here we do not assume that the metric on *all* of Q is growth-constrained; in particular, very dense clusters of potential targets are allowed.

We will show that for any query to a target in Q algorithm $\mathcal{A}(2)$ finds an exact nearest neighbor, and does so in at most $\log(\Delta)$ steps; if this is the case, we say that algorithm $\mathcal{A}(2)$ is *Q-exact*.

Theorem 6.7. *Consider a set $Q \subset V$ of potential targets. Assume either of:*

(a) *the metric on Q has grid dimension α, and S_M is a random N-node subset of Q, or*

(b) *the metric on $S_M \cup \{q\}$ has grid dimension α, for any node $q \in Q$.*

Let $k = 2^{O(\alpha)} \log\left(\frac{1}{\delta} N|Q| \log \Delta\right)$ be the cardinality of each Meridian ring, for a given parameter $\delta > 0$. Suppose the Meridian rings are created by a random process and are well-formed (but not necessarily independent). Then with probability at least $1-\delta$ the nearest-neighbor selection algorithm $\mathcal{A}(2)$ is Q-exact.

Proof Sketch. Using the technique from Theorem 6.6(a), we prove that the distance to target decreases by a factor of at least 2 on each step except maybe the last one. We have to be careful about this last step, since in general the target is not a Meridian node and therefore not a member of any ring. In particular, this is why bounded grid dimension on just S_M does not suffice.

Part (b) is easier; some extra computation is needed in part (a) due to the fact that here instead of a hard bound on the grid dimension of S_M we need to use the assumption that S_M is a random subset of Q. □

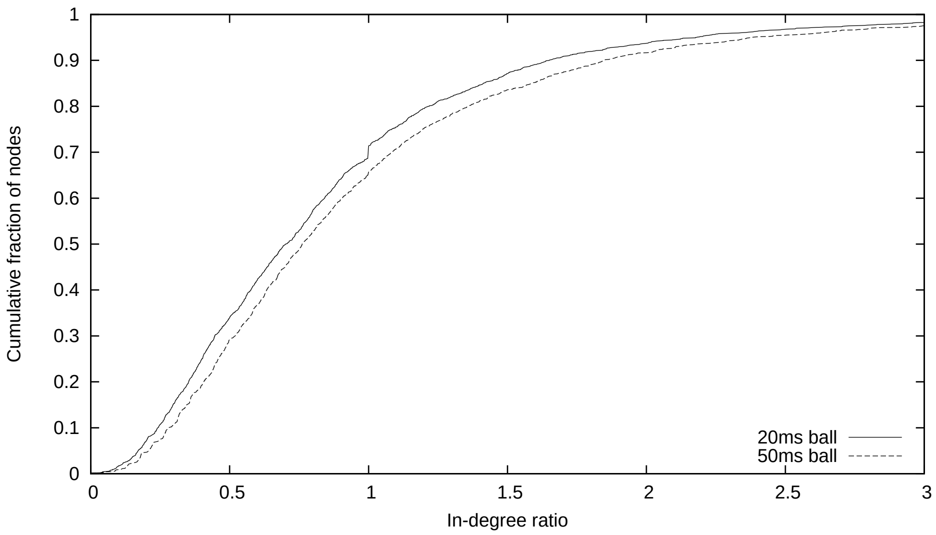

Figure 6.6: The in-degree ratio shows the average imbalance in incoming links within spherical regions. More than 90% of regions have a ratio less than 2.

6.2.5 Extensions: load-balancing

Ideally, the algorithm for nearest neighbor selection would balance the load among participating nodes. Intuitively, if $N_{\text{qy}}(\mathcal{A})$ is the maximal number of packets exchanged by a given algorithm $\mathcal{A}$ on a single query, then for m random queries we do not want any node to send or receive much more than $\frac{m}{N} N_{\text{qy}}(\mathcal{A})$ packets.

We make it precise as follows. Fix some set $Q \subset V$ and suppose each Meridian node u receives a query for a random target $t_u \in Q$. Say algorithm $\mathcal{A}$ is (γ, Q)-*balanced* if in this scenario under this algorithm any given node sends and receives at most $\gamma N_{\text{qy}}(\mathcal{A})$ packets.

We will use the setting of Theorem 6.7(a), with a further assumption that the m-rings are (stochastically) independent from each other:

Definition 6.8. Say that the Meridian rings are *independent* if the collection of all m-rings is a collection of independent random variables.

In simulations in [WSS05], this property has been verified indirectly by measuring the *in-degree ratio* of the nodes in the system. The in-degree ratio is defined as the number of incoming links to a node u over the average number of incoming links to nodes within a ball of radius r around u. If the ring sets are independent, then the in-degree ratio should be close to one; a ratio of one indicates that links to the region bounded by radius r around A are distributed uniformly across the nodes in the area. Figure 6.6 shows that Meridian distributes load evenly.

Theorem 6.9. *Consider a set $Q \subset V$ of nodes and assume that the metric on Q has grid dimension α. Let the set S_M of Meridian nodes be a random N-node subset of Q. For a parameter $\delta > 0$, let the cardinality of a Meridian ring be equal to*

$$k = 2^{O(\alpha)} \log(|Q|/\delta) \log(N) \log(\Delta).$$

Let $\gamma = 2^{O(\alpha)} \log(N\Delta/\delta)$. Suppose the Meridian rings are created by a random process and are well-formed and independent. Then with probability at least $1-\delta$ the nearest neighbor selection algorithm $\mathcal{A}(2)$ is (γ, Q)-balanced. Recall that it is Q-exact by Theorem 6.7(a).

Proof Sketch. This result is much harder to prove than all other results in this paper, essentially because we need to bound, over all nodes, not only the expected load (which is relatively easy), but also the actual load. We consider the probability space where the randomness comes from choosing Meridian nodes, Meridian neighbors, and the query targets t_u, $u \in S_M$. In this space, we consider the N nearest-neighbor queries propagating through the Meridian network. Ideally, we'd like to express the contribution of a given query i to the load on a given node u as a random variable L_{ui}, and use Chernoff Bounds to show that with high probability the sum $\sum_i L_{ui}$ does not deviate too much from its expectation. However, Chernoff Bounds only apply to *independent* random variables, which the L_{ui}'s are not. To remedy this, we need to be a lot more careful in splitting the load on u into a sum of random variables; see Section 6.5 for the full proof. □

6.2.6 Fine-tuned versions of the results

Our provable guarantees can be fine-tuned in two directions: to use relaxed versions of the grid dimension, and to rely on average (vs worst-case) guarantees.

First, our results hold under a less restrictive definition of the grid-dimension that only applies to balls that contain sufficiently many nodes: at least $\log(n)$ nodes in Theorem 6.6, and at least $\log(n|Q|)$ nodes in Theorem 6.7 and Theorem 6.9.

Second, the vicinity of a given node u could be significantly more 'well-behaved' than guaranteed by the (global) concept of grid dimension. We can show that in this case some of this node's m-rings can be made smaller. We would like the size of each m-ring S_{ui} to depend only on what happens in the corresponding ball B_{ui}. Specifically, let $r = \epsilon\, 2^{i-3}$ and choose a Meridian node v within distance $2^i - r$ from u such that the ball $B_v(r)$ has the smallest cardinality. Note that $B_v(r) \subset B_{ui}$. Define

$$\sigma_{ui} = |B_{ui}|/|B_v(r)|.$$

Now we can use this ratio, instead of the doubling dimension, to express the 'goodness' of ball B_{ui}. In particular, Theorem 6.3 it suffices to assume that the cardinality of each ring S_{ui} is at least $2.2\, \sigma_{ui} \ln(n^2/\delta)$.

Third, our guarantees are worst-case; *on average* it suffices to query only a fraction of neighbors of a given ring. To take advantage of this observation, we need a minor modification to the search algorithm. Recall that on every step in algorithm $\mathcal{A}(\beta_0)$ we look at a subset S of neighbors and forward the query to the node $w \in S$ that is closest to the target t subject to the constraint that the *progress* of w, defined as the ratio d_{ut}/d_{wt}, is at least β_0. For $\beta_0 \leq 2$, suppose instead we forward the query to an arbitrary progress-2 node in S if such node exists. It is easy to check that all our results for $\mathcal{A}(\beta_0)$ carry over to this modified algorithm.

Now in Theorem 6.6(a) (used in conjunction with Theorem 6.3) instead of asking all neighbors of a given ring at once, we can ask them in random batches of size $k_0 = O(1)^\alpha$; then in expectation one such batch will suffice to find a progress-2 neighbor. Therefore on average on every step (except the last one) we will use only k_0 randomly selected neighbors from a given ring. Similarly, we can take $k_0 = O(\frac{1}{\epsilon})^\alpha$ for Theorem 6.6(bc) (used in conjunction with Theorem 6.3), and $k_0 = O(1)^\alpha$ for Theorem 6.7. We obtain similar improvements for Theorem 6.6 used in conjuction with Theorem 6.5 for doubling metrics.

6.3 Full proof of Theorem 6.6 on central leader election

Let us recap the definitions from the proof sketch. Let use fix the set of targets T, and let $d_T(u)$ be the average distance from node u to the targets in T. Let v^* be the central leader, i.e. the Meridian node that minimizes d_T. For a node u, let $r(u) = d_T(u)/d_T(v^*)$ be the approximation ratio.

Recall that if the query is forwarded from node u to node v, we say that the *progress* at u is $d_T(u)/d_T(v)$. More generally, if node v is a Meridian neighbor of node u, say that v is a *progress-β* neighbor, for $\beta =$

$d_T(u)/d_T(v)$. We will use a function

$$f_\epsilon(\beta) = \beta(1+\epsilon)/(1-\beta\epsilon).$$

Note that for $\beta \in (1,\, 1/\epsilon)$ this function is continuously increasing to infinity.

The following claim captures the performance of a single step of the central leader election algorithm.

Claim 6.10. *Assume the rings are ϵ-nice, $\epsilon \leq 1/3$. Let u be any Meridian node, and suppose $r_T(u) = f_\epsilon(\beta)$ for some $\beta \in (1, \frac{1}{\epsilon})$. Then a progress-$\beta$ neighbor of u exists and is found by the algorithm $\mathcal{A}^*(\beta)$. Moreover, if $\beta = 2$ then such neighbor is found by algorithm $\mathcal{A}(2)$ as well.*

Proof. First we claim that such neighbor exists. Indeed, pick the smallest i such that $d(u, v^*)(1+\epsilon) \leq 2^i$. Since the rings are ϵ-nice, node u has a Meridian neighbor w within distance $\epsilon\, d(u, v^*)$ from node v^*. Then

$$\begin{aligned} d_T(w) &\leq d_T(v^*) + d(w, v^*) \leq d_T(v^*) + \epsilon\, d(u, v^*) \\ &\leq d_T(v^*) + \epsilon\ (d_T(u) + d_T(v^*)) \\ &\leq \epsilon\, d_T(u) + (1+\epsilon)\, d_T(u)/f_\epsilon(\beta) \\ &= d_T(u)/\beta, \end{aligned}$$

claim proved.

It is easy to see that w lies in $S_{ui} \cup S_{(u,i-1)}$. To prove that node w is found by $\mathcal{A}^*(\beta)$ it suffices to show that both m-rings are considered by this algorithm, i.e. that $i \leq 1 + \lceil \log d_T(u) \rceil$. Indeed,

$$\begin{aligned} d(u, v^*) &\leq d_T(u) + d_T(v^*) \leq d_T(u)\ \left(1 + f_\epsilon(\beta)^{-1}\right) \\ &\leq 2\, d_T(u)/(1+\epsilon), \\ 2^i &< 2\, d(u, v^*)\, (1+\epsilon) \leq 4\, d_T(u). \end{aligned}$$

Finally, for the case $\beta = 2$ we need to show that node w is found by algorithm $\mathcal{A}(2)$ as well. Specifically, we need to prove two things:

(i) if $w \in S_{(u,i-1)}$ then $i - 1 \geq j - 1$.

(ii) if $w \in S_{ui}$ then $i \geq j - 1$.

First let us note that by the triangle inequality we have

$$d(u, w) \geq d_T(u) - d_T(w) \geq d_T(u)/2. \tag{6.4}$$

Now if $w \in S_{(u,i-1)}$ then $d_T(u) \leq 2\, d(u, w) \leq 2^i$ by (6.4); it follows that $i \geq j$, proving (i). For (ii) recall that

$$d(u, w) \leq d(u, v) + d(v, w) \leq (1+\epsilon)\, d(u, v) \leq 2^i. \tag{6.5}$$

By (6.4), (6.5) and the definition of j it follows that

$$2^j \leq 2\, d_T(u) \leq 4\, d(u, w) \leq 2^{i+2},$$

so $j \leq i + 1$, proving (ii). □

Let us state some properties of the function $f_\epsilon(\beta)$ that will be used in the forth-coming proof of Theorem 6.6. Out of these five properties, most crucial is property (c): in conjunction with Claim 6.10 it shows that in one step our search algorithm passes from an $f_\epsilon(1+\gamma)$-approximate neighbor to an $f_\epsilon(1+\gamma/2)$-approximate neighbor.

Claim 6.11. *Some useful properties of the function $f_\epsilon(\beta)$:*

(a) function $f_\epsilon(2)$ is at most 8 whenever $\epsilon \leq \frac{1}{3}$, and at most 3 whenever $\epsilon \leq \frac{1}{8}$.

(b) $f_\epsilon(1+\gamma)/(1+\gamma) \leq f_\epsilon(1+\gamma/2)$, for any $\epsilon \leq \frac{1}{3}$ and any $\gamma \in (0,1)$.

(c) $f_\epsilon(1+\epsilon^2/2) \leq 1+3\epsilon$ for any $\epsilon \leq \frac{1}{4}$.

(d) $f_\epsilon(1+\gamma/2) \leq 1+3\epsilon+\gamma$, for any $\epsilon \leq \frac{1}{4}$ and any $\gamma \in (0,\frac{2}{5})$.

Proof Sketch. Part (a) are trivial: just plug in the definition of $f_\epsilon(2)$. For parts (bcd), we plug in the definition of $f_\epsilon(\cdot)$ and carefully solve the resulting inequality for ϵ.

In part (b) the inequality reduces to $\epsilon \leq 1/(2+\gamma)$, which holds for any $\epsilon \leq \frac{1}{3}$.

In part (c) we get $g(\epsilon) := \epsilon((1+3\epsilon)^2+20) \leq 6$, which is true for any $\epsilon \leq \frac{1}{4}$ since the function $g(\epsilon)$ is increasing in ϵ and $g(\frac{1}{4}) < 6$.

Finally, in part (d) the inequality reduces to

$$g(\epsilon) := \epsilon^2(3\gamma+6) + \epsilon(\gamma^2+4\gamma-2) - \gamma \leq 0.$$

Since $g(0) = -\gamma < 0$ and the polynomial $g(\epsilon)$ is a quadratic in ϵ, it has two roots, call them ϵ_1 and ϵ_2, and it is negative for any $\epsilon \in (\epsilon_1;\epsilon_2)$. Therefore it suffices to show that $g(\frac{1}{4}) < 0$. Indeed, solving the latter inequality for γ we get $\gamma < (\sqrt{41}-3)/8$, which is more than $\frac{2}{5}$. □

Now we are ready to prove Theorem 6.6.

Proof of Theorem 6.6(a): We need to prove that algorithm $\mathcal{A}(2)$ finds a 3-approximate neighbor of q. By Claim 6.10 while the query visits nodes u such that $r_T(u) \geq f_\epsilon(2)$, the algorithm finds a progress-2 neighbor of u and forwards the query to it. The distance $d_T(u)$ goes down by a factor of at least 2 at each step, so after at most $\log(\Delta)$ steps the query should arrive at some node v such that $r(v)$ is less than $f_\epsilon(2)$, which is at most 3 by Claim 6.11(a). □

Proof of Theorem 6.6(b): We will show that $\mathcal{A}^*(\beta)$ finds a $(1+3\epsilon)$-approximate neighbor of q. The query proceeds in two stages. In the first stage, while the query visits nodes u such that $r_T(u) \geq f_\epsilon(2)$, by Claim 6.10 the distance $d_T(u)$ goes down by a factor of at least 2 at each step. So after at most $\log(\Delta)$ steps the query should arrive at some node v such that $r(v)$ is less than $f_\epsilon(2)$, which is at most 8 by Claim 6.11(a).

In the second phase the progress at each step is smaller than 2. Specifically, by Claim 6.10 and Claim 6.11b our search algorithm passes from an $f_\epsilon(1+\gamma)$-approximate central leader to an $f_\epsilon(1+\gamma/2)$-approximate central leader, for any $\gamma \in (0,1)$. By induction on i we show that after i more steps the query will arrive at node w such that $r(w) < f_\epsilon(1+2^{-i})$. So $i = \lceil \log(2/\epsilon^2) \rceil$ steps suffices by Claim 6.11c. □

Proof of Theorem 6.6(c): The proof is similar to that for part (b); in the second stage $i = \lceil \log 2/\gamma \rceil$ steps suffices by Claim 6.11d. □

6.4 Full proof of Theorem 6.7 on exact nearest neighbors

Let us start with two easy applications of Chernoff Bounds. Their proofs are fairly standard; we include them here for the sake of completeness.

Claim 6.12. *Consider sets $T \subset V$. Suppose we choose a k-node subset $S \subset V$ uniformly at random from V. Then with failure probability at most $e^{-(1-1/\mu)^2\mu/2}$ some node from S lands in T, where $\mu = k|T|/|V|$.*

Proof. Denote the desired event by A. The distribution of S_{ui} is that of the following process P: pick nodes from V independently and uniformly at random, until we gather k distinct nodes. For simplicity consider a slightly modified process P': pick k nodes from B_{ui} independently and uniformly at random, possibly with repetitions. Obviously, P' is doing exactly the same as P, except P might stop later and, accordingly, choose some more nodes. Therefore $\Pr_P[A] \geq \Pr_{P'}[A]$.

Let's analyze process P'. Let X_j be a 0-1 random variable that is equal to 1 if and only if the j-th chosen node lands in $B_v(r)$. Then $\Pr[X_j = 1] = |T|/|V|$, so $\mu = E(\sum X_j)$. The claim follows from Lemma 2.1(a) with $y = 1$ and $1 - \epsilon = 1/\mu$. □

Claim 6.13. *Consider two sets $S' \subset S$ and suppose n nodes are chosen independently and uniformly at random from S; say X of them land in S'. Let $\lambda = n|S'|/|S|$. Then:*

(a) $\Pr[X < \lambda/2] \leq e^{-\lambda/8}$*,*

(b) $\Pr[X > k] \leq e^{-k/16}$ *for any* $k \geq 2\lambda$*,*

(c) $\Pr[X > 2\lambda] \leq (e/4)^{\lambda}$*.*

Proof. Let X_j be a 0-1 random variable that equals 1 if and only if the j-th chosen node lands in S'. Then $X = \sum_{j=1}^{n} X_j$ is a sum of independent random variables, so

$$E(X) = n \cdot \Pr[X_j = 1] = n \cdot |S'|/|S| = \lambda.$$

For part (a), use Lemma 2.1(a) with $y = 1$ and $\epsilon = 1/2$. Parts (bc) follow from Lemma 2.1(b) with $y = 1$ and $\beta = 2$; specifically, take $\mu = k/2$ in part (b), and take $\mu = \lambda$ in part (c). □

We prove part (b) of the Theorem first since it is simpler.

Proof of Theorem 6.7(b): Let the size of a Meridian ring be $k = 2.2{\cdot}10^{\alpha}\ln(1/p)$, where $p = \delta/N|Q|\log(\Delta)$. Let $q \in Q$ be the target, and let $v \in S_M$ be its exact nearest neighbor. Fix some Meridian node u, let $d = d_{uq}$ and choose the smallest i such that $1.5d \leq 2^i$.

We claim that either $v \in S_{ui}$, or with failure probability at most p node u has a Meridian neighbor $w \in B_q(d/2)$. Indeed,

$$\begin{aligned} B_{ui} &\subset B_q(2^i + d) \subset B_q(4d) \\ |B_{ui}| &\subset |B_q(4d)| \leq 8^{\alpha}\,|B_q(d/2)|, \end{aligned}$$

so if $|B_{ui}| \geq k$ then the claim follows from Claim 6.12; the constant 2.2 in front of k works numerically as long as e.g. $n|Q| > 55^2$ and $\delta < e^{-2}$, which is quite reasonable. Finally, if $|B_{ui}| \leq k$ then every node in B_{ui} is in ring S_{ui}, including v, claim proved.

Recall that, letting $j = \lceil \log d \rceil$, algorithm $\mathcal{A}(2)$ at node u considers the m-rings $S_{(u,j-1)}$, S_{uj} and $S_{(u,j+1)}$. Since by the triangle inequality $d/2 \leq d_{uw} \leq 3d/2$, node w lies in one of these three m-rings, and therefore is found by $\mathcal{A}(2)$. So the progress is at least 2 at every step except maybe the last one, with failure probability at most p. Therefore the algorithm makes at most $\log \Delta$ steps before completion.

Finally, for a single (u, q) pair the failure probability for a single step is at most p. Taking the Union Bound over all $N|Q|$ possible (u, q) pairs and all $\lceil \log \Delta \rceil$ possible steps, it follows that the total probability is at most δ. □

Theorem 6.7(a) is proved using the same idea, except we need to address the fact that Meridian nodes themselves are chosen at random from Q.

Proof of Theorem 6.7(a): Let $Q_u(r)$ denote the closed ball in Q of radius r around node u, i.e. the set of all nodes in Q within distance r from u. Denote $Q_{ui} = Q_u(2^i)$ and let the cardinality of a Meridian ring be

$$k = 8 \cdot 8^{\alpha} \ln\left(\tfrac{2}{\delta}\, N|Q| \log \Delta\right). \tag{6.6}$$

Let q be the target and let $v \in S_M$ be its exact nearest neighbor. Fix some Meridian node u, let $d = d_{uq}$ and $B = B_q(d/2)$; choose the smallest i such that $1.5\, d \leq 2^i$.

Note that without loss of generality we can view the process of selecting S_M from Q as follows: choose the cardinality x for B_{ui} from the appropriate distribution, then choose, independently and uniformly at random, x nodes from Q_{ui}, and $n - x$ nodes from $Q \setminus Q_{ui}$.

We claim that with failure probability at most $\delta' = \delta/N|Q| \log(\Delta)$ either $v \in S_{ui}$, or node u has a Meridian neighbor $w \in B$. Indeed, if the cardinality of B_{ui} is at most k, then all of B_{ui} lies in the ring S_{ui}, including v. Now assume the cardinality of B_{ui} is some fixed number $x > k$. Since by the triangle inequality $Q_{ui} \subset Q_q(2^i + d) \subset Q_q(4d)$, it follows that

$$\frac{x}{E(|B|)} = \frac{|Q_{ui}|}{|Q_u(d/2)|} \leq \frac{|Q_u(4d)|}{|Q_u(d/2)|} \leq 8^{\alpha},$$

where the last inequality holds by definition of the grid dimension. Therefore by Claim 6.13(a) with failure probability at most $\delta'/2$ the cardinality of B is at least half the expectation. If it is indeed the case that, then by Claim 6.12 with failure probability at most $\delta'/2$ some node in ring S_{ui} lands in B. So the total failure probability is at most δ', claim proved.

As in the proof of part (b), we show that node w is found by algorithm $\mathcal{A}(2)$. Therefore the progress is at least 2 at every step except maybe the last one, with failure probability at most δ'. Finally, we take the Union Bound over all $N|Q|$ possible (u, q) pairs and all $\log \Delta$ possible steps to show that the probability that any such pair fails on any step is at most δ. □

6.5 Full proof of Theorem 6.9 on load-balancing

In this section we will prove Theorem 6.9 on load-balancing. A large part of the proof is the setup (Sections 6.5.1 and 6.5.2): it is non-trivial to restate the algorithm and define the random variables so that the forth-coming Chernoff Bounds-based argument works through. For technical reasons we introduce some minor changes in the definition of the m-rings and in the search algorithm; these changes do not (really) affect the practical implementation of Meridian. Proving our result for the exact version of Meridian that is implemented leads to mathematical difficulties that are far beyond the scope of this paper.

Recall that for the present theorem we use the setting of Theorem 6.7(a). Compared to the latter, we increase the ring cardinalities by a factor of $O(\log N)(\log \Delta)$. This is essentially because we cannot use Chernoff Bounds on collections of random variables that are *almost* independent – we need exact independence, which is hard to come by. We conjecture that this blow-up can be avoided by a more careful analysis of almost-independent random variables. However, such analysis is again beyond the scope of this paper.

6.5.1 Setup: Meridian rings and the search algorithm

For convenience, for any $x > 0$ let us define a set of integers $[x] = \{0, 1 \ldots \lceil x \rceil\}$.

Recall that each m-ring S_{ui} was defined as a subset of the corresponding ring R_{ui}, as long as $|B_{ui}| \geq k$. Here to simplify the proofs let us allow each S_{ui} to be an arbitrary subset of B_{ui}:

Definition 6.14. The distribution of each Meridian ring S_{ui} is the distribution of a set of k nodes that are drawn independently and uniformly at random from the corresponding ball B_{ui}, possibly with repetitions.

Note that all previous results for growth-constrained metrics work under this definition as well.

Recall that on every step in algorithm $\mathcal{A}(\cdot)$ we look at a subset S of neighbors, and either the search stops, or the query is forwarded the node $w \in S$ that is closest to the target. We will relax this as follows: if w is a progress-2 node, then instead of forwarding to w the algorithm can forward the query to an arbitrary progress-2 node in S. It is easy to check that all our results for $\mathcal{A}(\cdot)$ carry over to this modification.

We will now proceed to define a specific version of $\mathcal{A}(2)$ which can be seen as a rule to select between different progress-2 nodes; we denote it $\mathcal{A}$.

Recall that each ring S_{ui} consists of k nodes from B_{ui}. More formally, let us say that S_{ui} consists of k *slots*, each of which is a node id selected independently and uniformly at random from B_{ui}. Let us partition these slots into $L\log(\Delta)$ collections of size k' each, where

$$\begin{aligned} L &= 6\ln\left(\tfrac{1}{\delta}N\log\Delta\right), \\ k' &= 8\cdot 10^{\alpha}\ln(2K/\delta), \\ K &= N|Q|L\log(\Delta). \end{aligned}$$

We will denote these collections by $\mathcal{C}_{ui}(j,l)$, where $j \in [\log\Delta]$ and $l \in [L]$. Each collection will just consist of k' consecutively numbered slots, starting from the slot number $(jL+l)k'$. Let $S_{ui}(j,l)$ be the set of nodes whose ids are stored in the slots in collection $\mathcal{C}_{ui}(j,l)$. Obviously, $S_{ui}(j,l) \subset B_{ui}$, and the union of all sets $S_{ui}(\cdot,\cdot)$ is S_{ui}.

Say a j*-step query* is a query on the j-th step of the algorithm. When node u receives a j-step query to target q, it chooses $l \in [L]$ in a round-robin fashion (the round-robin is separate for each uj pair) and (essentially) lets algorithm $\mathcal{A}(2)$ handle this query using only the neighbors in $S_{ui}(j,l)$, for the corresponding i. Specifically, node u sets $i = 1 + \lfloor\log d_{uq}\rfloor$ and asks every node in $S_{ui}(j,l)$ to measure the distance to q. Out of these nodes, let w be one that is closest to q. If w is a progress-2 node, then the query is forwarded to w; else, the search stops, and node w is reported to the node that originated the query.

Using the argument from part (a) we can show that for a given tuple (u,q,j,l) either the corresponding set $S_{ui}(j,l)$ contains a progress-2 node or it contains a nearest neighbor of q, with failure probability at most δ/K. The Union Bound over all K possible (u,q,j,l) tuples shows that our algorithm is Q-exact with failure probability at most δ.

Note that algorithm $\mathcal{A}$ can be seen as $\mathcal{A}(2)$ with a rule to select between different progress-2 nodes if such nodes exist: namely, choose a progress-2 node from the corresponding $S_{ui}(j,l)$.

6.5.2 Setup: randomization and random variables

Recall that each $S_{ui}(j,l)$ is a set of k' nodes drawn from B_{ui} independently and uniformly at random, possibly with repetitions. Moreover, once the set S_M of all Meridian nodes is fixed then (since the m-rings are independent), the collection of all sets

$$\{S_{ui}(j,l):\ u \in S_M,\ i,j \in [\log\Delta], l \in [L]\}$$

is a collection of independent random variables.

We consider the probability distribution induced by several independent random choices, namely:

- a random N-node subset S_M of Q,
- random subsets $S_{ui}(j,l) \subset B_{ui}$, independently for each tuple (u,i,j,l),
- target t_u for each node u.

For a collection of independent random choices, without loss of generality we can assume that a given choice happens any time before its result is actually used. In particular, we will assume the following order of events. First, S_M and t_u's are chosen. After that the time proceeds in $\log(\Delta)$ epochs. In a given epoch j, all subsets $S_{ui}(j,\cdot)$ are chosen, and then all queries are advanced for one step.

Recall that all queries are handled separately, even if a given node simultaneously receives multiple queries for the same target. When node u handles a j-step query and in the process measures distance to its neighbor v, we say that v receives a j*-step request* from u. Let's define several families of random variables; here j ranges between 0 and $\log\Delta$:

- $X_{uv}(j,l)$ is the number of j-step queries forwarded from u to v, and handled at u using, for some i, a set $S_{ui}(j,l)$.
- X_u^j is the number of all j-step queries forwarded to node u; set $X_u^0 = 1$.
- $Y_{uv}(j,l)$ is the number of j-step requests that are received by v from u, and handled at u using, for some i, a set $S_{ui}(j,l)$.
- Y_u^j is the number of all j-step requests received by node u.

Note that $X_{uv}(j,l) \leq X_u^j/L$ and $Y_{uv}(j,l) \leq X_u^{j-1}/L$.

6.5.3 The actual proof

First let us analyze the choice of S_M and the queries. Let T be the set of all N queries. For $q \in T$, let $t(q)$ be the corresponding target. Let $T_v(r)$ be the set of queries $q \in T$ such that $t(q)$ is within distance r from v. Let $t(S)$ be the set of all targets in the set S of queries. Let $\psi = N/|Q|$. By Claim 6.13 $|B_u(r)|$ and $|T_u(r)|$ are close to its expectation:

Claim 6.15. *With failure probability at most δ, for any $u \in S_M \cup t(T)$ and radius r the following holds:*

(*) if $z = \psi|Q_u(r)| \geq k_0$ then $|B_u(r)|$ and $|T_u(r)|$ are within a factor of 2 from z, else they are at most $2k_0$, where $k_0 = O(\log(n/\delta))$.

For every j-step query received, a given node sends some constant number c of packets to each of the k' neighbors in the corresponding set $S_{ui}(j,l)$. Therefore a given node u sends $ck'\sum_j X_u^j$ packets total, and receives $c\sum_j Y_u^j$ packets total. Since a single query involves exchanging at most $ck'\log(\Delta)$ packets, algorithm $\mathcal{A}$ is (γ, Q)-balanced if and only if

$$\textstyle\sum_j (k'X_u^j + Y_u^j) \leq 2\gamma k' \log(\Delta) \tag{6.7}$$

for every node u. Recall that γ is a parameter in the theorem statement.

Definition 6.16. Property $\mathbf{P}(\mathbf{j})$ holds if and only if for each node v it is the case that $X_v^j \leq \gamma$ and $Y_v^j/k' \leq \gamma$.

By (6.7) it suffices to prove that with high probability $\mathbf{P}(\mathbf{j})$ holds for all j; recall that j ranges between 0 and $\log\Delta$. It suffices to prove the following inductive claim:

Claim 6.17. *If property $\mathbf{P}(\mathbf{j-1})$ holds, then with failure probability $\leq \delta/\log(\Delta)$ property $\mathbf{P}(\mathbf{j})$ holds, too.*

Then we can take the Union Bound over all $\log\Delta$ steps j to achieve the desired failure probability δ.

Let's prove Claim 6.17. Suppose all queries have completed $j-1$ steps and are assigned to the respective sets $S_{ui}(j,l)$. Now the only remaining source of randomness before the j-th step is the choice of these sets.

In particular, each random variable $X_{uv}(j,l)$ depends only on one set $S_{ui}(j,l)$, and so does $Y_{uv}(j,l)$. Since these sets are chosen independently, for any fixed node v the random variables

$$\{X_{uv}(j,l):\ u \in S_M, l \in [L]\}$$

are independent, and so are the random variables

$$\{Y_{uv}(j,l):\ u \in S_M, l \in [L]\}.$$

First we claim that $P(j)$ holds in expectation:

Claim 6.18. *For every Meridian node v and every step j,* **(a)** $E(X_v^j) \leq \gamma/2$, *and* **(b)** $E(Y_v^j/k') \leq \gamma/2$.

Let us assume for now that the above claim holds.

Suppose property $P(j-1)$ holds. Let's bound the load on some fixed node v. Note that

$$X_v^j = \sum_{\text{all pairs } (u,l)} X_{uv}(j-1,l)$$

is a sum of independent random variables, each in $[0,y]$ for $y = \gamma/L$. Applying Claim 2.1(b) with $\mu = \gamma/2$, we see that

$$\Pr[X_v^j > \gamma] \leq (e/4)^{L/2} \leq \delta/2N\log(\Delta).$$

Similarly, $Y_v^j = \sum_{(u,l)} Y_{uv}(j,l)$ is a sum of independent random variables, each in $[0,y]$, so by Claim 2.1(b) we can upper-bound $\Pr[Y_v^j/k' > \gamma]$. By the Union Bound property $P(j)$ holds with the total failure probability at most δ. This completes the proof of Claim 6.17 and Theorem 6.9.

It remains to prove Claim 6.18. Let S_0 be the set of queries $q \in T$ such that v is a nearest neighbor of the target $t(q)$.

Claim 6.19. $|S_0| \leq O(2^\alpha)\log(N/\delta)$.

Proof. Choose target $t \in t(S_0)$ such that d_{vt} is maximal. Let $d = d_{vt}$. Then $B_t(d/\tau) \in \{q\}$ for any $\tau > 1$, so by Claim 6.15 $|Q_t(d/\tau)| \leq O(\log(n/\delta))$. Note that $S_0 \subset B_t(2d) \subset Q_t(2d)$ and

$$|Q_t(2d)| \leq (2\tau)^\alpha |Q_t(d/\tau)| \leq (2\tau)^\alpha O(\log(n/\delta)).$$

Claim follows if we take small enough $\tau > 1$. □

Let r_0 be the smallest r such that $B_v(r)$ has cardinality at least twice the k_0 from Claim 6.15. Let $R_i = T_v(r_0\, 2^i)$. Let $S \subset T$ be the set of queries that get forwarded to v on step j; recall that $X_v^j = |S|$.

Claim 6.20. *For any query $q \in T \setminus (S_0 \cup R_0)$, letting $t = t(q)$, we have*

$$\Pr[q \in S] \leq O(2^\alpha)/|B_v(d_{vt})|.$$

Proof. Let $d = d_{vt}$ and suppose query q is currently at node u. Since $q \notin S_0$ this query gets forwarded to some node $w \in B_q(d_{ut}/2)$, so if $d > d_{ut}/2$ then clearly $q \notin S$. Assume $d \leq d_{ut}/2$. Since $B_v(d) \subset B_t(2d)$, by Claim 6.15 we have

$$\begin{aligned}
|B_v(d)| &\leq |B_t(2d)| \leq 2\psi |Q_t(2d)| \leq 2\psi\, 2^\alpha |Q_t(d)| \leq 4\, 2^\alpha |B_t(d)|,\\
\Pr[q \in S] &= 1/|B_t(d_{ut}/2)| \leq 1/|B_t(d)|,
\end{aligned}$$

which is at most $4\, 2^\alpha/|B_v(d)|$, as required. □

Now for $R = R_{i+1} \setminus (R_i \cup S_0)$ and $r = r_0\, 2^i$

$$\begin{aligned}
\psi_i &:= E|S \cap R| \leq |R_{i+1}|\ \Pr[q \in S :\ q \in R] \\
&\leq O(2^i)|R_{i+1}|/|R_i| \leq O(4^\alpha), \\
E|X_v^j| &= E|S| \leq |S_0| + |R_0| + \sum \psi_i \\
&\leq O(2^\alpha)\log(n/\delta) + O(4^\alpha)\log(\Delta) \\
&\leq O(4^\alpha)\log(n\Delta/\delta) \leq \gamma/2.
\end{aligned}$$

This completes the proof of Claim 6.18(a). For Claim 6.18(b), let S be the set of queries that cause a j-step request to v. Suppose a j-step query q is at node u; let $t = t(q)$ and $d = d_{ut}$. Node v receives a j-step request due to t only if $d_{uv} \leq 2d$, so let's assume it is the case. Then $d_{vt} \leq d + d_{uv} \leq 3d$, so

$$\begin{aligned}
B_u(d_{vt}) &\subset B_u(d_{uv} + d_{vt}) \subset B_u(5d) \\
|B_u(d_{vt})| &\leq |B_u(5d)| \leq 4\,(2.5)^\alpha\, |B_u(2d)| \\
\Pr[v \in S] &\leq 1/|B_u(2d)| \leq 4\,(2.5)^\alpha\, |B_u(d_{vt})|
\end{aligned}$$

as long as $|B_u(d_{vt})|$ is at least twice as large as the k_0 from Claim 6.15. The rest of the proof of Claim 6.18(b) is similar to that of Claim 6.18(a). This completes the proof of Claim 6.18 and Theorem 6.9.

Chapter 7

Distance Estimation and Object Location via Rings of Neighbors

In *node labeling* problems one needs to assign short labels to nodes of a graph so that they capture some (problem-specific) global information about distances and routes in the graph. We consider four problems of this type: low-stretch routing schemes [PU89], distance labeling [GPPR04], searchable small worlds [Kle00b], and triangulation-based distance estimation [KSW04].

We approach these problems with a common technique called *rings of neighbors*, which refers to a sparse distributed data structure that underlies all our constructions. The idea is that every node u stores pointers to some nodes called 'neighbors'; these pointers are partitioned into several 'rings', so that for some increasing sequence of balls $\{B_i\}$ around u, the neighbors in the i-th ring lie inside B_i; the radii of these balls and the selection of neighbors depend on the specific application. In effect, rings of neighbors form an overlay network with a certain structure imposed by the balls $\{B_i\}$.[1]

For the problems that we consider, the input is a metric or, more generally, an undirected weighted graph that induces a shortest-paths metric. We focus on doubling metrics (see Section 2.3.2 for background). Throughout this section, we let α denote the doubling dimension.

The results in this chapter have been published in Slivkins [Sli05a].

7.1 The four problems and relevant background

Let us discuss each of the four problems in more detail.

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

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

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

(b) an algorithm that inputs a routing table and a packet header, and outputs the next hop for this packet;

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

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

The algorithms in (b) and (c) must be polynomial-time computable (with respect to the input length). By a slight abuse of notation, we can talk about a routing scheme on a particular graph $G \in \mathcal{G}$ once the underlying family $\mathcal{G}$ of graphs is clear. Such routing scheme consists of routing labels, routing tables, and the algorithms in (b) and (c). [2]

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

In a trivial stretch-1 routing scheme, each node stores full routing table of the all-pairs shortest paths algorithm. However, this routing table takes up $\Omega(n \log n)$ bits, which does not scale well with n. *Compact* low-stretch routing schemes have been introduced in Peleg and Upfal [PU89], and explored in a number of subsequent papers (see [GP03, Pel00] for a survey). In particular, for any integer $k \geq 2$ there exists a $(4k-5)$-stretch routing scheme on weighted graphs with $o(k \log^2 n)$-bit packet headers and $\tilde{O}(n^{1/k})$-bit routing tables [TZ05, TZ01]; this trade-off between the stretch and the size of routing tables is essentially optimal [PU89]. Moreover, there is no routing scheme on weighted graphs with stretch less than 3 and $o(n)$-bit routing tables [GG01].

OUR CONTRIBUTIONS: We focus on routing schemes for weighted undirected graphs that induce doubling metrics (for simplicity, let's call them *doubling graphs*). In this setting, Talwar [Tal04] has achieved compact $(1+\delta)$-stretch routing schemes, for any given $\delta > 0$; his result has been improved by Chan et al. [CGMZ05]. Using rings of neighbors, we re-derive the result in [CGMZ05] via the construction and proof of correctness that are significantly shorter and simpler than the ones in [CGMZ05]; our guarantees (Theorem 7.1) are slightly improved, too. Moreover, we can give a *really* simple derivation (Theorem 7.11) if we use our result on distance labeling and allow an extra $(\log n)$ factor in the routing table size. The quantitative results are summarized in Table 7.1. All these results extend to a related model of *routing schemes on metrics*,[3] with poly-logarithmic out-degrees; see Section 7.4.1 for more details.

We note that the above guarantees are unsatisfactory if the aspect ratio Δ (the largest distance divided by the smallest distance) is very large, e.g. $\Delta = 2^n$. We wish to alleviate the dependency on Δ; we do it by replacing the $(\log \Delta)$ factor with $(\log n)(\log \log \Delta)$. The first step in this direction is Theorem 7.11, where the improvement is for packet headers only. Furthermore, in Section 7.4 we improve both packet headers and routing tables for routing schemes on metrics, and also (Theorem 7.12) for routing schemes on weighted graphs that contain near-shortest paths with small hop-counts; the latter property is, intuitively, a natural property of a "good" network topology.

FOLLOW-UP WORK. Following the publication of the conference version of Slivkins [Sli05a], and building on our techniques, Abraham et al. [AGGM06] further alleviate the dependency on the aspect ratio Δ for routing schemes on graphs (see Table 7.1). In particular, one of their results essentially improves the packet header size in Theorem 7.11 to $\lceil \log n \rceil$. They also provide an extension where they get rid of the dependence on Δ altogether, at the cost of extra poly-log(n) factors in both routing table size and packet header size. This result elaborates on our Theorem 7.12, eliminating the requirement of near-shortest paths with small hop-counts. Abraham et al. [AGGM06] also refine our results on doubling metrics, see Section 7.4 for further details.

Related work on routing schemes. An important version of routing schemes is *name-independent routing* [ABNLP90, AP90], where the node labels are a part of the input: essentially, each node is given a

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

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

	routing table size, bits	packet header size, bits
Talwar [Tal04]	$O(\frac{1}{\alpha\delta})^{\alpha}(\log^{2+\alpha}\Delta)$	$O(\alpha\log\Delta)$
Chan et al. [CGMZ05]	$(\frac{\alpha}{\delta})^{O(\alpha)}(\log\Delta)(\log D_{\text{out}})$	$O(\alpha\log\frac{1}{\delta})(\log\Delta)$
Theorem 7.1	$(\frac{1}{\delta})^{O(\alpha)}(\log\Delta)(\log D_{\text{out}})$	same as above
Theorem 7.11	$(\frac{1}{\delta})^{O(\alpha)}(\log\Delta)(\log n)(\log\log\Delta)$	$2^{O(\alpha)}(\log n)\log(\frac{1}{\delta}\log\Delta)$
Follow-up work [AGGM06]	$(\frac{1}{\delta})^{O(\alpha)}(\log\Delta)(\log n)$	$\lceil\log n\rceil$
	$(\frac{1}{\delta})^{O(\alpha)}(\log^4 n)$	$2^{O(\alpha)}(\log^3 n)$

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

Table 7.1: $(1+\delta)$-stretch routing schemes for doubling graphs

unique $\lceil\log n\rceil$-bit identifier that cannot be changed by our construction. Currently the best known results for arbitrary weighted graphs are: stretch $O(k)$ with $\tilde{O}(n^{1/k}\log\Delta)$-bit tables [AGM04a], and stretch 3 with $\tilde{O}(\sqrt{n})$-bit tables [AGM$^+$04b]; both routing schemes use poly-log packet headers.

For weighted graphs that induce doubling metrics, the extra restriction of name-independence results in more demanding storage requirements: $(1+\delta)$-stretch routing with $o(n)$-bit routing tables is no longer possible for any $\delta < 2$ [AGGM06]. However, there *is* a routing scheme with $O(1)$-stretch and polylog storage/headers [AGGM06]. Moreover, for any $\delta > 0$ there exists a $(1+\delta)$-stretch routing scheme on low-dimensional *Euclidean* metrics [AM04], also with polylog storage and headers, which is 'almost' name-independent (node labels include Euclidean coordinates).

A number of results on name-independent routing has focused on the case of bounded grid dimension (see the intro to [HKR04] for a short survey). The best current results [AMD04, AM05] achieve $(1+\delta)$-stretch with poly-log storage/headers for routing on metrics and on graphs, respectively.

Searchable small-world networks. The small-world networks have been an active topic in many branches of social and natural sciences. The 'small-world phenomenon', also known as the 'six degrees of separation', has been discovered in a seminal work of Milgram [Mil67] and recently confirmed by Dodds et al. [DMW03]. Motivated by Watts and Strogatz [WS98], Kleinberg [Kle00b, Kle00a] has articulated another striking aspect of 'small worlds': that a greedy routing algorithm can find short paths to most targets using only local information. Kleinberg went on to suggest several mathematical models where this happens [Kle00b, Kle01]. In particular, he considered a constant-dimensional grid and proved that if every node chooses a constant number of long-range contacts from a fairly natural probability distribution, then in expectation a greedy routing algorithm finds $O(\log^2 n)$-hop paths for every query. The follow-up work (e.g. [LS04, MN04, MNW04, FGP04, MN05]) has focused on small worlds on hierarchies and grid-like graphs, with versions of the basic greedy routing from [Kle00b]. This line of work has also found applications in the design of peer-to-peer systems (e.g. [MBR03]). For more background on small-world networks, refer to a very recent survey by Kleinberg [Kle06].

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

Most of the previous work has considered the distance induced by a given (possibly directed) unweighted graph of short-range contacts; note that one could start from this notion of distance and recover the short-range contacts as all nodes within distance 1. Abstracting away the useful small-world properties of grids and hierarchies, Kleinberg [Kle01] introduced searchable small worlds on distance functions induced by certain families of node sets. Here we take a somewhat different (and perhaps more basic) approach: we consider distance functions that are metrics, and we wish to extend Kleinberg's small worlds beyond those induced by hierarchies and grid-like graphs. Namely, we extend them to doubling metrics.

We use routing algorithms such that the next hop is chosen by only looking at the current node's contacts, which is a desirable property since (intuitively) this is the minimal amount of information a routing algorithm can be allowed to use. More formally, the next hop is chosen among the current node's contacts, by looking *only* at distances to these contacts and distances from these contacts to the target. Let us call such routing algorithms *strongly local*. The *greedy* algorithm used in [Kle00b] is a strongly local routing algorithm that just chooses the contact that is closest to the target.

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

We note in passing that our results trivially extend to a setting where we are given a graph of local contacts, and we add exactly one long-range contact per node; see Section 7.5.3 for further discussion.

Related work on small-world networks. In the literature on searchable small-world networks several non-greedy routing algorithms have been suggested. In Manku at al. [MNW04] the algorithm looks at all contacts of contacts of the current node, and (greedily) forwards the message to one that is closest to the target. In Martel and Nguyen [MN04] and Fraigniaud et al. [FGP04] the algorithm looks at several nodes that are closest to the current node u, looks at their contacts, among these contacts chooses one (let us call it v) that is closest to the target, and tries to deliver the message to v by forwarding it to one of the contacts of u. Finally, in Lebhar and Schabanel [LS04] the algorithm has access to contacts of the previously visited nodes. Note that all these non-greedy algorithms are *not* strongly local.

Following the publication of the conference version of Slivkins [Sli05a], we became aware that concurrently with our work, two other papers have independently considered extending searchable small worlds to broader classes of graphs. Specifically, Duchon et al. [DHLS05] consider graphs of low grid dimension, and Fraigniaud [Fra05] work on graphs of bounded treewidth. An even more recent paper [AG06] considers weighted minor-excluding graphs. Furthermore, Fraigniaud et al. [FLL06] have recently provided a complementary impossibility result for searchable small-worlds on an infinite family of graphs of large doubling dimension.[4]

Triangulation. Let us recall the definition; we state it in a slightly less general form, which however suffices for the purposes of this chapter. A *triangulation* of order k is defined as a labeling of the nodes such that a label of a given node u consists of distances from u to each node in a set S_u of at most k other nodes. Then given the labels of two nodes u and v, one can use the triangle inequality to upper-bound the uv-distance by $D^+_{uv} = \min(d_{ub} + d_{vb})$, and lower-bound it by $D^-_{uv} = \max |d_{ub} - d_{vb}|$, where the $\max$ and

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

$\min$ are taken over all $b \in S_u \cap S_v$. An (ϵ, δ)-*triangulation* is a triangulation such that $D^+_{uv}/D^-_{uv} \leq 1+\delta$ for all but an ϵ-fraction of node pairs uv. In particular, this inequality holds whenever there exists some node $b \in S_u \cap S_v$ that lies within distance $\delta d_{uv}/3$ from u or v. Note that if it holds then either bound can be seen as a $(1+\delta)$-approximate estimate on the uv-distance, and, moreover, these bounds provide a "quality certificate" for the estimate.

The distributed algorithms for triangulation from Chapter 3 offered no guarantees for some small fraction of the node pairs. Here we consider triangulation as an off-line problem of constructing the corresponding data structure given a metric, and we seek for guarantees for all node pairs. Specifically, given a metric and a parameter $\delta > 0$ we want to construct a $(0, \delta)$-triangulation of low order.

OUR CONTRIBUTION: We construct a $(0, \delta)$-triangulation of order $(\frac{1}{\delta})^{O(\alpha)} O(\log n)$.

Distance labeling. In a distance labeling scheme (DLS), each node is assigned a short label so that the distance between any two nodes can be efficiently approximated just by looking at their labels. Formally, a k-approximate DLS for a class $\mathcal{M}$ of metrics consists of a polynomial-time computable real-valued function $f(x, y)$ and, for each metric $M \in \mathcal{M}$, an assignment of labels L_u to nodes u of M such that for each node pair uv, $f(L_u, L_v)$ is within factor of k of the true uv-distance. By a slight abuse of notation, we can talk about a DLS on a particular metric $M \in \mathcal{M}$ once the underlying family $\mathcal{M}$ of metrics is clear. Given k, we'd like to minimize the maximal bit-length of node labels.

In a trivial DLS, the label of node u would encode the distances to all other nodes, taking up $O(n \log \Delta)$ bits. Exact DLS are known for two families of unweighted graphs: for bounded-genus graphs and for graphs with constant-size separators, with $\tilde{O}(\sqrt{n})$- and $O(\log^2 n)$-bit labels, respectively [GPPR04]. For weighted graphs, approximate DLS with sublinear label length have been introduced by Peleg [Pel99], see [GP03, Pel00] for a survey. In particular, for any integer k there exists a $(2k-1)$-approximate DLS on weighted graphs with $\tilde{O}(n^{1/k} \log \Delta)$-bit labels [TZ05]; a complimentary lower bound of $\tilde{O}(n^{1/k})$ is given in [TZ05, GKK+01].

Major improvements are possible for doubling metrics. For any $\delta \in (0, \frac{1}{2})$ Gupta et al. [GKL03] provided an embedding into ℓ_∞ which trivially translates into a $(1+\delta)$-approximate DLS with $(\frac{1}{\delta})^{O(\alpha)}(\log n)(\log \Delta)$ bits per label, where α is the doubling dimension and Δ is the aspect ratio. Using a different technique, Talwar [Tal04] improved this by a factor of $(\log n)$, and gave a lower bound of $(\frac{1}{\delta})^{\Omega(\alpha)}$.

OUR CONTRIBUTIONS: We observe that since the aspect ratio Δ can be arbitrarily large with respect to n, it is desirable to alleviate the dependency on Δ. Using our result on triangulation, with the upper bound D^+ as a distance estimate, we obtain a $(1+\delta)$-approximate DLS with $(\frac{1}{\delta})^{O(\alpha)}(\log n)(\log n + \log\log \Delta)$ bits per label. We further improve it to $(\frac{1}{\delta})^{O(\alpha)}(\log n)(\log\log \Delta)$ bits per label using the ideas from our first result on routing schemes. For any $\Delta \geq n^{\log n}$ and bounded α, δ this is optimal up to constant factors.[5]

BIBLIOGRAPHICAL NODE: The above results have been published in Slivkins [Sli05a]. Our initial result on DLS for doubling metrics, with $(\frac{1}{\delta})^{O(\alpha)}(\log^2 n)(\log n + \log\log \Delta)$ bits per label, has been published in Slivkins [Sli05b]. In the period between the latter and the former publications, this initial result has been improved by a $(\log n)$ factor in [MHP05], using a different technique.

The unifying technique. In this paper we present results on four related, yet different problems. These results are unified by a common technique: rings of neighbors. Moreover, these results are intertwined, in the sense that one result elaborates ideas pioneered in another. This flow of ideas is represented in Figure 7.1. Note that both Theorem 7.11 and Theorem 7.12 build on Theorem 7.8; however, Theorem 7.11 just uses it as a black box, whereas Theorem 7.12 imports its techniques and elaborates on them. In fact, the proof of Theorem 7.12 is the culmination of our techniques for routing schemes, triangulation and distance labeling.

[5] The lower bound is from [MHP05], see (7.1) on page 97 in this thesis.

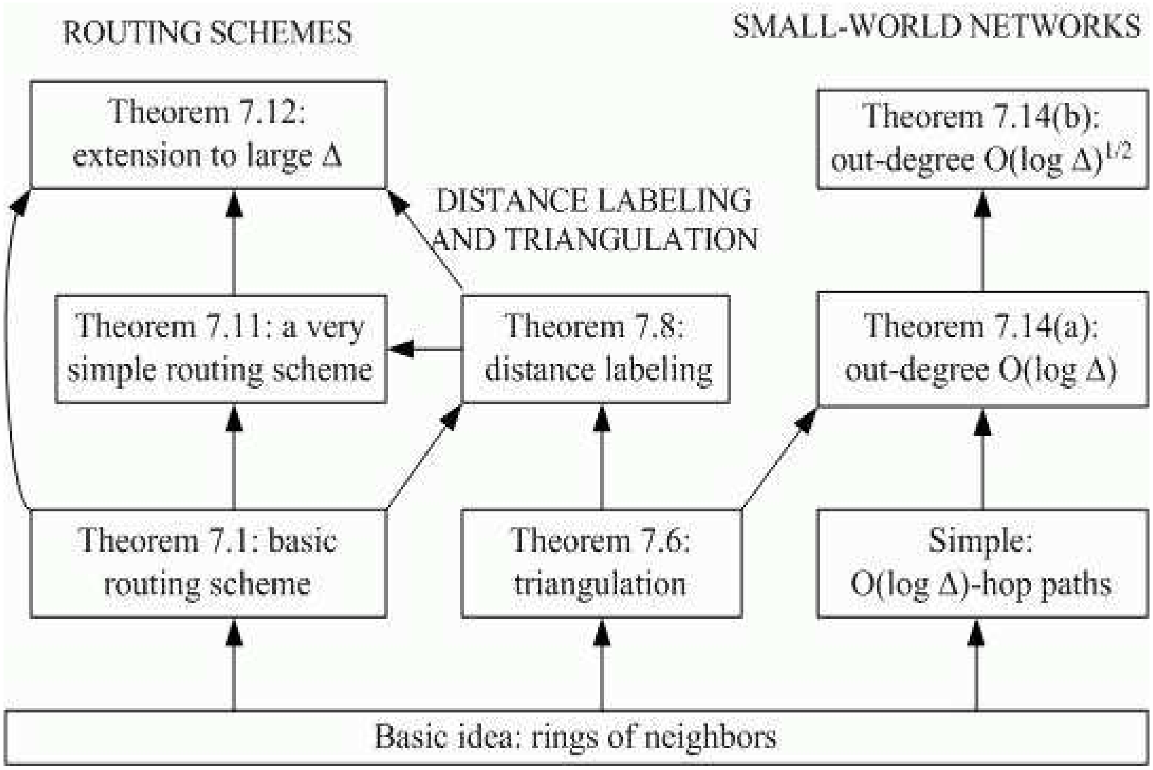
ROUTING SCHEMES
SMALL-WORLD NETWORKS
DISTANCE LABELING AND TRIANGULATION
Theorem 7.12: extension to large Δ
Theorem 7.14(b): out-degree O(log Δ)^{1/2}
Theorem 7.11: a very simple routing scheme
Theorem 7.8: distance labeling
Theorem 7.14(a): out-degree O(log Δ)
Theorem 7.1: basic routing scheme
Theorem 7.6: triangulation
Simple: O(log Δ)-hop paths
Basic idea: rings of neighbors

Figure 7.1: Interconnections between our results: arrows indicate the flow of ideas.

Recall that in *rings of neighbors*, the i-ring neighbors of a given node u lie in a ball B_i around u, for some increasing sequence of balls $\{B_i\}$; the radii of these balls and the distribution of neighbors in a given ring depend on the specific application. One trick that has been particularly useful in our proofs is to combine the following two collections of rings of neighbors. In the first collection, the cardinalities of the balls B_i grow exponentially, and the i-ring neighbors are distributed uniformly on the node set of B_i. In the second collection, the *radii* of the B_i's grow exponentially, and (if one draws on the analogy between doubling metrics and low-dimensional Euclidean metrics) the i-ring neighbors are distributed uniformly in the *space region* that corresponds to B_i. For some applications, e.g. in Section 7.2, the second collection alone suffices.

In a more abstract view, a collection of rings of neighbors is a tractable representation for the fine structure of the underlying graph. The idea of using a tractable structure-preserving representation as a unifying technique for various problems on graphs is not new; several representations have been suggested in the literature, e.g. [AGLP89, ABNLP90] for general graphs and [Tal04, MHP05] for doubling graphs. Our representation seems to be particularly suitable to the problems that we consider in this paper.

Map of the chapter. We start with a simple proof of the main result in Chan et al. [CGMZ05] on routing schemes. In Section 7.3 we present our results on triangulation and distance labeling. We return to routing schemes in Section 7.4; in particular, we consider routing schemes on metrics in Section 7.4.1. Finally, in Section 7.5 we discuss our results on searchable small-world networks.

Notation. Define an *enumeration* of a finite set S as a bijection $S \to [k]$, where $k = |S|$. Recall that for $k \in \mathbb{N}$ we define $[k]$ as the set $\{0, 1 \ldots k-1\}$.

7.2 A low-stretch routing scheme for doubling metrics

In this section we will use rings of neighbors to derive a significantly shorter and simpler standalone proof of the main result in Chan et al. [CGMZ05]; the ideas from this proof will be used in the subsequent results.

Theorem 7.1. *Consider a weighted graph G with out-degree D_{out}. Suppose its shortest-paths metric has doubling dimension α and aspect ratio Δ. Then for any $\delta \in (0, \frac{1}{4})$ there is a $(1+\delta)$-stretch routing scheme on G with $O(\alpha \log \frac{1}{\delta})(\log \Delta)$-bit packet headers and routing tables of $(\frac{1}{\delta})^{O(\alpha)}(\log \Delta)(\log D_{out})$ bits. Moreover, such routing scheme can be efficiently computed.*

Proof. Let d be the shortest-paths metric of G. For each $j \in [\log \Delta]$ let $\mathcal{G}_j$ be some $\Delta/2^j$-net on d; let $r_j = 4\Delta/\delta 2^j$ and define the *j-th ring of neighbors* of node u as the set $Y_{uj} = B_u(r_j) \cap \mathcal{G}_j$. Note that by Lemma 2.20 each node has at most $K = (16/\delta)^\alpha$ j-ring neighbors. The nodes in $\cup_j Y_{uj}$ are called the *neighbors* of u. Intuitively, we think that u has a virtual link to each of its neighbors; note that these virtual links are *not* the physical links in the underlying connectivity graph G.

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

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

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

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

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

Claim 7.2. *Given the routing table of u and the routing label of t we can find j_{ut} and g_{utj} for each $j \leq j_{ut}$.*

Now using this claim we will define the routing algorithm and prove its correctness. Then we provide a more space-efficient way to define routing tables and routing labels which will satisfy Claim 7.2 and lead to the desired storage complexity.

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

Claim 7.3. *For any two nodes (u, t) and any $j \leq \lceil \log(\Delta/\delta d_{ut}) \rceil$ we have $f_{tj} \in Y_{uj}$. In particular, for any node t and any $j \in [\log \Delta]$ letting $f = f_{(t,j-1)}$ we have $f_{tj} \in Y_{fj}$.*

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

ROUTING ALGORITHM. For a packet with target t, the header consists of the routing label of t and the number $j \in [\log \Delta]$ such that f_{tj} is the current *intermediate target*. Suppose node u wants to send a packet to target t. Then using Claim 7.2 node u computes $j = j_{ut}$ and g_{utj}, chooses f_{tj} to be the intermediate target, and sends the packet along the hop g_{utj}.

Now suppose node u receives a packet with target t and intermediate target f_{tj}. We will prove that in this case we have $j_{ut} \geq j$ (see Claim 7.4b). First node u checks whether it is the target.[6] If not, then via Claim 7.2 it computes j_{ut} and g_{utj} and, in particular, checks whether the intermediate target is u itself. If it is not, i.e. if g_{utj} is not null, then u just forwards the packet along the hop g_{utj}.

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

Claim 7.4. *Here are some key properties of the routing algorithm-*

(a) each intermediate target is at least $\frac{1}{\delta}$ times closer to the target than the previous one.

(b) if node v receives a packet with primary target t and intermediate target f_{tj} then $j_{vt} \geq j$.

(c) each packet follows a shortest path to each intermediate target.

Proof: **(a)** The next intermediate target is chosen when the current intermediate target u is reached; it is defined as f_{tj} such that $j = j_{ut}$. By Claim 7.3 $j \geq \lceil \log(\Delta/\delta d_{ut}) \rceil$, so f_{tj} lies within distance δd_{ut} from t.

(b) Let P be this packet. We will use induction on the path traversed by P. This path starts when some node u chooses $w = f_j(t)$ as an intermediate target; then the current level is set to j_{ut}. For the induction step, assume node v receives P from some node u such that $j_{ut} \geq j$; we need to show that $j_{vt} \geq j$, too. Since $w \in Y_{uj}$, and by the specs $d_{vw} < d_{uw} \leq r_j$, it follows that $w \in Y_{vj}$. It remains to show that $f_{ti} \in Y_{ui}$ for every $i < j$. Indeed, by the triangle inequality $d_{vt} \leq d_{vw} + d_{wt} < r_j + \Delta/2^j$. It is easy to check that $d_{vt} + \Delta/2^i \leq r_i$. Therefore, $f_{ti} \in B_t(\Delta/2^i) \subset B_v(r_i)$.

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

Let $\rho(u)$ be the path traversed by P from u to w, and let $\rho_L(u)$ be its metric length. We will prove that $\rho_L(u) = d_{uw}$ using induction on $\rho(u)$. Indeed, consider an edge $xy \in \rho(u)$, and assume we proved that $\rho_L(u) = d_{yw}$. Since xy is the first hop of a shortest xw-path, $d_{xw} = d_{xy} + d_{yw} = d_{xy} + \rho_L(u) = \rho_L(x)$. □

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

Claim 7.5. *Every packet reaches its target and follows a path of stretch* $1 + O(\delta)$.

Proof. Consider a packet send by node u to target t. By Claim 7.4b the algorithm is well-defined. By Claim 7.4c the packet reaches each intermediate target, and by Claim 7.4a it reaches t. The distance from the i-th intermediate target to t is at most $\delta^i d_{ut}$ by Claim 7.4a, so by Claim 7.4c the total path length is at most $\sum_{i=0} \delta^i d_{ut}(1+\delta) \leq d_{ut}(1 + O(\delta))$. □

It remains to provide space-efficient routing tables and routing labels which satisfy Claim 7.2. Recall that our goal is to replace $\lceil \log n \rceil$-bits global node identifiers with shorter 'local' identifiers.

For each node u, let us fix some enumeration $\varphi_{uj}(\cdot)$ of each ring Y_{uj}; let us call it the j-th *host enumeration* of u. Recall that an enumeration of a set S is a bijection $S \to [k]$, where $k = |S|$. Since the rings Y_{u0} coincide for all nodes u, without loss of generality so do the corresponding enumerations φ_{u0}.

Consider nodes $f = f_{tj}$ and $w = f_{(t,j+1)}$, for some target t and integer j. Note that by Claim 7.3 we have $w \in Y_{(f,j+1)}$. Consider some node u such that $f \in Y_{uj}$ and $w \in Y_{(u,j+1)}$. For such triangles (u, f, w)

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

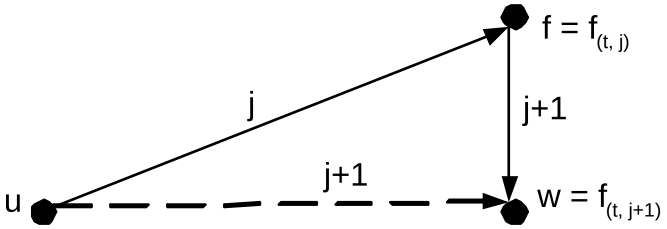

Figure 7.2: Translation between host enumerations of u and $f = f_{tj}$.

(see Figure 7.2) we will provide a 'translation' between host enumerations of u and f, in the following sense: knowing $\varphi_{uj}(f)$ and $\varphi_{(f,j+1)}(w)$ we will be able to find $\varphi_{(u,j+1)}(w)$.

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

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

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

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

For every node t, let us encode its zooming sequence via host enumerations of its elements as follows: let us define $n_{t0} = \varphi_{t0}(f_{t0})$, and for each $j \geq 1$ let $n_{tj} = \varphi_{fj}(f_{tj})$, where $f = f_{(t,j-1)}$. This is well-defined because by Claim 7.3 f_{tj} is a j-ring neighbor of node f. It is easy to see that the sequence $\{n_{tj}\}$ can be stored using $O(\log K)(\log \Delta)$ bits.

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

Having defined routing tables and routing labels, it remains to prove Claim 7.2. The proof follows in a straightforward way from our discussion of the translation functions. Indeed, let $m_j = \varphi_{uj}(f_{tj})$; this is well-defined for all $j \leq j_{ut}$. We will use induction on j to compute m_j for all $j \leq j_{ut}$. Host enumerations φ_{u0} coincide for all nodes u, so $m_0 = n_{u0}$. Suppose for some $j < j_{ut}$ we know m_j and we'd like to compute m_{j+1}. Let $f = f_{tj}$ and $w = f_{(t,j+1)}$. Since we know $m_j = \varphi_{uj}(f)$ and $n_{tj} = \varphi_{fj}(w)$, we can find $m_{j+1} = \varphi_{(u,j+1)}(w)$ using the translation function ζ_{uj}. We iterate the above procedure while we can, i.e., while $w \in Y_{(u,j+1)}$. We stop exactly at $j = j_{ut}$. This completes the proof of Claim 7.2 and Theorem 7.1. □

7.3 Triangulation and distance labeling schemes

We start with the result on triangulation, then we elaborate it using the ideas from the previous section and achieve an optimal $(1+\delta)$-approximate distance labeling scheme. We will use Lemma 3.19 on the existence of (ϵ, μ)-packings.

Theorem 7.6. *For any $\delta \in (0,1)$ any metric has a $(0,\delta)$-triangulation of order $[O(\frac{1}{\delta})]^{O(\alpha)} \log n$, where α is the doubling dimension. Moreover, such triangulation can be efficiently computed.*

Proof. The label of every node u will consist of distances to a subset of nodes which we call the *neighbors* of u. These neighbors will be partitioned into two types of rings: there will be X_i-*neighbors* and Y_i-*neighbors*,

$i \in [\log n]$. All X_i-neighbors and all Y_i-neighbors of u will be contained in the open balls $B_{(u,i-1)}$ and $B_u(12r_{ui}/\delta)$, respectively, where $r_{ui} = r_u(2^{-i})$ and $B_{ui} = B_u(r_{ui})$. This is the construction:

- For each $i \in [\log n]$ let $\mathcal{F}_i$ be a $(2^{-i}, \mu)$-packing guaranteed by Lemma 3.19, where μ is the counting probability measure. Fix one point $h_B \in B$ for every ball $B \in \mathcal{F}_i$. Define the X_i-neighbors of u as all nodes h_B such that $B \subset B_{(u,i-1)}$.
- Let us greedily construct a sequence of nested r-nets $G_{\lceil \log \Delta \rceil} \subset \ldots \subset G_1 \subset G_0$, where $\mathcal{G}_j$ is a 2^j-net for each $j \in \lceil \log \Delta \rceil$. Then for each $i \in [\log n]$ let us define the Y_i-neighbors of u as all nodes in $B_u(12r_{ui}/\delta)$ that lie in $\mathcal{G}_j$, $j = \lfloor \log(\delta r_{ui}/4) \rfloor$.

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

Let's bound the number of neighbors. Fix node u. Since each ball $B_{(u,i-1)}$ contains at most $2^{O(\alpha)}$ balls $B \in \mathcal{F}_i$, there are at most $2^{O(\alpha)}$ X_i-neighbors for each i. By Lemma 2.20 there are at most $[O(1/\delta)]^{O(\alpha)}$ Y_i-neighbors. It remains to prove that our construction is indeed a $(0, \delta)$-triangulation. First we need a basic fact about the radii r_{ui}:

Claim 7.7. *For any edge uv and any $i \in [\log n]$ we have $|r_{ui} - r_{vi}| \leq d_{uv}$.*
Proof: Since $B_v(r_{vi}) \subset B_u(d + r_{vi})$, the latter ball contains at least $n/2^i$ nodes, so it follows that $r_{ui} \leq d_{uv} + r_{vi}$. Similarly, since $B_{ui} \subset B_v(d_{uv} + r_{ui})$ it follows that $r_{vi} \leq d_{uv} + r_{ui}$. □

Fix a node pair uv and let $d = d_{uv}$. We need to show that a ball of radius δd around either u or v contains a common neighbor of both u and v. Suppose there is no such node. Let $r = (1 + \delta)d$ and choose i such that $r_{ui} < r + d \leq r_{(u,i-1)}$. We choose i with respect to u, but by Claim 7.7 this yields some bounds on r_{vj}'s as well; specifically, $r_{(v,i-1)} \geq r$ and $r_{vi} \leq r + 2d$.

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

Indeed, consider the ball $B = B_v(\delta d)$ and let $j = \lfloor \log(\delta d) \rfloor$. Then there exists a node $w \in \mathcal{G}_j \cap B$. Now since $r_{ui} \geq x$ it follows that $B \subset B_u(12r_{ui}/\delta)$ and $j \leq \log(6r_{ui})$; moreover, $j \geq \lfloor \log(\delta r_{ui}/4) \rfloor$ since $r_{ui} \leq r + d$. Therefore by definition w is a Y_i-neighbor of u. Similarly, w is a Y_i-neighbor of v, contradiction.[7] Theorem proved.

□

Our $(0, \delta)$-triangulation can be extended to a $(1 + \delta)$-approximate distance labeling scheme where each label consists of $[O(\frac{1}{\delta})]^{O(\alpha)}(\log n)(\log n + \log\log \Delta)$ bits, matching the result of Mendel and Har-Peled [MHP05]. Indeed, we assign each node u a unique $\lceil \log n \rceil$-bit identifier $\textsc{id}(u)$ and store each neighbor u of v as a pair $(\textsc{id}(u), d_{uv})$. We use the upper bound D^+ for the distance estimate, so it suffices to store d_{uv} as a $O(\log \frac{1}{\delta})$-bit mantissa and $O(\log\log \Delta)$-bit exponent.

Extending a result from [GPPR04], Mendel and Har-Peled [MHP05] constructed a family of doubling metrics for which any 1.9-approximate distance labeling scheme needs

$$\Omega(\log n)(\log\log \Delta - \log\log n) \tag{7.1}$$

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

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

Our next result shows that we can elaborate our distance labeling scheme, getting rid of the $\lceil \log n \rceil$-bit node identifiers and achieving $O_{\alpha,\delta}(\log n)(\log\log\Delta)$-bit labels. This is an improvement whenever $\log\log\Delta = o(\log n)$. Moreover, for any $\Delta \geq n^{\log n}$ and fixed α, δ we match the lower bound (7.1) up to constant factors.

Theorem 7.8. *For any $\delta \in (0,1)$ any metric has a $(1+\delta)$-approximate distance labeling scheme where each label consists of $[O(\frac{1}{\delta})]^{O(\alpha)}(\log n)(\log\log\Delta)$ bits, where α is the doubling dimension and Δ is the aspect ratio. Moreover, such scheme can be efficiently computed.*

Proof Sketch. We will elaborate the construction in the proof Theorem 7.6 using the ideas from the proof of Theorem 7.1. Specifically, we will use the zooming sequences and the host/foreign enumeration technique.

Keep the notation from the proof of Theorem 7.6. Recall that for each $j \in [\log\Delta]$ we fix some 2^j-net $\mathcal{G}_j$. For each node u and each $i \in [\log n]$ fix a node $f_{ui} \in \mathcal{G}_l$, $l = \lfloor \log(r_{ui}/4) \rfloor$, that lies within distance $r_{ui}/4$ from u. Such node is a Y_i-neighbor of u by definition of the Y_i-neighbors; it is possible that $f_{ui} = u$. Call the sequence $\{f_{ui} : i \in [\log n]\}$ a *zooming sequence*, and denote it f_u. Moreover, for each node u fix some enumeration $\varphi_u(\cdot)$ of all its neighbors.

From the proof of Theorem 7.6 we know that for any given node pair uv there exists a node w_0 within distance δd_{uv} from u or v such that w_0 is a common neighbor of u and v; recall that distances from w_0 to u and v give us a desired estimate. However, we know such w_0 exists, it is non-trivial to identify it since we do not have global node ids. In our context, to *identify* a common neighbor w of u and v means to find $\varphi_u(w)$ and $\varphi_v(w)$.

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

The problem is that $f_{(v,i+1)}$ might not be a neighbor of f_{vi}, and w_0 might not be a neighbor of f, so we cannot use the host/foreign enumeration technique the way it is used to prove Theorem 7.1. Instead, for every node we will define another set of nodes called *virtual neighbors*, so that each $f_{(v,i+1)}$ is a *virtual* neighbor of f_{vi}, and w_0 is a *virtual* neighbor of f. These virtual neighbors are used for enumeration only: for each node u fix some enumeration $\psi_u(\cdot)$ of all its virtual neighbors, and define a pointer from one 'true' neighbor u to another 'true' neighbor v as $\psi_u(v)$. It follows that virtual neighbors are cheap: if every node has at most N of them, then each pointer between the 'true' neighbors will use only $\lceil \log N \rceil$ bits. We will actually have $N = O_{\alpha,\delta}(\log n)\log(\Delta)$. [8]

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

In the remainder of this section we give the full proof of Theorem 7.8. Keep the notation from the proof of Theorem 7.6. Let X_{ui} and Y_{ui} denote the sets of X_i- and Y_i-neighbors of u, respectively; let $X_u = \cup_i X_{ui}$ and $Y_u = \cup_i Y_{ui}$. Recall that for each $j \in [\log\Delta]$ we fix some 2^j-net $\mathcal{G}_j$. For each node u and each $i \in [\log n]$ fix a node $f_{ui} \in \mathcal{G}_l$, $l = \lfloor \log(r_{ui}/4) \rfloor$, that lies within distance $r_{ui}/4$ from u. Such node lies in Y_{ui} by definition of the Y_i-neighbors; it is possible that $f_{ui} = u$. Call the sequence $\{f_{ui} : i \in [\log n]\}$ a *zooming sequence*, and denote it f_u.

[8] A minor technical note: $\log N = (\log\log n) + (\log\log\Delta) + O_{\alpha,\delta}(1)$. To avoid the ugly $(\log\log n)$ factor in the theorem statement, we note that due to Lemma 2.18 it is subsumed by $(\log\log\Delta)$.

For each node u we define the sets

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

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

Claim 7.9. *Fix node u and $i \in [\log n]$; let $r = r_{(u,i-1)}$ and $f = f_{(u,i-1)}$. Then*

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

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

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

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

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

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

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

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

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

and *null* otherwise.

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

By the proof of Theorem 7.6, the cardinality of each X_{ui}, Y_{ui} and Z_{uj} is upper-bounded by some $K = [O(1/\delta)]^{O(\alpha)}$. Therefore each node has at most $K \log n$ neighbors, and each map ζ_{ui} is represented by at most K^2 triples, each triple taking at most $2 \log K + \log |T_u|$ bits to store. Since $|T_u| < O(K^2) \log(n) \log(\Delta)$, the label size is within the claimed bounds.

Let's show how to estimate distances from the labels. As in the proof of Theorem 7.6, fix a node pair uv, denote $d = d_{uv}$, let $r = (1+\delta)d$ and choose i such that $r_{ui} < r + d \leq r_{(u,i-1)}$. It follows that $r_{(v,i-1)} \geq r$ and $r_{vi} \leq r + 2d$. We know that there exists a node w_0 within distance δd from u or v such that w_0 is a common neighbor of u and v; recall that distances from w_0 to u and v give us a desired estimate. However, we know such w_0 exists, but we do not know how to identify it: this is non-trivial since we do not have global node ids. In our context, to *identify* a common neighbor w of u and v means to find $\varphi_u(w)$ and $\varphi_v(w)$.

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

Claim 7.10. *For any $j \leq i-1$ we have $f_{uj} \in Y_{vj}$ and $f_{vj} \in Y_{uj}$.*
Proof: Let $w = f_{vj}$ and note that $w \in \mathcal{G}_l$ for $l = \lfloor \log(r_{vj}/4) \rfloor$. Since $r_{uj} \geq r + d$, by Claim 7.7 it follows that $r_{uj}/2 < r_{vj} < 2r_{uj}$ and $d_{uw} \leq d + r_{vj}/4 \leq 1.5 r_{uj}$. Therefore $w \in Y_{uj}$ by definition of Y_{uj}. Similarly, we can show that $f_{uj} \in Y_{vj}$. □

In particular, for any $j \leq i-1$ nodes f_{uj} and f_{vj} are common neighbors of u and v. Moreover, we can *identify* them sequentially using the translation maps ζ_{uj} and ζ_{vj}. For instance, it is easy to identify f_{u0} since it is numbered the same in any host enumeration. Then, inductively, suppose that we have identified some f_{uj}, $j \leq i-2$ and we need to identify $f = f_{(u,j+1)}$. Then by Claim 7.9c f is a virtual neighbor of f_{uj}, so we can find $\varphi_u(f)$ using map ζ_{uj} and (by Claim 7.10) we can find $\varphi_v(f)$ using map ζ_{vj}.

Now, assuming w_0 is closer to v than to u, we will identify it using $f = f_{(v,i-1)}$: it suffices to show that w_0 is a virtual neighbor of f. (If w_0 is closer to u, we can identify it similarly using $f_{(u,i-1)}$.) According to the proof of Theorem 7.6, letting $x = \delta d$ we can assume that either

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

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

In case (a) $w_0 \in T_f$ by Claim 7.9a since $r_{(v,i-1)} > d \geq 2x$; in case (b) $w_0 \in T_f$ by Claim 7.9b since x matches the conditions in the claim. This completes the proof of Theorem 7.8.

7.4 Low-stretch routing schemes, revisited

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

Theorem 7.11. *In the setting of Theorem 7.1, for any $\delta \in (0,1)$ there exists a $(1+\delta)$-stretch routing scheme with $2^{O(\alpha)}(\phi \log n)$-bit packet headers and routing tables of $(\frac{1}{\delta})^{O(\alpha)}(\phi \log n)(\log \Delta)$ bits, where $\phi = \log(\frac{1}{\delta} \log \Delta)$. Such routing scheme can be efficiently computed.*

Proof. For every node u, let $\textsc{id}(u)$ be its unique $\lceil \log n \rceil$-bit ID. Fix a $3/2$-approximate distance labeling scheme with $2^{O(\alpha)}(\log n)(\log\log \Delta)$-bit labels, which is guaranteed by Theorem 7.8; for each node u let L_u be the label of u in this scheme, and let $D(\cdot,\cdot)$ be the non-contracting distance function on labels. Without loss of generality assume that L_u contains $\textsc{id}(u)$. Each packet header consists of L_t and $\textsc{id}(t')$ where t is the target and t' is the *intermediate target*. The routing table of node u contains labels L_v of some nodes v which we call *neighbors* of u; we will specify them later. For each such v we also store the first node $g_u(v)$ on some shortest path to v.

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

Let's define the neighbors: for each $j \in [\log \Delta]$ let F_j be some 2^j-net; let $r_j = 2^{j+2}/\delta$ and $F_j(u) = B_u(r_j) \cap F_j$. Elements of $F_j(u)$ are called *j-level neighbors* of u; by Lemma 2.20 each node has at most $[O(\frac{1}{\delta})]^{O(\alpha)}$ j-level neighbors for each j.

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

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

Now Claim 7.4c holds: each packet follows a shortest path to each intermediate target. To reach the i-th intermediate target, $i \geq 1$, the packet traverses path at most $d_{ut}(1+2\delta)\delta^{i-1}$. Therefore the total path length is at most $d_{ut}[1+O(\delta)]$. □

We note that the bounds in Theorem 7.1 are unsatisfactory for metrics with large aspect ratio, and an extension that alleviates the dependency on Δ for weighted graphs that contain near-shortest paths with small hop-counts; this property is, intuitively, a natural property of a "good" network topology. For concreteness we will state this result for an illustrative special case; so as not to disrupt the flow, the general case and the full proof are deferred to Section 7.6.

Theorem 7.12. *Suppose the aspect ratio is 2^n and any two nodes in the input graph G are connected by a $(1+\delta)$-stretch path with at most $k \log n$ hops, where $k = (\frac{1}{\delta})^{O(\alpha)}$ and α is the doubling dimension. Then for any $\delta \in (0,1)$ there exists a $(1+\delta)$-stretch routing scheme on G with $O(k \log^3 n)$-bit routing tables and $O(k \log^2 n)$-bit packet headers. Such routing scheme can be efficiently computed.*

Proof Sketch. We will combine the ideas of Theorem 7.8 and Theorem 7.1, and add some new tricks. In particular, we will use (i) the basic rings of neighbors, (ii) zooming sequences and intermediate targets, (iii) the first-hop pointers, and (iv) host/virtual enumerations. We will use the rings, the zooming sequences, and the enumerations as defined in Theorem 7.8. In fact, we will just use all definitions from the proof Theorem 7.8, for the same value of δ, including the sets of X_i- and Y_i-neighbors. We also need a unique $\lceil \log n \rceil$-bit identifier $\text{ID}(u)$ for every node u.

The routing will have *two modes*. One is an elaboration of the routing in Theorem 7.1: we use intermediate targets that zoom in towards the true target. If at the current node u the intermediate target is not set, we select a new intermediate target w among the neighbors of u, using the zooming sequence f_t and other data in the routing label of t. To save space in the packet header, this w will be represented not by a global id, but by its number in a virtual enumeration of some f_{ti}. Now suppose an intermediate target w is set, and the packet is at node v. If w is a neighbor of v and, moreover, v can *identify* this w (i.e. find $\varphi_v(f)$), then v forwards the packet using the first-hop pointer to w.

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

If the first routing mode fails, we will go into the *second routing mode*, and we never come back. By Lemma 3.19 there exists a ball $B \in \mathcal{F}_i$ of cardinality at least $n/2^{i+O(\alpha)}$ such that $B \subset B_u(6r_{ui})$. Let $w = h_B$ be the node selected from B in Theorem 7.6; without loss of generality say it is a center of B. It is easy to see that the ball $B' = B_{(h,i-1)}$ contains target t. The nodes in B will collectively store the routes to all nodes in B'; specifically, each node in B will store full routes to $2^{O(\alpha)}$ nodes in B' so that exactly one node in B is responsible for each node in B'. Moreover, the nodes in B will maintain a labeled shortest-paths tree T_B rooted at w, such that given $\text{ID}(t)$ it is possible to route from w to the node v_t that stores a path to t. Here it is crucial that we are free to choose the labels for T_B and the mapping v_t from B' to B any way we like. We will choose so that for a given link in the shortest-paths tree it suffices to specify a single *range* of target ids for which a packet should take that link.

This is how the packet will reach t. First the node w (which is a neighbor of u) is designated as the intermediate target, and the packet is routed to w via the first-hop pointers. From w the packet is routed to v_t via the shortest-paths tree T_B. Then node v_t puts the full route to t into the packet header and send the packet to t. More precisely, v_t will store a $(1+\delta)$-approximate shortest path to t with $k \log n$ hops (the

	out-degree	routing table size, bits	packet header size, bits
Chan et al. [CGMZ05]	$(\frac{\alpha}{\delta})^{O(\alpha)}(\log\Delta)$	$(\frac{\alpha}{\delta})^{O(\alpha)}(\phi\log\Delta)$	$O(\alpha\log\frac{1}{\delta})(\log\Delta)$
Theorem 7.1	$(\frac{1}{\delta})^{O(\alpha)}(\log\Delta)$	$(\frac{1}{\delta})^{O(\alpha)}(\phi\log\Delta)$	same as above
Theorem 7.11	same as above	$(\frac{1}{\delta})^{O(\alpha)}(\phi\log\Delta)(\log n)$	$2^{O(\alpha)}(\phi\log n)$
Theorem 7.12	$(\frac{1}{\delta})^{O(\alpha)}(\log n)$	$(\frac{1}{\delta})^{O(\alpha)}(\phi\log n)(\log\log n)$	$O(\alpha\phi\log n)$
Follow-up work [AGGM06]	$(\frac{1}{\delta})^{O(\alpha)}(\log n)$	$(\frac{1}{\delta})^{O(\alpha)}(\log^2 n)$	$\lceil\log n\rceil$
	doubling dimension α, aspect ratio Δ, and $\phi=\log(\frac{1}{\delta}\log\Delta)$		

Table 7.2: $(1+\delta)$-stretch routing schemes for doubling metrics

existence of such a path is guaranteed by the theorem statement). Each hop in this path can be encoded by $\lceil\log D_{\text{out}}\rceil \leq \lceil\log n\rceil$ bits, where D_{out} is the out-degree, so the entire path can be stored using at most $k\log^2 n$ bits.

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

7.4.1 Routing schemes on metrics

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

Extension to routing on metrics is almost trivial. In fact, in all our proofs we first construct a routing scheme on a low-degree overlay network (which is, by definition, a routing scheme on a metric), and then with some additional work adapt it to the underlying connectivity graph. The quantitative results are summarized in Table 7.2; we omit the appropriate modifications (simplifications) of the proofs. Note that in this setting Theorem 7.12 does not need any assumptions about the existence of near-shortest paths.

FOLLOW-UP WORK: Following the publication of the conference version of Slivkins [Sli05a], Abraham et al. [AGGM06] provided a fine-tuned version of Theorem 7.12, where they completely eliminate the dependence on aspect ratio Δ; see Table 7.2 for quantitative results.

7.5 Searchable small-world networks

In this section we consider searchable small-world networks on *metrics*. To the best of our knowledge, the most general previous result in this direction is for metrics such that the growth rate of balls (defined as the ratio $|B_u(2r)|/|B_u(r)|$) is both upper- and lower-bounded by a constant; let us call such metrics *UL-constrained*. This result can be easily achieved from Kleinberg's original construction for two-dimensional grids [Kle00b]. Here we extend small worlds to doubling metrics.

We will consider routing algorithms where the next hop is chosen among the current node's contacts, by looking only at distances to these contacts and distances from these contacts to the target. Recall from Section 7.1 that we call such routing algorithms *strongly local*. A very natural routing algorithm is the *greedy* algorithm- select the contact that is closest to the target.

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

Definition 7.13. A ***small-world model*** on a metric (V, d) consists of the following two items:

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

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

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

Now we can describe our results. Let Δ be the aspect ratio of the metric. While it is relatively straightforward to achieve out-degree $O(\log n)(\log \Delta)$ and $O(\log \Delta)$-hop paths, it is quite non-trivial to handle the case of super-polynomial aspect ratio. We obtain $O(\log n)$-hop paths even if Δ is exponential in n. In our first result the out-degree is (still) proportional to $\log(\Delta)$, suggesting that it is a natural lower bound since we need some long-range contacts for each one of the $(\log \Delta)$ distance scales. However, our second result breaks this barrier. Moreover, in Section 7.5.2 we argue that for UL-constrained metrics our small worlds essentially coincide with those induced by Kleinberg's *group structures* from [Kle01].[9]

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

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

Theorem 7.14. *Let α be the doubling dimension, and let Δ be the aspect ratio.*

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

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

Proof Sketch. To be consistent with the earlier parts of the paper, let us use words 'contact' and 'neighbor' interchangeably. A relatively straightforward solution is to use $(\log \Delta)$ rings of neighbors so that the radii of the rings grow exponentially, and the neighbors are distributed with respect to the doubling measure; let us call these neighbors the *Y-type neighbors*. It is easy to make sure that the greedy algorithm reduces the distance by at least a factor of two at each step, so any query will take $(\log \Delta)$ steps to complete. However,

[9] The guarantees in [Kle01] apply to UL-constrained metrics that are subsets of some ℓ_p space, $p \geq 1$. However, the construction itself is well-defined for any metric.

reducing the distance by a constant factor at each step does not suffice to guarantee $O(\log n)$-hop query paths when the aspect ratio Δ is large.

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

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

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

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

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

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

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

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

7.5.1 Full proof of Theorem 7.14

Let us fill in the details. For simplicity let us assume that in the input metric all distances are distinct. Recall that $B_{ui} = B_u(r_{ui})$, where $r_{ui} = r_u(2^{-i})$ is the radius of the smallest ball around u that contains at least $n/2^i$ nodes. Fix an absolute constant c to be specified later. Recall that α is the doubling dimension; let μ be the $2^{O(\alpha)}$-doubling measure. For each $i \in [\log n]$ select a node independently and uniformly at random from the ball B_{ui}. Repeat this $c \log n$ times, where c is a sufficiently large constant to make the Chernoff Bounds work out (see Footnote 10 below), and let X_{ui} be the set of selected nodes. Let $X_u = \cup X_{ui}$; these are the *X-type neighbors* of u. Note that $|X_u| \leq O(\log^2 n)$.

Proof of part (a): For a given node u, select the neighbors as follows. For each $j \in [\log \Delta]$ select a node independently from the ball $B = B_u(2^j)$ according to the probability distribution $\mu(\cdot)/\mu(B)$; repeat this $(2^{c\alpha} \log n)$ times, and let Y_{uj} be the set of selected nodes. Let $Y_u = \cup Y_{uj}$; these are the ***Y-type neighbors*** of u. Define the set of neighbors of u as $X_u \cup Y_u$. Note that the out-degree of u is within the claimed bound; in particular, we upper-bound $|X_u|$ using Lemma 2.18.

We need to prove that property (*) holds. Suppose t is the target and u is the current node. Let us choose i such that node u lies in the annulus $B_{(t,i-1)} \setminus B_{ti}$. Let us denote $d = d_{ut}$ and $j = \lceil \log(1.25\, d) \rceil$. Note that the set Y_{uj} contains a node w that is within distance $d/4$ from target t.[10] Therefore the greedy routing algorithm will choose such node for the next hop.

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

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

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

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

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

let us select a node independently from the ball $B = B_u(r_{ui} \cdot 2^j)$ according to the distribution $\mu(\cdot)/\mu(B)$. Repeat this $(2^{c\alpha} \log n)$ times and let $Y_{(u,i,j)}$ be the set of selected nodes. Let $Y_u = \cup_{ij} Y_{(u,i,j)}$; these are the ***Y-type*** neighbors of u. Define the set of neighbors of u as $X_u \cup Y_u \cup Z_u$.

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

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

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

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

$$r_{ul} \cdot 8^{x+1} < 1.25\, d < r_{(u,l-1)}/8^{x+1}. \tag{7.2}$$

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

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

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

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

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

$$d/8^x \leq d_{uw} \leq d. \tag{7.3}$$

In particular, by (7.2) and (7.3) it follows that

$$4r_{ul} < d_{uw} < r_{(u,\,l-1)}/4. \tag{7.4}$$

Now that we are at w we will be able to make progress towards t. To be consistent with our search algorithm, the next hop should get us from w to within distance $d_{wt}/4$ from t. Since $d_{wt} > d/4$ by our assumption, it suffices to get inside the ball $B_t(d/16)$. (Note that if the routing algorithm is allowed to remember the previous move, then getting inside $B_u(d/4)$ is sufficient, too.) We will achieve the desired progress using some neighbor in $Y_{(w,l,j)}$ for the appropriately chosen j.

Claim 7.15. $d_{uw} - r_{ul} \leq r_{wl} \leq d_{uw} + r_{ul}$.

Proof. The second inequality follows since the ball $B_w(d_{uw} + r_{ul})$ contains the ball B_{ul} and therefore has cardinality at least $n/2^l$. Suppose the first inequality fails. Then the balls B_{wl} and B_{ul} are disjoint; since both balls lie inside $B_u(d_{uw} + r_{wl})$, the latter ball has cardinality at least $n/2^{l-1}$. It follows that $r_{(u,l-1)} \leq d_{uw} + r_{wl}$. However, using (7.4) we have $d_{uw} + r_{wl} \leq 2d_{uw} + r_{ul} < r_{(u,l-1)}$, contradiction. □

Combining Claim 7.15 and (7.4), it follows that $r_{wl}/d_{uw} < (\frac{4}{3}, \frac{5}{3})$. Let us denote $r = d_{wt} + d/16$. Then

$$r \leq 1.07\, d + d_{uw} \leq d_{uw}\,(1.07 \cdot 8^x + 1) < 2^{3x+1}\, r_{wl} \tag{7.5}$$

In (7.5) the first inequality follows simply because $d_{wt} \leq d + d_{uw}$, and the second inequality holds by (7.3).

Let us choose j such that $2^{j-1} < r/r_{wl} \leq 2^j$. Then by (7.5) we have $j \leq 3x + 2$, and by definition of r we have $B_t(d/16) \subset B_w(r_{wl} \cdot 2^j)$. The radii of these two balls are within a constant factor because

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

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

7.5.2 Comparison with Kleinberg's small worlds

Let us argue that our small-world models generalize one of the Kleinberg's small worlds. Specifically, we consider the *group structures* from [Kle01] applied to balls in a metric (it was one of the two original applications described in [Kle01]). This small-world model, call it STRUCTURES, can be defined as follows. For any two nodes (u, v), let x_{uv} be the smallest cardinality of a ball containing both u and v. For each node u, define a probability distribution π_u on V (the set of all nodes) by $\pi_u(v) = c_1/x_{uv}$, where c_1 is the suitable normalization constant. Each node u has $\Theta(\log^2 n)$ neighbors chosen independently from distribution π_u. The routing algorithm is greedy.

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

Theorem 7.16. *For UL-constrained metrics, both small-world models in Theorem 7.14 share the following properties with* STRUCTURES*:*

(a) *with high probability, any target is found in* $O(\log n)$ *steps from any starting node.*
(b) *the local search algorithm is greedy.*
(c) *each node has* $k = \Theta(\log^2 n)$ *neighbors.*
(d) $\Pr[v$ *is a neighbor of* $u] = \Theta(\log n)/x_{uv}$*, for any nodes* (u, v).[11]

Proof Sketch. Part (a) is trivial because any UL-constrained metric has a polynomially bounded aspect ratio. For part (b) note that the routing algorithm in Theorem 7.14a is greedy by definition, and in Theorem 7.14b the non-greedy step is taken *only* if there is no neighbor that would reduce the distance to the target by the factor of 4. It is easy to show that if the underlying metric is UL-constrained then the set $X_u \cup Y_u$ will contain such a neighbor, so in Theorem 7.14b the routing algorithm is greedy as well and, moreover, the Z-type neighbors are never used.

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

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

By (iii), in Theorem 7.14b for every node u and each $i \in [\log n]$ there is at most a constant number of non-empty sets Y_{uij} (and obviously, there is at least one such set). Part (c) follows immediately.

In both parts of Theorem 7.14, for each node u we sample $\Theta(\log n)$ neighbors (namely, the X-type neighbors) uniformly at random from each of the balls $\{B_{ui},\, i \in [\log n]\}$. Here a given node v is selected with probability $\Theta(\log n)/|B_u(d_{uv})|$, which by (ii) is $\Theta(\log n)/x_{uv}$.

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

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

Let us briefly comment on an alternative setting where we are given a graph of local contacts, and we add exactly one long-range contact per node. This has been the original Kleinberg's model [Kle00b] (for two-dimensional grids). Recently (following the publication of the conference version of Slivkins [Sli05a]) it has been considered in [DHLS05, Fra05, AG06, FLL06] for, respectively, graphs that induce metrics of low grid dimension, graphs of bounded treewidth, graphs that exclude a fixed minor, and doubling graphs. The typical guarantee is that any query completes in *expected* poly-log number of hops.

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

Theorem 7.17. *Consider a graph* G *such that its shortest paths metric* d_G *has doubling dimension* α*. There is a randomized algorithm that assigns to every node exactly one long-range contact so that in the resulting small-world model on* d_G *the greedy algorithm completes each query in expected* $2^{O(\alpha)}(\log^2 \Delta)$ *hops.*

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

[11] For Theorem 7.14b we ignore Z-type neighbors since it turns out that on doubly constrained metrics they never get used.

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

Recall that Theorem 7.14 explored the interesting trade-off between the out-degree and the hop-count. Here, in Theorem 7.17, in order to make progress, a success event at any one node suffices; so if we allow larger out-degree, then the product of hop-count and out-degree stays constant. This seems a good way to capture the above-mentioned tradeoff. Unfortunately, it does not work in general. For instance, if we adapt Theorem 7.14(a) or Theorem 7.14(b) to the current setting then in order to make progress we need success events at two (resp. three) *consecutive* nodes. This results in poor probability of making progress at a given node, and, accordingly, in an unreasonably poor expected hop-count, as compared to a much less sophisticated Theorem 7.17. These considerations suggest that sometimes the current setting does not quite capture the richer setting of polylog out-degree.

7.6 Full proof of Theorem 7.12 on routing schemes

In this section we give the proof of Theorem 7.12 from Section 7.4. We moved it to the end of this chapter in order not do disrupt the flow of exposition.

We prove the theorem in the following more general form:

Theorem 7.18. *Suppose any two nodes in the input graph G are connected by a $(1+\delta)$-stretch path with at most N_δ hops. Let α be the doubling dimension, let Δ be the aspect ratio, and let D_{out} be the out-degree of G. Then for any $\delta \in (0,1)$ there exists a $(1+\delta)$-stretch routing scheme on G with*

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

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

We will combine the ideas of Theorem 7.8 and Theorem 7.1 with some new tricks. We will use (i) the basic rings of neighbors, (ii) zooming sequences and intermediate targets, (iii) the first-hop pointers, and (iv) host/virtual enumerations. Our basic setup is from the proof of Theorem 7.8. For simplicity let's assume $\delta \leq 1/8$ and let $\delta' = \delta/(1-\delta)$.

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

- From Theorem 7.6, we borrow, firstly, radii r_{ui} and balls B_{ui}; secondly, $(2^{-i}, \mu)$-packings $\mathcal{F}_i$ and sets X_{ui} of X_i-neighbors; and thirdly, 2^j-nets $\mathcal{G}_j$ and sets Y_{ui} of Y_i-neighbors.
- From Theorem 7.8, we borrow the zooming sequences $f_u = \{f_{ui} : i \in [\log n]\}$, the sets T_u of virtual neighbors; host enumerations $\varphi_u(\cdot)$, virtual enumerations $\psi_u(\cdot)$, and translation functions ζ_{ui}. For convenience set $\psi_u(v) = \mathit{null}$ whenever v is not a virtual neighbor of u.
- From Theorem 7.1 we borrow the first-hop pointers.

We use $(2^{-i}, \mu)$-packings $\mathcal{F}_i$ as provided by Lemma 3.19; for each $B \in \mathcal{F}_i$, let $h = h_B$ be a node and $r = r_B$ be a radius such that $B = B_h(r)$ and $6r_u(2^{-i}) \geq d_{uh} + r$. We need to fix h because B can have multiple centers, i.e. nodes v such that $B = B_v(r)$ for some r, whereas Lemma 3.19 guarantees this inequality only for one of them. We redefine the set X_{ui} of X_i-neighbors of u as follows as the set of all nodes $h = h_B$ such that $B \in F_i$ and $r_{(u,i-1)} \geq d_{uh} + r_B$.

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

- $\textsc{id}(t)$ as a unique global $\lceil \log n \rceil$-bit identifier for t;
- x_{ti} as the nearest X_i-neighbor of t;

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

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

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

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

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

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

The *routing table* of each node u includes:

- its label, radii r_{ui} for all i, and distances to all its neighbors (but not to its virtual neighbors),
- translation maps ζ_{ui}, for all $i \in [\log n]$.
- the first-hop pointer from u to each neighbor of u, which we can store using only $\lceil \log D_{\text{out}} \rceil$ bits.

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

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

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

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

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

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

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

First we show that (u, i, j)-landmarks and u-good nodes exist, then we show how to identify them. The following claim is an elaboration of the arguments in the proof of Theorem 7.8.

Claim 7.19. *Fix any nodes u and t, and let $d = d_{ut}$.*

(a) If $r_{ul} \geq \frac{4}{3} d_{ut}$ for some l then f_{tl} is a Y_l-neighbor of u.

(b) if $\delta d/6 \leq r_{ui} < 2d \leq r_{(u,i-1)}$ for some i, then there exists a u-good node.

Proof: **(a)** Let $d = d_{ut}$. Note that $w = f_{tl} \in \mathcal{G}_j$, $j = \lfloor \log(r_{tl}/4) \rfloor$, and by Claim 7.7 we have $|r_{ul} - r_{tl}| \leq d$. By definition of Y_l-neighbors, we need to check two things: that $d_{uw} \leq 12 r_{ul}/12$ and that $l \in J_{ul}$. Firstly,

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

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

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

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

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

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

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

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

so $w \in X_{ui}$. Since $r_{(t,\, i-1)} \geq r_{(u,\, i-1)} - d \geq d > 12r_{ti}$, by Claim 7.9a w is a virtual neighbor of f.

Now suppose $r_{ti} > x/6$. Note that $r_{ti} \leq r_{ui} + d < 3d$, so $x \in [\frac{\delta}{4};\, r_{ti} 6r_{ti}]$. Then $w = y_{tj} \in \mathcal{G}_j \cap B_t(z)$ satisfies all conditions in Claim 7.9b, hence is a virtual neighbor of f. Finally, u is a Y_i-neighbor of u since $12r_{ti}/\delta > 2d > d + d_{wt} \geq d_{uw}$ and $j = \lfloor \log x \rfloor \geq \lfloor \log \delta r_{ti}/4 \rfloor$. □

Claim 7.20. *Given the routing table of u and the routing label of t, one can efficiently:*

(a) check whether a u-good node w exists; if so, find $\varphi_u(w)$ and (i,j) such that w is (u,i,j)-good.

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

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

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

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

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

The first routing mode is an elaboration of the routing algorithm in the proof of Theorem 7.1. In this mode the packet is routed to an *intermediate target* w, until it reaches w or t, or switches to $\mathcal{M}_2$, or a new intermediate target is chosen. If the current intermediate target w has been chosen at node u, then the packet header contains the routing label of t, the distance $D_{\text{est}} = d_{uw}$, and the *intermediate target id*, which is a pair (i,j) such that w is (u,i,j)-good.

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

- If the intermediate target id is *null*, u checks whether a u-good node w exists; if so, u finds $\varphi_u(w)$ and a pair (i,j) such that w is (u,i,j)-good (see Claim 7.20a). If u-good nodes do not exist, the routing switches to $\mathcal{M}_2$. Else, u chooses w as the next intermediate target, sets $D_{\text{est}} = d_{uw}$, and sets the intermediate target id to (i,j).

- If the intermediate target id is (i,j), then u checks whether the (u,i,j)-landmark node w exists (see Claim 7.20b), finds $\varphi_u(w)$ it if it does, or switches to $\mathcal{M}_2$ if it doesn't.

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

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

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

Claim 7.21. *Let $u_0, u_1, \ldots, u_{k-1}$ be the nodes where the new intermediate target id has been set; let u_k be the last node that the packet has reached in $\mathcal{M}_1$. Then for a fixed $i < k$ we have:*

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

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

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

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

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

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

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

(c) The proof is similar to that of Claim 7.4, but somewhat more complicated since u_{i+1} is not necessarily equal to w. Let $u = u_i$ and $v = u_{u+1}$. Let $\rho(x)$ be the path traversed by the packet from node x to v; let $\rho_L(x)$ be the metric length of this path. We need to show that $\rho_L(u) = d_{uv} = d_{uw} - d_{vw}$.

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

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

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

SWITCHING BETWEEN THE MODES. It is crucial that the routing switches from $\mathcal{M}_1$ to $\mathcal{M}_2$ only if for the current node a certain condition (Lemma 7.22) holds. We will see later that under this condition $\mathcal{M}_2$ work efficiently. The forthcoming Lemma 7.22 is really the crux of the proof of Theorem 7.12.

Lemma 7.22. *Suppose the routing switches to $\mathcal{M}_2$ at node v. Then $6r_{vi}/\delta < \frac{4}{3}d_{vt} \le r_{(v,i-1)}$ for some i.*

Proof: Suppose such i does not exist. Let u be the last node that receives the packet in $\mathcal{M}_1$ with null intermediate target id. If $u = v$ then for i such that $r_{ui} < \frac{3}{2}d_{ut} \le r_{(u,i-1)}$ we must have $6r_{ui} \ge \frac{4}{3}\delta d_{ut}$, so by Claim 7.19b there exists a u-good node, contradiction. Therefore $u \ne v$. It follows that:

- the routing did not switch to $\mathcal{M}_2$ at u, so u has set the intermediate target id to a pair (i,j) such that there exists a (u,i,j)-good node w.

- node v received the packet with a non-null intermediate target id (equal to (i,j)), so it must be the case that $d_{xw} - d_{xv} > 2\delta' d_{uw}$, where x is the node visited by the packet immediately before v.

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

We need to check conditions (c1-c3) in the definition of a (v,i,j)-landmark. For condition (c3), we claim that for each $l \leq i-1$ we have $f_{tl} \in Y_{vj}$. Indeed, since w is (u,i,j)-good, it follows that $d_{wt} \leq \delta' d_{uw}$ and $r_{ul} \geq 2d_{uw}(1-\delta')$. By Claim 7.21c node v lies on a shortest uw-path, so $d_{uv} + d_{vw} = d_{uw}$. Moreover, $r_{vl} \geq r_{ul} - d_{uv}$ by Claim 7.7. Putting this all together and letting $\beta = \frac{4}{3}$, we have:

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

so the claim follows by Claim 7.19a.

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

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

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

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

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

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

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

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

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

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

SECOND ROUTING MODE. Suppose routing switches to $\mathcal{M}_2$ at node u; let $d = d_{ut}$. By Lemma 7.22 for some i it is the case that $6r_{ui}/\delta < \frac{4}{3}d \leq r_{(u,i-1)}$. By Lemma 3.19 there exists a ball $B \in \mathcal{F}_i$ of cardinality at least $n/2^{i+O(\alpha)}$ such that $B \subset B_u(6r_{ui})$. Let $w = h_B$ be the node selected from B in Theorem 7.6; recall that it is a center of B. It is easy to see that the ball $B' = B_{(w,\,i-1)}$ contains target t. Indeed, $d_{uw} \leq 6r_{ui} \leq \frac{4}{3}\delta d \leq d/6$ since $\delta \leq \frac{1}{8}$, and by Claim 7.7

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

The nodes in B will collectively store the routes to all nodes in B'; specifically, each node in B will store full routes to $2^{O(\alpha)}$ nodes in B'. Moreover, the nodes in B will maintain a shortest-paths tree T_B rooted at h. We label the edges of T_B so that given $\textsc{id}(t)$, $t \in B'$ it is possible to route from h to the node $v_t \in B$ that stores a path to this $t \in B'$. Specifically, we label each node v with a range R_v such that if a packet is at u, and edge $uv \in T_B$, and $\textsc{id}(t)$ is within this range, then the packet is forwarded to v.

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

Table 7.3: Space requirements; let $\phi=\log(\frac{1}{\delta}\log\Delta)$.

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

This is how the packet will reach target t. First the node h (which is a neighbor of u) is designated as the intermediate target, and the packet is routed to h via the first-hop pointers. From h the packet is routed to v_t via the shortest-paths tree. Then v_t puts the full route to t into the packet header and send the packet to t. More precisely, v_t will store a $(1+\delta)$-approximate shortest path to t with the smallest hop count, which is at most N_δ by definition of N_δ. Each hop in this path can be encoded by $\lceil\log D_{\text{out}}\rceil$ bits, where D_{out} is the maximal degree of the underlying connectivity graph, so the entire path can be stored using at most $N_\delta\lceil\log D_{\text{out}}\rceil$ bits. Since a given node can lie in only one ball $B\in F_i$, it has to store at most $2^{O(\alpha)}$ paths for each i, for a total of at most $2^{O(\alpha)}\log n$ paths. This completes the second routing mode.

Claim 7.23. *If the routing switches to $\mathcal{M}_2$ at node u, then from u to t it has stretch $1+O(\delta)$.*

PROOF OF CORRECTNESS. The space requirements of both routing modes are summarized in Table 7.3. We need to show that our routing scheme has stretch $1+O(\delta)$. If the packet reaches the target without switching to $\mathcal{M}_2$, this follows from Claim 7.21. Now suppose it switches to $\mathcal{M}_2$ at node w in the middle of a path to some intermediate target v. Let u be the node that set v as the intermediate target and let $d=d_{ut}$. Let ρ_{xy} be the distance traversed by the packet on its path from node x to node y.

By Claim 7.23 $\rho_{wt}/d_{wt}\leq 1+O(\delta)$. By Claim 7.21ab $v\in B_t(6\delta d)$. By Claim 7.21c, node w lies on some shortest path from u to v, and the packet followed this path from u to w. Putting this together, we get

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

Suppose the packet originated at node s. If $s=u$ then we are done. If $s\neq u$ then by Claim 7.21 $\rho_{su}\leq(1+O(\delta))\,d_{st}$ and by Claim 7.21ab $d\leq\delta d_{st}$. Therefore,

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

as claimed. This completes the proof of Theorem 7.18.

Chapter 8

Conclusions and further directions

Concurrent with numerous theoretical results on metric embeddings, a growing body of research in the networking community has studied the distance matrix defined by node-to-node latencies in the Internet, resulting in a number of recent approaches that approximately embed this distance matrix into low-dimensional Euclidean space. A fundamental distinction between the theoretical approaches to embeddings and this recent Internet-related work is that the latter operates under the additional constraint that it is only feasible to measure a linear number of node pairs, and typically in a highly structured way. Indeed, the most common framework here is a *beacon-based* approach: one randomly chooses a small number of nodes ('beacons') in the network, and each node measures its distance to these beacons only. Moreover, beacon-based algorithms have been designed for the more basic problem of *triangulation*, in which one uses the triangle inequality to infer the distances that have not been measured.

In this dissertation we provide a theoretical foundation for distributed distance reconstruction approaches, with extensions to metric embeddings and node-labeling problems. We give beacon-based algorithms with provable performance guarantees for triangulation and embedding; in addition to multiplicative error in the distances, such guarantees typically must include a notion of "slack" – an ϵ-fraction of all distances may be arbitrarily distorted, which is a novel notion for theoretical work on metric embeddings. We extend these results in a number of directions: embeddings with slack that work for all ϵ at once; distributed algorithms for triangulation and embedding with low overhead on all participating nodes; distributed triangulation with guarantees for all node pairs; node-labeling problems for graphs and metrics; extensive provable guarantees for a systems project on location-aware node selection in a large-scale distributed network.

This thesis is based on a line of work started in [KSW04]. This paper gave rise to several open questions which motivated, and got resolved in, a number of the subsequent papers. Specifically, we have been wondering whether there exist: ϵ-slack embedding for arbitrary metrics (the original result on ϵ-slack embeddings was for doubling metrics); fully distributed embeddings; triangulation-style data structure with good guarantees for all edges; fully distributed triangulation with guarantees for all edges. These questions have been addressed respectively, in [CDG$^+$05, Sli05b, Sli05a, Sli06]. Moreover, the existence of gracefully degrading embeddings for arbitrary metrics has been the main technical contribution of a recent effort [ABN06] by another research group.

The above line of work has also motivated a number of new new open questions:

Gracefully degrading distortion. The result in [ABN06] is that any metric can be embedded into any ℓ_p, $p \geq 1$ with gracefully degrading distortion $O(\log \frac{1}{\epsilon})$; this is optimal up to constant factors. Recall that in Chapter 4 we obtained $O(\log \frac{1}{\epsilon})^{1/p}$ for decomposable metrics. An intriguing open question is whether $o(\log \frac{1}{\epsilon})$ is possible for other families of metrics, too. In particular, is gracefully degrading distortion

$\tilde{O}(\sqrt{\log 1/\epsilon}\,)$ possible for embedding finite subsets of ℓ_1 into ℓ_2? This would mirror the corresponding $\tilde{O}(\sqrt{\log n}\,)$ distortion result [ALN05] which is closely related to the recent break-through on the sparsest cut problem [ARV04, AHK04].

Triangulation. Several open questions concern triangulation.

First, while our strong guarantees are for doubling metrics, it is not clear how crucial this restriction is. In particular, what can we say about a metric that allows a good triangulation? Can we achieve good triangulation for other families of metrics?

Second, recall that our result in Chapter 7 achieves $(0, \delta)$-triangulation of order $O_{\alpha,\delta}(\log n)$, where α is the doubling dimension. However, the lower bound (7.1) on distance labeling (see Section 7.3), which is the only lower bound for triangulation that we have, does not preclude triangulations of order $O_{\alpha,\delta}(1)$ for polynomially bounded aspect ratio, and triangulations of order $O_{\alpha,\delta}(\log\log n)$ otherwise. Can we provide doubling metrics with a triangulation-specific lower bound of $\Omega(\log n)$, or, alternatively, construct triangulations of sub-logarithmic order? Intuitively, the latter would be very surprising. Indeed, consider balls around a given node u. Then there are $\Omega(\log n)$ exponentially increasing size scales, and at least as many exponentially increasing distance scales. If the size scales are roughly aligned with the distance scales, then, intuitively, a label of u should include distances to at least one node in each of these scales.

Distributed algorithms. An array of open questions concerns the distributed framework described in Chapter 3. One direction here is to provide provable guarantees for decentralized network algorithms that build on triangulation or virtual coordinates. This might require a more fine-tuned (and possibly application-specific) notion of slack. Another direction is to design decentralized algorithms for triangulation and embedding that can handle churn in the system. In particular, consider the simple setting of an *online embedding*: nodes arrive sequentially; each node must compute its coordinates based on distances to (and coordinates of) the previously arrived nodes. Finally, in some settings it could be possible to get rid of the ϵ-slack, i.e. obtain guarantees for all node pairs. Recall that we did obtain such result for fully distributed triangulation in growth-constrained metrics. Can we obtain a similar result for embeddings? Is it possible to extend these result to a more general family of metrics?

Rings of neighbors. Rings of neighbors can be used in a distributed system as a layer that supports various applications. Recall that we used this framework in Chapters 5, 6 and 7. While this framework has already lead to significant results, rings that we can define theoretically are much 'better' than the ones that we know how to construct in a distributed fashion (either in theory or in practice). Closing this gap is an interesting open question.

Internet latencies. In Chapter 3 we obtained strong guarantees for triangulation in an overlay network, assuming that the matrix of Internet latencies is a doubling metric. To which extent do the Internet latencies resemble a doubling metric? More generally, what are the special properties of this distance matrix? A clean and experimentally sound model for Internet latencies is fundamental for theoretical studies of algorithms operating on the Internet. However, there has been essentially no work on this question, beyond the study of the triangle inequality violations. This is a multi-facet research direction that involves systems issues of gathering sufficiently large and clean data sets, analytical issues of inventing the right set of properties to look at, algorithmic issues of mining for these properties in the large and (potentially) incomplete and dirty data, and statistical issues of modeling and filtering out the noise.

Node labeling problems. Several open questions concern the node labeling problems considered in Chapter 7: routing schemes, small-world networks, and distance labeling.

First, for routing schemes on graphs and for searchable small-world networks it is desirable to further alleviate the dependency on the aspect ratio Δ, e.g. by replacing the $(\log \Delta)$ factor by $(\log n)(\log \log \Delta)$ like we did for distance labeling schemes and routing schemes on metrics. A more ambitious task is to obtain poly-log(n) upper bounds that do not depend Δ altogether. After the conference version of Slivkins [Sli05a] has appeared, such results for routing schemes have been obtained by Abraham et al. [AGGM06].

Second, we would like extend our results on all four problems considered in Chapter 7 to decomposable metrics. This direction seems promising since similar extensions (from doubling metrics to decomposable metrics) have been obtained, in [KLMN05] and in Chapter 4 of this thesis, in the context of metric embeddings. Also, recent results of Abraham at al. [AGM05, AG06] construct low-stretch routing schemes, distance labeling schemes, and small-world networks for graphs excluding a fixed minor.

Bibliography

[ABC$^+$05] Ittai Abraham, Yair Bartal, Hubert T-H. Chan, Kedar Dhamdhere, Anupam Gupta, Jon Kleinberg, Ofer Neiman, and Aleksandrs Slivkins. Metric Embeddings with Relaxed Guarantees. In *46th Symp. on Foundations of Computer Science (FOCS)*, pages 83–100, 2005.

[ABN06] Ittai Abraham, Yair Bartal, and Ofer Neiman. Advances in Metric Embedding Theory. In *38th ACM Symp. on Theory of Computing (STOC)*, 2006.

[ABNLP90] Baruch Awerbuch, Amotz Bar-Noy, Nati Linial, and David Peleg. Improved routing strategies with succinct tables. *J. of Algorithms*, 11(3):307–341, 1990.

[AG06] Ittai Abraham and Cyril Gavoille. Object location using path separators. In *25th Annual ACM SIGACT-SIGOPS Symp. on Principles Of Distributed Computing (PODC)*, 2006.

[AGGM06] Ittai Abraham, Cyril Gavoille, Andrew Goldberg, and Dahlia Malkhi. Routing in networks with low doubling dimension. In *26th IEEE Intl. Conf. on Distributed Computing Systems (ICDCS)*, 2006.

[AGLP89] Baruch Awerbuch, Andrew Goldberg, Michael Luby, and Serge Plotkin. Network decomposition and locality in distributed computation. In *30th Symp. on Foundations of Computer Science (FOCS)*, pages 364–369, 1989.

[AGM04a] Ittai Abraham, Cyril Gavoille, and Dahlia Malkhi. Routing with improved communication-space tradeoff. In *18th Annual Conf. on Distributed Computing (DISC)*, pages 305–319, 2004.

[AGM$^+$04b] Ittai Abraham, Cyril Gavoille, Dahlia Malkhi, Noam Nisan, and Mikkel Thorup. Compact name-independent routing with minimum stretch. In *16th ACM Symp. on Parallel Algorithms and Architectures (SPAA)*, pages 20–24, 2004.

[AGM05] Ittai Abraham, Cyril Gavoille, and Dahlia Malkhi. Compact routing for graphs excluding a fixed minor. In *19th Annual Conf. on Distributed Computing (DISC)*, pages 442–456, 2005.

[AHK04] Sanjeev Arora, Elad Hazan, and Satyen Kale. $0(\sqrt{\log n})$ Approximation to SPARSEST CUT in $O(n^2)$ Time. In *45th Symp. on Foundations of Computer Science (FOCS)*, pages 238–247, 2004.

[AKPW95] Noga Alon, Richard M. Karp, David Peleg, and Douglas West. A graph-theoretic game and its application to the k-server problem. *SIAM J. Comput.*, 24(1):78–100, 1995.

[ALN05] Sanjeev Arora, James R. Lee, and Assaf Naor. Euclidean distortion and the sparsest cut. In *37th ACM Symp. on Theory of Computing (STOC)*, pages 553–562, 2005.

[ALV75] S. V. Konyagin A. L. Volberg. On measures with the doubling condition. *Izv. Acad. Nauk SSSR*, 51:1987, 666-675. In Russian; English transl. Math. USSR Izv. 30 (1988), 629638.

[AM04] Ittai Abraham and Dahlia Malkhi. Compact routing on Euclidean metrics. In *23rd Annual ACM SIGACT-SIGOPS Symp. on Principles Of Distributed Computing (PODC)*, pages 141–149, 2004.

[AM05] Ittai Abraham and Dahlia Malkhi. Name independent routing for growth bounded networks. In *17th ACM Symp. on Parallel Algorithms and Architectures (SPAA)*, pages 49–55, 2005.

[AMD04] Ittai Abraham, Dahlia Malkhi, and Oren Dobzinski. LAND: Stretch $(1+\epsilon)$ locality-aware networks for DHTs. In *15th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 550–559, 2004.

[AP90] Baruch Awerbuch and David Peleg. Sparse partitions. In *31st Symp. on Foundations of Computer Science (FOCS)*, pages 503–513, 1990.

[ARV04] Sanjeev Arora, Satish Rao, and Umesh Vazirani. Expander flows, geometric embeddings, and graph partitionings. In *36th ACM Symp. on Theory of Computing (STOC)*, pages 222–231, 2004.

[AS00] Noga Alon and Joel Spencer. *The Probabilistic Method*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons, New York, 2nd edition, 2000.

[Ass83] Patrice Assouad. Plongements lipschitziens dans $\mathbf{R}^n$. *Bull. Soc. Math. France*, 111(4):429–448, 1983.

[Bar96] Yair Bartal. Probabilistic approximations of metric spaces and its algorithmic applications. In *37th Symp. on Foundations of Computer Science (FOCS)*, pages 184–193, 1996.

[Bar98] Yair Bartal. On approximating arbitrary metrics by tree metrics. In *30th ACM Symp. on Theory of Computing (STOC)*, pages 161–168, 1998.

[BAS04] A. Bharambe, M. Agrawal, and S. Seshan. Mercury: Supporting Scalable Multi-Attribute Range Queries. In *ACM SIGCOMM*, 2004.

[BC03] Bo Brinkman and Moses S. Charikar. On the impossibility of dimension reduction in l1. In *44th Symp. on Foundations of Computer Science (FOCS)*, pages 514–523, 2003.

[BDH96] C. Barber, D. Dobkin, and H. Huhdanpaa. The Quickhull Algorithm for Convex Hulls. *Transactions on Mathematical Software*, 22(4):469–483, 1996.

[BLMN05] Yair Bartal, Nathan Linial, Manor Mendel, and Assaf Naor. On metric ramsey-type phenomena. *Annals of Mathematics*, 162(2):643–709, 2005. Preliminary version in *35th ACM STOC, 2003*.

[Bou85] Jean Bourgain. On Lipschitz embeddings of finite metric spaces in Hilbert space. *Israel J. of Mathematics*, 52(1-2):46–52, 1985.

[Bou86] Jean Bourgain. The metrical interpretation of superreflexivity in Banach spaces. *Israel J. of Mathematics*, 56(2):222–230, 1986.

[BS82] William B.Johnson and G. Schechtman. Embedding l_p^m into l_1^n. *Acta Mathematica*, 149:71–85, 1982.

[CCRK04] M. Costa, M. Castro, A. Rowstron, and P. Key. PIC: Practical Internet coordinates for distance estimation. In *24th IEEE Intl. Conf. on Distributed Computing Systems (ICDCS)*, 2004.

[CDG$^+$05] Hubert T-H. Chan, Kedar Dhamdhere, Anupam Gupta, Jon Kleinberg, and Aleksandrs Slivkins. Metric Embeddings with Relaxed Guarantees, 2005. This is a full version of [ABC$^+$05], currently under submission. Available from *http://www.cs.cornell.edu/people/slivkins/research*. The conference version [ABC$^+$05] combined independent efforts of two research groups, who decided to publish their full versions separately.

[CGMZ05] Hubert T-H. Chan, Anupam Gupta, Bruce M. Maggs, and Shuheng Zhou. On hierarchical routing in bounded-growth metrics. In *16th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 762–771, 2005. Full and updated version available as a Carnegie Mellon University ETR CMU-PDL-04-106.

[CH88] G.M. Crippen and T.F. Havel. *Distance geometry and molecular conformation*. Wiley, New York, NY, 1988.

[CLGS04] A. Crainiceanu, P. Linga, J. Gehrke, and J. Shanmugasundaram. Querying Peer-to-Peer Networks Using P-Trees. In *Intl. Workshop on the Web and Databases*, Paris, France, June 2004.

[DCKM04] F. Dabek, R. Cox, Frans Kaashoek, and Robert Morris. Vivaldi: A decentralized network coordinate system. In *ACM SIGCOMM*, 2004.

[DF98] Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity*. Springer-Verlag, New York, 1998.

[DGH$^+$87] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry. Epidemic Algorithms for Replicated Database Maintenance. In *6th Annual ACM SIGACT-SIGOPS Symp. on Principles Of Distributed Computing (PODC)*, pages 1–12, 1987.

[DHLS05] Philippe Duchon, Nicolas Hanusse, Emmanuelle Lebhar, and Nicolas Schabanel. Could any graph be turned into a small world? In *19th Annual Conf. on Distributed Computing (DISC)*, pages 511–513, 2005. Full version to appear in *Theoretical Computer Science*, special issue on complex networks, 2006.

[Die97] Reinhard Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer-Verlag, Heidelberg, 3rd edition, 2005 (2000, 1997). Hyperlinked electronic edition is available at *http://www.math.uni-hamburg.de/home/diestel/books/graph.theory/*.

[DL97] Michel Marie Deza and Monique Laurent. *Geometry of cuts and metrics*, volume 15 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, 1997.

[DMW03] Peter S. Dodds, Roby Muhamad, and Duncan J. Watts. An experimental study of search in global social networks. *Science*, 301:827–829, 2003.

[Dvo59] A. Dvoretzky. A theorem on convex bodies and applications to Banach spaces. *Proc. of the Nat. Acad. Sci. USA*, 45:223–226, 1959.

[FGP04] Pierre Fraigniaud, Cyril Gavoille, and Christophe Paul. Eclecticism shrinks even small worlds. In *23rd Annual ACM SIGACT-SIGOPS Symp. on Principles Of Distributed Computing (PODC)*, pages 169–178, 2004.

[FJJ$^+$01] P. Francis, S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang. IDMaps: A global Internet host distance estimation service. *IEEE/ACM Transactions on Networking*, 9:525–540, October 2001. Preliminary version in IEEE INFOCOM 1999.

[FkcHM01] M. Fomenkov, k. claffy, B. Huffaker, and D. Moore. Macroscopic Internet topology and performance measurements from the DNS root name servers. In *Usenix LISA*, 2001.

[FLL06] Pierre Fraigniaud, Emmanuelle Lebhar, and Zvi Lotker. A doubling dimension threshold $\Theta(\log\log n)$ for augmented graph navigability. In *14th Annual European Symp. on Algorithms (ESA)*, 2006.

[FLM77] T. Figiel, Joram Lindenstrauss, and V.D. Milman. The dimension of almost spherical sections of convex bodies. *Acta Mathematica*, 139:53–94, 1977.

[Fra05] Pierre Fraigniaud. A new perspective on the small-world phenomenon: greedy routing in tree-decomposed graphs. In *13th Annual European Symp. on Algorithms (ESA)*, pages 791–802, 2005.

[Fri03] Joel Friedman. A proof of Alon's second eigenvalue conjecture. In *35th ACM Symp. on Theory of Computing (STOC)*, pages 720–724, 2003. Full version available as *CoRR cs.DM/0405020*, 2004.

[FRT04] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *J. of Computer and System Sciences*, 69(3):485–497, 2004. Preliminary version in *35th ACM STOC*, 2003.

[FT03] Jittat Fakcharoenphol and Kunal Talwar. An improved decomposition theorem for graphs excluding a fixed minor. *RANDOM-APPROX*, pages 36–46, 2003.

[GG01] Cyril Gavoille and Marc Gengler. Space-efficiency for routing schemes of stretch factor three. *Journal of Parallel and Distributed Computing*, 61(5):679–687, 2001.

[GGR98] Oded Goldreich, Shafi Goldwasser, and Dana Ron:. Property Testing and its Connection to Learning and Approximation. *Journal of the ACM*, 45(4):653–750, 1998. Preliminary version in *37th IEEE FOCS*, 2006.

[GKK$^+$01] Cyril Gavoille, Michal Katz, Nir A. Katz, Cristophe Paul, and David Peleg. Approximate distance labeling schemes. In *9th Annual European Symp. on Algorithms (ESA)*, pages 476–487, 2001.

[GKL03] Anupam Gupta, Robert Krauthgamer, and James R. Lee. Bounded geometries, fractals, and low–distortion embeddings. In *44th Symp. on Foundations of Computer Science (FOCS)*, pages 534–543, 2003.

[GMS04] Christos Gkantsidis, M. Mihail, and Amin Saberi. Random Walks in Peer-to-Peer Networks. In *IEEE INFOCOM*, 2004.

[GNRS04] Anupam Gupta, Ilan Newman, Yuri Rabinovich, and Alistair Sinclair. Cuts, trees and ℓ_1-embeddings of graphs. *Combinatorica*, 24(2):233–269, 2004. (Preliminary version in 40th FOCS, 1999.).

[GP03] Cyril Gavoille and David Peleg. Compact and localized distributed data structures. *J. of Distributed Computing*, 16:111–120, 2003.

[GPPR04] Cyril Gavoille, David Peleg, Stephane Perennes, and Ran Raz. Distance labeling in graphs. *J. of Algorithms*, 53(1):85–112, 2004. (Preliminary version in *12th ACM-SIAM SODA*, 2001).

[GS95] J. Guyton and M. Schwartz. Locating nearby copies of replicated Internet servers. In *ACM SIGCOMM*, 1995.

[GSG02] Krishna P. Gummadi, Stefan Saroiu, and Steven D. Gribble. King: Estimating Latency between Arbitrary Internet End Hosts. In *2nd ACM SIGCOMM Internet Measurement Workshop (IMW)*, pages 5–18, 2002.

[Gup00] Anupam Gupta. Embedding tree metrics into low dimensional Euclidean spaces. *Discrete and Computational Geometry*, 24(1):105–116, 2000. Preliminary version in *31st ACM STOC*, 1999.

[Gup01] Anupam Gupta. Steiner points in tree metrics don't (really) help. In *12th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 220–227, Philadelphia, PA, 2001.

[Hei01] J. Heinonen. *Lectures on analysis on metric spaces*. Universitext. Springer-Verlag, New York, 2001.

[HFP$^+$02] B. Huffaker, M. Fomenkov, D.J. Plummer, D. Moore, and k claffy. Distance Metrics in the Internet. In *IEEE International Telecommunications Symposium (ITS)*, 2002.

[HKMR04] Kirsten Hildrum, John Kubiatowicz, Sean Ma, and Satish Rao. A note on the nearest neighbor in growth-restricted metrics. In *15th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 560–561, 2004.

[HKR04] Kirsten Hildrum, John Kubiatowicz, and Satish Rao. Object location in realistic networks. In *16th ACM Symp. on Parallel Algorithms and Architectures (SPAA)*, pages 25–35, 2004.

[IM04] Piotr Indyk and Jiri Matoušek. Low-distortion embeddings of finite metric spaces. In Jacob E. Goodman and Joseph O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, Discrete Mathematics and its Applications (Boca Raton), chapter 8, pages xviii+1539. Chapman & Hall/CRC, Boca Raton, FL, second edition, 2004.

[Ind99] Piotr Indyk. Sublinear time algorithms for metric space problems. In *31st ACM Symp. on Theory of Computing (STOC)*, pages 428–434, 1999.

[Ind01] Piotr Indyk. Algorithmic applications of low-distortion geometric embeddings. In *42nd Symp. on Foundations of Computer Science (FOCS)*, pages 10–33, 2001. Invited survey.

[JL84] William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz maps into a Hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

[KKD04] David Kempe, Jon Kleinberg, and Alan Demers. Spatial Gossip and Resource Location Protocols. *J. ACM*, 51(6):943–967, 2004. Preliminary version in *33rd ACM STOC*, 2001.

[KL03] Robi Krauthgamer and James R. Lee. The Intrinsic Dimensionality of Graphs. In *35th ACM Symp. on Theory of Computing (STOC)*, pages 438–447, 2003.

[KL04] Robi Krauthgamer and James R. Lee. Navigating nets: simple algorithms for proximity search. In *15th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 798–807, 2004.

[Kle00a] Jon Kleinberg. Navigation in a small world. *Nature*, 406:485, 2000.

[Kle00b] Jon Kleinberg. The small-world phenomenon: an algorithmic perspective. In *32nd ACM Symp. on Theory of Computing (STOC)*, pages 163–170, 2000.

[Kle01] Jon Kleinberg. Small-world phenomena and the dynamics of information. In *15th Advances in Neural Information Processing Systems (NIPS)*, pages 431–438, 2001.

[Kle06] Jon Kleinberg. Complex networks and decentralized search algorithms. In *International Congress of Mathematicians*, 2006.

[KLMN05] Robert Krauthgamer, James Lee, Manor Mendel, and Assaf Naor. Measured descent: A new embedding method for finite metrics. *Geometric and Functional Analysis (GAFA)*, 15(4):839–858, 2005. Preliminary version in *45th IEEE FOCS*, 2004.

[KPR93] Philip Klein, Serge A. Plotkin, and Satish B. Rao. Excluded minors, network decomposition, and multicommodity flow. In *25th ACM Symp. on Theory of Computing (STOC)*, pages 682–690, 1993.

[KR02] D.R. Karger and M. Ruhl. Finding Nearest Neighbors in Growth-restricted Metrics. In *34th ACM Symp. on Theory of Computing (STOC)*, pages 63–66, 2002.

[KS03] R. Krauthgamer and O. Sasson. Property testing of data dimensionality. In *14th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 18–27, 2003.

[KSB01] C. Kommareddy, N. Shankar, and B. Bhattacharjee. Finding close friends on the Internet. In *12th IEEE Intl. Conf. on Network Protocols (ICNP)*, 2001.

[KSW04] Jon Kleinberg, Aleksandrs Slivkins, and Tom Wexler. Triangulation and Embedding Using Small Sets of Beacons. In *45th Symp. on Foundations of Computer Science (FOCS)*, pages 444–453, 2004.

[Laa02] Tomi J. Laakso. Plane with A_∞-weighted metric not bi-Lipschitz embeddable to $\mathbb{R}^N$. *Bull. London Math. Soc.*, 34(6):667–676, 2002.

[Lee05] James R. Lee. On distance scales, embeddings, and efficient relaxations of the cut cone. In *16th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 92–101, 2005.

[LHC03] H. Lim, J. Hou, and C. Choi. Constructing Internet coordinate system based on delay measurement. In *ACM SIGCOMM Internet Measurement Conf. (IMC)*, 2003.

[Lin02] Nathan Linial. Finite metric spaces – combinatorics, geometry and algorithms. In *International Congress of Mathematicians, Vol. III (Beijing, 2002)*, pages 573–586, Beijing, 2002.

[LL04] L. Lehman and S. Lerman. Pcoord: Network position estimation using peer-to-peer measurements. In *3rd IEEE International Symposium on Network Computing and Applications (IEEE NSA)*, 2004.

[LLR95] Nathan Linial, Eran London, and Yuri Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995. (Preliminary version in *35th IEEE FOCS*, 1994).

[LMN04] James R. Lee, Manor Mendel, and Assaf Naor. Metric Structures in L1: Dimension, Snowflakes, and Average Distortion. In *Latin-American Conf. on Combinatorics, Graphs and Applications (LATIN)*, Buenos Aires, Argentina, 2004.

[LN04] James R. Lee and Assaf Naor. Embedding the diamond graph in Lp and dimension reduction in $L1$. *Geometric and Functional Analysis (GAFA)*, 4(4):745–747, 2004.

[LP01] Urs Lang and Conrad Plaut. Bilipschitz Embeddings of Metric Spaces Into Space Forms. *Geom. Dedicata*, 87(1-3):285–307, 2001.

[LS98] Jouni Luukkainen and Eero Saksman. Every complete doubling metric space carries a doubling measure. *Proc. Amer. Math. Soc.*, 126(2):531534, 1998.

[LS04] Emmanuelle Lebhar and Nicolas Schabanel. Close to optimal decentralized routing in long-range contact networks. In *31st Intl. Colloquium on Automata, Languages and Programming (ICALP)*, pages 894–905, 2004.

[LW02] Nati Linial and Avi Wigderson. Course notes: Expander graphs and their applications. http://www.math.ias.edu/ boaz/ExpanderCourse/, 2002.

[Mat90] Jiří Matoušek. Bi-Lipschitz embeddings into low dimensional Euclidean spaces. *Commentationes Mathematicae Universitatis Carolinae*, 31(3):589–600, 1990.

[Mat97] Jiří Matoušek. On embedding expanders into l_p spaces. *Israel J. Math.*, 102:189–197, 1997.

[Mat99] Jiří Matoušek. On embedding trees into uniformly convex Banach spaces. *Israel J. of Mathematics*, 114:221–237, 1999. (Czech version in : *Lipschitz distance of metric spaces*, C.Sc. degree thesis, Charles University, 1990).

[Mat02a] Jiří Matoušek. *Lectures on discrete geometry*, volume 212 of *Graduate Texts in Mathematics*, chapter 15 (Embedding Finite Metric Spaces into Normed Spaces), pages 323–362. Springer-Verlag, New York, 2002.

[Mat02b] Jiří Matoušek. *Lectures on discrete geometry*, volume 212 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 2002.

[MBR03] Gurmeet Manku, Mayank Bawa, and Prabhakar Raghavan. Symphony: Distributed hashing in a small world. In *4th USENIX Symp. on Internet Technologies and Systems (USITS)*, 2003.

[MHP05] Manor Mendel and Sariel Har-Peled. Fast construction of nets in low dimensional metrics, and their applications. In *21st ACM Symp. on Computational Geometry (SoCG)*, pages 150–158, 2005.

[Mil67] Stanley Milgram. The small world problem. *Psychology Today*, 2:60–67, 1967.

[MN04] Chip Martel and Van Nguyen. Analyzing Kleinberg's (and other) small-world models. In *23rd Annual ACM SIGACT-SIGOPS Symp. on Principles Of Distributed Computing (PODC)*, pages 179–188, 2004.

[MN05] Chip Martel and Van Nguyen. Analyzing and characterizing small-world graphs. In *16th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 311–320, 2005.

[MNW04] Gurmeet Manku, Moni Naor, and Udi Wieder. Know thy neighbor's neighbor: the power of lookahead in randomized P2P networks. In *36th ACM Symp. on Theory of Computing (STOC)*, pages 54–63, 2004.

[MR95] Rajeev Motwani and Prabhakar Raghavan. *Randomized algorithms*. Cambridge University Press, Cambridge, 1995.

[NZ02] T.S. Eugene Ng and Hui Zhang. Predicting Internet network distance with coordinates-based approaches. In *IEEE INFOCOM*, 2002.

[NZ04] T. Ng and H. Zhang. A network positioning system for the Internet. In *USENIX*, 2004.

[PCW$^+$03] M. Pias, J. Crowcroft, S. Wilbur, T. Harris, and S. Bhatti. Lighthouses for scalable distributed location. In *2nd Intl. Workshop on Peer-to-Peer Systems (IPTPS)*, 2003.

[Pel99] David Peleg. Proximity-preserving labeling schemes and their applications. In *The 25th Intl. Workshop on Graph-Theoretical Concepts in Computer Science (LNCS 1665)*, pages 30–41, 1999.

[Pel00] David Peleg. *Distributed computing: a locality-sensitive approach*. SIAM, 2000.

[Pit87] Boris Pittel. On spreading a rumor. *SIAM J. Appl. Math.*, 47(1):213–223, 1987.

[PR03] Michal Parnas and Dana Ron. Testing metric properties. *Information and Computation*, 187(2):155–195, 2003. Preliminary version in *33th ACM STOC*, 2001.

[PRR99] C. Greg Plaxton, Rajmohan Rajaraman, and Andrea W. Richa. Accessing nearby copies of replicated objects in a distributed environment. *Theory Comput. Syst.*, 32(3):241–280, 1999. Preliminary version in *9th ACM SPAA*, 1997.

[PU89] David Peleg and Eli Upfal. A trade-off between space and effciency for routing tables. *J. of the ACM*, 36(3):510–530, 1989. (Preliminary version in *20th ACM STOC*, 1988).

[PV03] R. Percacci and A. Vespignani. Scale-free behavior of the Internet global performance. *The European Physical Journal B*, 32(4):411–414, 2003. Also appeared as *arXiv e-print cond-mat/0209619*, September 2002.

[Rao99] Satish B. Rao. Small distortion and volume preserving embeddings for planar and Euclidean metrics. In *15th ACM Symp. on Computational Geometry (SoCG)*, pages 300–306, 1999.

[RD01] A. Rowstron and P. Druschel. Pastry: Scalable, Distributed Object Location and Routing for Large-Scale Peer-to-Peer Systems. In *3rd IFIP/ACM Intl. Conf. on Distributed Systems Platforms (Middleware)*, pages 329–350, 2001.

[RFH$^+$01] S. Ratnasamy, P. Francis, M. Hadley, R. Karp, and S. Shenker. A Scalable Content-Addressable Network. In *ACM SIGCOMM*, 2001.

[RR98] Yuri Rabinovich and Ran Raz. Lower bounds on the distortion of embedding finite metric spaces in graphs. *Discrete Comput. Geom.*, 19(1):79–94, 1998.

[Sem96] Stephen Semmes. On the nonexistence of bi-Lipschitz parameterizations and geometric problems about A1-weights. *Rev. Mat. Iberoamericana*, 12(2):337–410, 1996.

[SJ89] Alistair Sinclair and Mark Jerrum. Approximate Counting, Uniform Generation, and Rapidly Mixing Markov Chains. *Information and Computation*, 82:93–133, 1989.

[Sli05a] Aleksandrs Slivkins. Distance estimation and object location via rings of neighbors. In *24th Annual ACM SIGACT-SIGOPS Symp. on Principles Of Distributed Computing (PODC)*, pages 41–50, July 2005. Accepted to the special issue of *Distributed Computing*, currently under revision.

[Sli05b] Aleksandrs Slivkins. Distributed Approaches to Triangulation and Embedding. In *16th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 640–649, January 2005.

[Sli06] Aleksandrs Slivkins. Network Distance Estimation with Guarantees for All Node Pairs. Technical Report cul.cis/TR2006-2035, Cornell University Computing and Information Science, July 2006. Available online at *http://techreports.library.cornell.edu:8081/Dienst/UI/1.0/Display/cul.cis/TR2006-2035*.

[SMK$^+$01] Ion Stoica, Robert Morris, David Karger, Frans Kaashoek, and H. Balakrishnan. Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. In *ACM SIGCOMM*, 2001.

[ST03] Y. Shavitt and T. Tankel. Big-bang simulation for embedding network distances in euclidean space. In *IEEE INFOCOM*, 2003.

[Tal04] Kunal Talwar. Bypassing the embedding: approximation schemes and compact representations for growth restricted metrics. In *36th ACM Symp. on Theory of Computing (STOC)*, pages 281–290, 2004.

[TC03] L. Tang and M. Crovella. Virtual landmarks for the Internet. In *ACM SIGCOMM Internet Measurement Conf. (IMC)*, 2003.

[TZ01] Mikkel Thorup and Uri Zwick. Compact routing schemes. In *13th ACM Symp. on Parallel Algorithms and Architectures (SPAA)*, pages 1–10, 2001.

[TZ05] Mikkel Thorup and Uri Zwick. Approximate distance oracles. *J. of the ACM*, 52(1):1–24, 2005. (Preliminary version in *33rd ACM STOC*, 2001).

[VPSV02] A. Vazquez, R. Pastor-Satorras, and A. Vespignani. Large-scale topological and dynamical properties of Internet. *Phys. Rev. E*, 65, 066130, 2002.

[WS98] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–42, 1998.

[WSS05] Bernard Wong, Aleksandrs Slivkins, and Emin Gun Sirer. Meridian: A Lightweight Network Location Service without Virtual Coordinates. In *ACM SIGCOMM*, 2005.

[Wu98] Jang-Mei Wu. Hausdorff dimension and doubling measures on metric spaces. *Proc. Amer. Math. Soc.*, 126(5):14531459, 1998.

[Xia03] David Y. Xiao. The Evolution of Expander Graphs. BA Thesis, Harvard University, 2003. Available from *http://www.cs.princeton.edu/ dxiao*.

[ZHR$^+$04] B.Y. Zhao, L. Huang, S.C. Rhea, J. Stribling, A.D. Joseph, and J.D. Kubiatowicz. Tapestry: a resilient global-scale overlay for service deployment. *IEEE Journal on Selected Areas in Communications*, 22(1):41–53, 2004.