Algorithmica (2012) 62:713–732
DOI 10.1007/s00453-010-9478-x

Additive Spanners and Distance and Routing Labeling Schemes for Hyperbolic Graphs

Victor Chepoi · Feodor F. Dragan · Bertrand Estellon · Michel Habib · Yann Vaxès · Yang Xiang

Received: 28 May 2008 / Accepted: 25 November 2010 / Published online: 24 December 2010

Abstract δ-Hyperbolic metric spaces have been defined by M. Gromov in 1987 via a simple 4-point condition: for any four points u, v, w, x, the two larger of the distance sums $d(u, v) + d(w, x), d(u, w) + d(v, x), d(u, x) + d(v, w)$ differ by at

The results in this paper have not appeared before in conference proceedings, with the exception of distance approximating trees (Proposition 2, Proposition 4), which appeared in a preliminary version in proceedings of SoCG 2008 [17].

V. Chepoi, F.F. Dragan, B. Estellon and Y. Vaxés were supported in part by the ANR grant BLAN06-1-138894 (project OPTICOMB).
Y. Xiang was supported in part by the NSF under Grant #1019343 to the CRA for the CIFellows Project.

V. Chepoi · B. Estellon · Y. Vaxès
Laboratoire d'Informatique Fondamentale, Faculté des Sciences de Luminy, Université de la Mediterranée, Marseille, 13288 Marseille Cedex 9, France

V. Chepoi
e-mail: chepoi@lif.univ-mrs.fr

B. Estellon
e-mail: estellon@lif.univ-mrs.fr

Y. Vaxès
e-mail: vaxes@lif.univ-mrs.fr

F.F. Dragan (✉)
Algorithmic Research Laboratory, Department of Computer Science, Kent State University, Kent, OH 44242, USA
e-mail: dragan@cs.kent.edu

M. Habib
LIAFA, University Paris Diderot - Paris 7, Case 7014, 75205 Paris Cedex 13, France
e-mail: habib@liafa.jussieu.fr

Y. Xiang
Department of Biomedical Informatics, The Ohio State University, Columbus, OH 43210, USA
e-mail: yxiang@bmi.osu.edu

most 2δ. They play an important role in geometric group theory, geometry of negatively curved spaces, and have recently become of interest in several domains of computer science, including algorithms and networking. In this paper, we study unweighted δ-hyperbolic graphs. Using the Layering Partition technique, we show that every n-vertex δ-hyperbolic graph with $\delta \geq 1/2$ has an additive $O(\delta \log n)$-spanner with at most $O(\delta n)$ edges and provide a simpler, in our opinion, and faster construction of distance approximating trees of δ-hyperbolic graphs with an additive error $O(\delta \log n)$. The construction of our tree takes only linear time in the size of the input graph. As a consequence, we show that the family of n-vertex δ-hyperbolic graphs with $\delta \geq 1/2$ admits a routing labeling scheme with $O(\delta \log^2 n)$ bit labels, $O(\delta \log n)$ additive stretch and $O(\log_2(4\delta))$ time routing protocol, and a distance labeling scheme with $O(\log^2 n)$ bit labels, $O(\delta \log n)$ additive error and constant time distance decoder.

Keywords Algorithms · Distance and routing labeling schemes · Additive spanners · δ-Hyperbolic graphs

1 Introduction

This paper investigates whether δ-hyperbolic graphs admit good spanners and compact and efficient routing and distance labeling schemes. Commonly, when we make a query concerning a pair of vertices in a graph (adjacency, distance, shortest route, etc.), we need to make an access to a global data structure storing the information. A compromise to this approach is to store enough information locally in a label associated with a vertex such that the query can be answered using only the information in the labels of two vertices in question and nothing else. Motivation of localized data structure in distributed computing is surveyed and widely discussed in [33, 51].

We are mainly interested here in the distance and routing labeling schemes, introduced by Peleg (see, e.g., [51]). Informally, *distance labeling schemes* (DLS, for short) are schemes that label the vertices of a graph with short labels in such a way that the distance between any two vertices u and v can be determined or estimated efficiently by merely inspecting the labels of u and v, without using any other information. A *routing scheme* is a mechanism that can deliver packets of information from any vertex of the network to any other vertex. *Routing labeling schemes* (RLS, for short) are schemes that label the vertices of a graph with short labels in such a way that given the label of a current vertex and the label of a destination, it is possible to compute efficiently the port number of the edge from the current vertex that heads in the direction of the destination. Routing is one of the basic tasks that a distributed network of processors must be able to perform.

More formally, a graph family $\mathcal{D}$ is said to have an $l(n)$ *bit* (s,r)*-approximate distance labeling scheme* if there is a function L labeling the vertices of each n-vertex graph in $\mathcal{D}$ with distinct labels of up to $l(n)$ bits, and there exists an algorithm/function f, called *distance decoder*, that given two labels $L(v), L(u)$ of two vertices v, u in a graph G from $\mathcal{D}$, computes, in time polynomial in the length of the given labels, a value $f(L(v), L(u))$ such that $d_G(v,u) \leq f(L(v), L(u)) \leq$

$s \cdot d_G(v, u) + r$ (where $d_G(v, u)$ is the length of a shortest path between u and v in G). Note that the algorithm is not given any additional information, other that the two labels, regarding the graph from which the vertices were taken. Similarly, a family $\Re$ of graphs is said to have an *$l(n)$ bit (s, r)-approximate routing labeling scheme* if there exist a function L, labeling the vertices of each n-vertex graph in $\Re$ with distinct labels of up to $l(n)$ bits, and an efficient algorithm/function f, called the *routing decision* or *routing protocol*, that given the label of a current vertex v and the label of the destination vertex (the header of the packet), decides in time polynomial in the length of the given labels and using only those two labels, whether this packet has already reached its destination, and if not, to which neighbor of v to forward the packet. Furthermore, the routing path from any source s to any destination t produced by this scheme in a graph G from $\Re$ must have the length at most $s \cdot d_G(s, t) + r$. For simplicity, $(1, r)$-approximate labeling schemes (distance or routing) are called *r-additive labeling schemes* (with *additive stretch r*), and $(s, 0)$-approximate labeling schemes are called *s-multiplicative labeling schemes* (with *multiplicative stretch factor s*). The distance and routing labeling schemes, we propose for δ-hyperbolic graphs, are additive in nature.

Introduced by Gromov [38], δ-hyperbolicity measures, to some extent, the deviation of a metric from a tree metric. Recall that a metric space (X, d) embeds into a tree network (with positive real edge lengths), that is, d is a *tree metric,* iff for any four points u, v, w, x, the two larger of the distance sums $d(u, v) + d(w, x), d(u, w) + d(v, x), d(u, x) + d(v, w)$ are equal. A metric space (X, d) is called *δ-hyperbolic* if the two largest distance sums differ by at most 2δ. A connected graph $G = (V, E)$ is *δ-hyperbolic* (or of *hyperbolicity* δ) if the metric space (V, d_G) is δ-hyperbolic, where d_G is the standard shortest path metric defined on G. 0-Hyperbolic metric spaces are exactly the tree metrics (therefore, in what follows, we will be interested in more general δ-hyperbolic metric spaces where $\delta > 0$).[1] On the other hand, the Poincaré half space in $\mathbf{R}^k$ with the hyperbolic metric is δ-hyperbolic with $\delta = \log_2 3$. Several classes of geodesic metric spaces are known to be hyperbolic [6, 42] (a metric space (X, d) is called *hyperbolic* if it is δ-hyperbolic for some constant δ).

A spanning subgraph H of a graph $G = (V, E)$ is called a *(s, r)-spanner* of G if $d_H(u, v) \leq s \cdot d_G(u, v) + r$ holds for any $u, v \in V$. A $(s, 0)$-spanner is usually called a multiplicative s-spanner, a $(1, r)$-spanner is usually called an additive r-spanner.

Our Results Using the Layering Partition technique (developed in [10, 13] for chordal and k-chordal graphs), we present two new networking primitives for unweighted δ-hyperbolic graphs with $\delta \geq 1/2$. We show that

- every n-vertex δ-hyperbolic graph has an additive $O(\delta \log n)$-spanner with at most $O(\delta n)$ edges and
- the family of δ-hyperbolic graphs with n vertices admits an $O(\delta \log n)$-additive routing labeling scheme with $O(\delta \log^2 n)$ bit labels and $O(\log_2(4\delta))$ time routing protocol.

[1]For unweighted graphs different from trees, by definition, 2δ is a positive integer, and therefore $\delta \in \{\frac{1}{2}, 1, \frac{3}{2}, 2, \frac{5}{2}, 3, \ldots\}$ (i.e., δ is an integer or a half-integer).

Table 1 Our results on routing labeling schemes and spanners of δ-hyperbolic graphs along with related known results (see Sect. 1.2 and Sect. 6). The sign '+' indicates that the stretch is additive, '×' indicates that the stretch is multiplicative

	Routing Labeling Scheme			Sparse Spanner		
graphs	stretch	label size	ref.	stretch	# of edges	ref.
trees	$+0$	$O(\log n)$	[27, 58]	–	–	–
k-chordal ($k \geq 3$)	$+2\lfloor\frac{k}{2}\rfloor$	$O(\frac{\log^3 n}{\log\log n})$	[23, 25]	$+(k+1)$	$O(n)$	[15]
	$+(k+1)$	$O(\log^2 n)$	[20]			
tree-length λ ($\lambda > 0$)	$+(6\lambda-2)$	$O(\lambda\log^2 n)$	[20]	$+4\lambda$	$O(\lambda n)$	[22]
δ-hyperbolic ($\delta \geq 1/2$)	$+O(\delta\log n)$	$O(\delta\log^2 n)$	here	$+O(\delta\log n)$	$O(\delta n)$	here
general	$\times(4k-5), \forall k \geq 2$	$\tilde{O}(kn^{1/k})$	[58]	$\times(2k-1), \forall k \geq 1$	$O(n^{1+1/k})$	[59]

To the best of our knowledge, for δ-hyperbolic graphs, these networking primitives are constructed for the first time. In Table 1, we put our results in the context of related, previously known results. The class of δ-hyperbolic graphs generalizes the class of k-chordal graphs and the class of tree-length λ graphs.

Additionally, we provide a faster and, in our opinion, simpler to understand construction of distance approximating trees of δ-hyperbolic graphs on n vertices with an additive error $O(\delta\log n)$. As a consequence, we obtain also a faster and easier to construct $O(\delta\log n)$-additive distance labeling scheme with $O(\log^2 n)$ bit labels and constant time distance decoder. Note that these results are comparable with results known in literature (see [37, 38] for distance approximating trees and [30] for approximate distance labeling schemes) in the additive error incurred and in the label size, however the construction of our approximating tree, and therefore of distance labeling scheme, is faster and simpler. The construction of our approximating tree for a connected graph $G=(V,E)$ can be done in linear $O(|E|)$ time while the construction of [37, 38] needs $O(|V|^2)$ time. Using our approximating tree, a distance labeling scheme for G can be constructed in $O(|E|+|V|\log|V|)$ time, while the construction of [30], based on a tree from [37, 38], needs $O(|V|^2)$ time.

Bibliographic Note Proposition 2 and the result on distance approximating trees (see Sect. 4, Proposition 4) were announced in SoCG'2008 paper [17] and the current paper is their journal version. All other results are new and have not appeared before.

1.1 δ-Hyperbolicity

δ-Hyperbolic metric spaces play an important role in geometric group theory and in geometry of negatively curved spaces [4, 37, 38]. δ-Hyperbolicity captures the basic common features of "negatively curved" spaces like the classical real-hyperbolic space $\mathbf{H}^k$, Riemannian manifolds of strictly negative sectional curvature, and of discrete spaces like trees and the Caley graphs of word-hyperbolic groups. It is remarkable that a strikingly simple concept leads to such a rich general theory [4, 37, 38].

More recently, the concept of δ-hyperbolicity emerged in discrete mathematics, algorithms, and networking. For example, it has been shown empirically in [53] (see also [3]) that the internet topology embeds with better accuracy into a hyperbolic space than into an Euclidean space of comparable dimension. A few algorithmic problems in hyperbolic spaces and hyperbolic graphs have been considered in recent papers [3, 14, 17, 30, 44, 50]. Kleinberg showed [44] that every connected finite graph has an embedding in the hyperbolic plane so that the greedy routing based on the virtual coordinates obtained from this embedding is guaranteed to work. Krauthgamer and Lee [50] presented a PTAS for the Traveling Salesman Problem when the set of cities lie in $\mathbf{H}^k$. They also show how to preprocess a finite subset of a δ-hyperbolic geodesic space with a uniformly bounded local geometry to efficiently answer nearest-neighbor queries with an additive error $O(\delta)$. Chepoi and Estellon [14] established a relationship between the minimum number of balls of radius $R + 2\delta$ covering a finite subset S of a δ-hyperbolic geodesic space and the size of the maximum R-packing of S and showed how to compute such coverings and packings in polynomial time. Chepoi et al. [17] gave efficient algorithms for fast and accurate estimations of diameters and radii of δ-hyperbolic geodesic spaces and graphs.

The class of δ-hyperbolic graphs generalizes the class of k-chordal graphs and the class of tree-length λ graphs, as bounded chordality implies bounded tree-length (each k-chordal graph has the tree-length at most $\lfloor \frac{k}{2} \rfloor$ [21]) and bounded tree-length implies bounded hyperbolicity (each tree-length λ graph has hyperbolicity at most λ [17]). Recall that a graph is *k-chordal* (or of *chordality k*) if its induced cycles are of length at most k, and it is of *tree-length* λ if it has a Robertson-Seymour tree-decomposition into bags of diameter at most λ [21]. Notice also that the graphs of bounded hyperbolicity and the graphs of bounded tree-width (i.e., graphs admitting a Robertson-Seymour tree-decomposition into bags of bounded size) are incomparable: a complete graph K_n on n vertices has tree-width $n-1$ and hyperbolicity 0, while an induced cycle C_n on $n = 4k$ vertices has tree-width 2 and hyperbolicity $k = n/4$.

In [8, 17, 49], some more classes of graphs with small hyperbolicity were investigated. For chordal graphs as well as dually chordal graphs and strongly chordal graphs one can construct trees approximating the graph-distances within an additive error 2 or 3 [10], from which it follows that those graphs have low δ-hyperbolicity (this result has been extended in [13] to all k-chordal graphs, also implying that k-chordal graphs are $\lfloor \frac{k}{2} \rfloor$-hyperbolic). In general, the distance in a δ-hyperbolic space on n points can be approximated within an additive error of $2\delta \log_2 n$ by a weighted tree metric [37, 38] and this approximation is tight. For n-vertex δ-hyperbolic graphs G, we describe here an alternative (linear time) construction of a tree approximating the distances of G with an additive error of $O(\delta \log_2 n)$ (see Sect. 4).

1.2 Related Work on Distance and Routing Labeling Schemes

Distance Labeling The main results in this area are that general graphs support an (exact) distance labeling scheme with labels of $O(n)$ bits [36], and that trees [5, 52], bounded tree-width graphs [36], distance-hereditary graphs [31], bounded clique-width graphs [18], non-positively curved plane graphs [16], all support distance labeling schemes with $O(\log^2 n)$ bit labels. The $O(n)$ bit upper bound is tight for general graphs, and a lower bound of $\Omega(\log^2 n)$ bit on the label length is known for trees

[36], implying that all the results mentioned above are tight as well, since all those graph families contain trees. Later, [9, 32] showed an optimal bound of $O(\log n)$ bits for interval graphs, permutation graphs, and their generalizations.

Other results concern approximate distance labeling schemes. For arbitrary graphs, the best scheme to date is due to Thorup and Zwick [59]. They proposed a $(2k-1)$-multiplicative DLS, for each integer $k \geq 1$, with labels of $O(n^{1/k}\log^2 n)$ bits. Moreover, $\Omega(n^{1/k})$ bit labels are required in the worst-case for every s-multiplicative DLS with $s < 2k+1$, for $k = 1, 2, 3, 5$, and with $s < 4k/3+2$, for all other values of k. In [35], it is proved that trees (and bounded tree-width graphs as well) dadmit a $(1+1/\log n)$-multiplicative DLS with labels of $O(\log n \Delta \log\log n)$ bits, and this is tight in terms of label length and approximation. They also design some $O(1)$-additive DLS with $O(\log^2 n)$ bit labels for several families of graphs, including the graphs with bounded longest induced cycle, and, more generally, the graphs of bounded tree-length. Interestingly, it is easy to show that every exact DLS for these families of graphs needs labels of $\Omega(n)$ bits in the worst-case [35]. Recently, metrics with doubling dimension α have been considered, i.e., metrics for which, for every r, each ball of radius $2r$ can be covered by at most 2^α balls of radius r. It generalizes Euclidean metrics and bounded growth graphs, and includes many realistic networks. After several successive improvements [39, 41, 54, 56], the best scheme for them to date, due to Slivkins [55], is a $(1+\epsilon)$-multiplicative DLS with $O(\epsilon^{-O(\alpha)}\log n \Delta \log\log n)$ bit labels. This is optimal for bounded α, by combining the results of [41] and the lower bound of [35] for trees. Note also that weighted planar graphs admit a $(1+\epsilon)$-multiplicative DLS with labels of $O(\epsilon^{-1}\log^3 n)$ bits (see [40, 57]). This has been generalized in [1] to graphs excluding a fixed minor with the same stretch and space bounds.

The existence of a $O(\delta \log n)$-additive distance labeling scheme with $O(\log^2 n)$ bit labels for n-vertex δ-hyperbolic graphs was already indicated in [30]. Its construction uses a distance labeling scheme for trees and a Gromov's result that the distances in a δ-hyperbolic space can be approximated by the weighted tree distances (see Theorem 1). The additive error incurred by our result is slightly weaker (but of the same order), however the construction of our distance approximating tree, and therefore of our distance labeling scheme, is simpler (our tree can be constructed in linear $O(|E|)$ time while the construction in [30] needs $O(|V|^2)$ time). Note also that our distance approximating tree has n vertices while that of [30] may have about $O(n^2)$ vertices. Paper [30] contains also a lower bound result which says that the label length $O(\log^2 n)$ is optimal up to some constant for every additive error up to n^ϵ.

Routing Labeling For general graphs there is an evident shortest path (i.e., with $s = 1$ and $r = 0$) RLS with labels of $O(n \log d)$ bits (so-called, *full tables*; here d is the maximum degree of a vertex) and this upper bound on the label size is tight (see [34]). A better routing scheme is known for trees. In [27, 58], a shortest path RLS for trees of arbitrary degree and diameter is described that assigns each vertex of an n-vertex tree a $(1+o(1))\log_2 n$ bit label. A shortest path routing labeling schemes with $O(\log^2 n)$ bit labels are known for bounded tree-width graphs [23, 36] and non-positively curved plane graphs [16].

To obtain routing schemes for general graphs that use $o(n)$ bit label for each vertex, one has to abandon the requirement that packets are always routed on shortest paths, and settle instead for the requirement that packets are routed on paths which are close to optimal [19, 26, 58]. A 3-multiplicative RLS that uses labels of size $\tilde{O}(n^{2/3})$ was obtained in [19],[2] and a 5-multiplicative RLS that uses labels of size $\tilde{O}(n^{1/2})$ was obtained in [26]. Later, authors of [58] further improved these results. They presented a $(4k-5)$-multiplicative RLS with only $\tilde{O}(kn^{1/k})$ bit labels, for every $k \geq 2$. Note that, each routing decision takes constant time in their scheme, and the label size is optimal, up to a logarithmic factor (see [28, 34]). For planar graphs, a shortest path RLS which uses $8n + o(n)$ bits per vertex is developed in [29], and a $(1+\epsilon)$-multiplicative RLS for every $\epsilon > 0$ which uses $O(\epsilon^{-1}\log^3 n)$ bits per vertex is developed in [57]. This has been generalized in [1] to graphs excluding a fixed minor with the same stretch and space bounds. Routing in metric spaces (including complete weighted graphs) with doubling dimension α has been considered in [2, 12, 45–48, 55, 56]. It was shown that any graph with doubling dimension α admits a $(1+\epsilon)$-multiplicative routing labeling scheme with labels of size $\epsilon^{-O(\alpha)}\log^2 n$ bits.

Recently, the routing result for trees of Thorup and Zwick [58] was used in designing $O(1)$-additive routing labeling schemes with $O(\log^{O(1)} n)$ bit labels for several families of graphs, including chordal graphs, chordal bipartite graphs, circular-arc graphs, AT-free graphs and their generalizations, the graphs with bounded longest induced cycle, the graphs of bounded tree-length, the bounded clique-width graphs, etc. (see [20, 23–25] and papers cited therein).

In this paper, using the Layering Partition technique developed in [10, 13] and new properties of δ-hyperbolic graphs established here, we generalize the routing labeling scheme of [20], developed for tree-length λ graphs, to all δ-hyperbolic graphs. Note that any tree-length λ graph is λ-hyperbolic [17] but that the converse is not true.

2 Preliminaries: Geodesic δ-Hyperbolic Spaces

Let (X, d) be a metric space. A (closed) *ball* $B(c, r)$ of radius r centered at $c \in X$ consists of all points $x \in X$ at distance at most r to c, i.e., $B(c, r) = \{x \in X : d(c, x) \leq r\}$. A *geodesic segment* joining two points x and y from X is a map ρ from the segment $[a, b]$ of length $|a-b| = d(x, y)$ to X such that $\rho(a) = x$, $\rho(b) = y$, and $d(\rho(s), \rho(t)) = |s-t|$ for all $s, t \in [a, b]$. A metric space (X, d) is *geodesic* if every pair of points in X can be joined by a geodesic. Every unweighted graph $G = (V, E)$ equipped with its standard distance d_G can be transformed into a geodesic (network-like) space (X, d) by replacing every edge $e = (u, v)$ by a segment $[u, v]$ of length 1; the segments may intersect only at common ends. Then (V, d_G) is isometrically embedded in a natural way in (X, d).

In case of geodesic metric spaces, there exist several equivalent definitions of δ-hyperbolicity involving different but comparable values of δ [4, 11, 37, 38]. A *geodesic triangle* $\Delta(x, y, z)$ with vertices $x, y, z \in X$ is union $[x, y] \cup [x, z] \cup [y, z]$ of three geodesic segments connecting these vertices. Let m_x be the point of the geodesic segment $[y, z]$ located at distance $\alpha_y := (d(y, x) + d(y, z) - d(x, z))/2$ from y. Then m_x is located at distance $\alpha_z := (d(z, y) + d(z, x) - d(y, x))/2$ from z

[2]Here, $\tilde{O}(f)$ means $O(f \text{ polylog}(n))$.

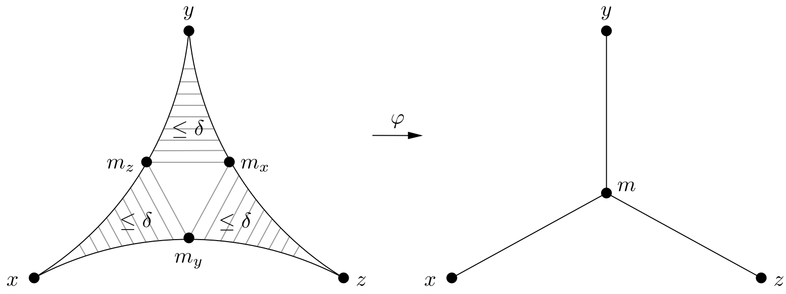

Fig. 1 A geodesic triangle $\Delta(x, y, z)$, the points m_x, m_y, m_z, and the tripod $\Upsilon(x, y, z)$

because $\alpha_y + \alpha_z = d(y, z)$. Analogously, define the points $m_y \in [x, z]$ and $m_z \in [x, y]$ both located at distance $\alpha_x := (d(x, y) + d(x, z) - d(y, z))/2$ from x; see Fig. 1 for an illustration. There exists a unique isometry φ which maps $\Delta(x, y, z)$ to a star $\Upsilon(x, y, z)$ consisting of three solid segments $[x, m]$, $[y, m]$, and $[z, m]$ of lengths α_x, α_y, and α_z, respectively. This isometry maps the vertices x, y, z of $\Delta(x, y, z)$ to the respective leaves x, y, z of $\Upsilon(x, y, z)$ and the points m_x, m_y, and m_z to the center m of this tripod. Any other point of $\Upsilon(x, y, z)$ is the image of exactly two points of $\Delta(x, y, z)$. A geodesic triangle $\Delta(x, y, z)$ is called δ-*thin* if for all points $u, v \in \Delta(x, y, z)$, $\varphi(u) = \varphi(v)$ implies $d(u, v) \leq \delta$. A geodesic triangle $\Delta(x, y, z)$ is called δ-*slim* if for any point u on the side $[x, y]$ the distance from u to $[x, z] \cup [z, y]$ is at most δ. The notions of geodesic triangles, δ-slim and δ-thin triangles can also be defined in case of graphs. The single difference is that for graphs, the center of the tripod is not necessarily the image of any vertex on the geodesic of $\Delta(x, y, z)$. Nevertheless, if a point of the tripod is the image of a vertex of one side of $\Delta(x, y, z)$, then it is also the image of another vertex located on another side of $\Delta(x, y, z)$. The following result shows that hyperbolicity of a geodesic space is equivalent to having thin or slim geodesic triangles (the same result holds for graphs).

Proposition 1 [4, 11, 37, 38] *Geodesic triangles of geodesic δ-hyperbolic spaces and δ-hyperbolic graphs are 4δ-slim and 4δ-thin. Conversely, geodesic spaces and graphs with δ-thin triangles are 2δ-hyperbolic and geodesic spaces and graphs with δ-slim triangles are 8δ-hyperbolic.*

Gromov [37, 38] established that any δ-hyperbolic metric on n points can be approximated in $O(n^2)$ time by a tree-metric with an additive error $O(\delta \log n)$:

Theorem 1 [37, 38] *For a δ-hyperbolic space (X, d) on n points with a root-point s there exists a weighted tree T and a mapping $\varphi : X \mapsto T$ such that $d_T(\varphi(s), \varphi(x)) = d(s, x)$ for any $x \in X$ and $d(x, y) - 2\delta \log_2 n \leq d_T(\varphi(x), \varphi(y)) \leq d(x, y)$ for any $x, y \in X$. The tree T can be constructed in $O(n^2)$ time.*

We conclude this section with a property of δ-hyperbolic graphs formulated and proven in several texts on Gromov hyperbolic spaces (in particular, in [4, 11]) for all δ-hyperbolic spaces. This result is used in the proof of the fundamental property

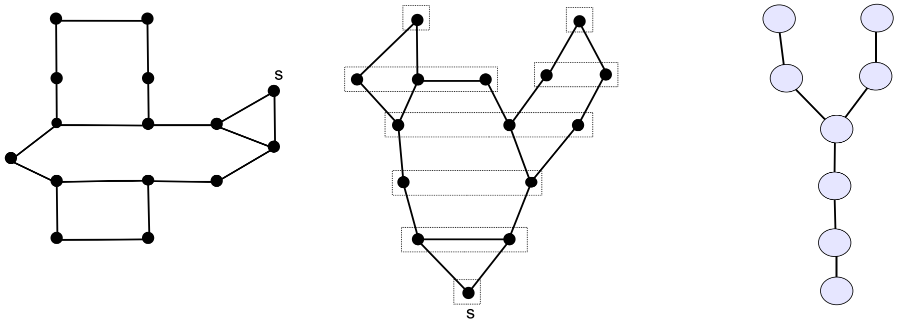

Fig. 2 A graph, its layering partition, and the tree Γ associated with that layering partition

of δ-hyperbolic spaces established in [38] that geodesics in such spaces diverge at exponential rate. For a proof, see also [17]. For a simple path ρ of a graph G, let $l(\rho)$ denote its length.

Lemma 1 [4, 11] *Let $G = (V, E)$ be a graph with δ-thin geodesic triangles and let ρ be a simple path connecting two vertices p, q of G. If $[p, q]$ is a geodesic segment between p and q, then for every vertex $x \in [p, q]$, the distance from x to a closest vertex y of ρ is at most $1 + \delta \log_2 l(\rho)$.*

3 Properties of Layering Partitions of δ-Hyperbolic Graphs

In this section, we describe layering partitions of δ-hyperbolic graphs ($\delta \geq 1/2$) and present their metric properties. These properties are used in the construction of sparse spanners and routing schemes.

Let $G = (V, E)$ be an unweighted connected graph with a distinguished vertex s and let $r := \max\{d_G(s, x) : x \in V\}$. A *layering* of G with respect to s is the decomposition of V into the *spheres* $L^i = \{u \in V : d(s, u) = i\}$, $i = 0, 1, 2, \dots, r$. The corresponding *layering partition* $\mathcal{LP} = \{L^i_1, \dots, L^i_{p_i} : i = 0, 1, 2, \dots, r\}$ of G is a partition of each L^i into *clusters* $L^i_1, \dots, L^i_{p_i}$ such that two vertices $u, v \in L^i$ belong to the same cluster L^i_j if and only if they can be connected by a path outside the ball $B_{i-1}(s)$ of radius $i - 1$ centered at s. This partition has been introduced in [10, 13] and recently have been used also in [7, 20, 21]. It was shown in [13] that for a given unweighted graph G such a layering partition can be found in $O(|E|)$ time. Clearly, for a given starting vertex s, the corresponding layering partition of G is unique. However, considering different starting vertices in G, one can get different layering partitions of G.

We continue by showing that if G is a graph with n vertices and with δ-thin geodesic triangles, then the diameters of clusters in any layering partition of G are bounded by a function of δ and $\log_2 n$. Note that the *diameter* of a set $S \subseteq V$ in a graph $G = (V, E)$ is defined here as $diam_G(S) := \max\{d_G(u, v) : u, v \in S\}$. In the literature, it is often called *weak diameter* (see, e.g., [43]) because distances are measured in original graph G, not in the subgraph of G induced by S.

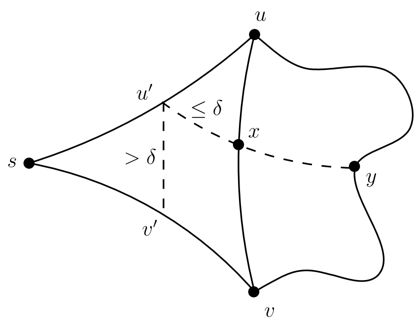

Fig. 3 To the proof of Proposition 2

Set $\Lambda_n := 4 + 3\delta + 2\delta \log_2 n$.

Proposition 2 *Let L^i_j be a cluster of a layering partition of a graph G with δ-thin geodesic triangles and n vertices, and let $u, v \in L^i_j$. Then $d_G(u, v) \le \Lambda_n$.*

Proof Suppose, by way of contradiction, that u, v belong to a common cluster L^i_j but $d_G(u, v) > \Lambda_n$. Let ρ be a simple path connecting the vertices u, v outside the ball $B_{i-1}(s)$. Let $[u, v]$ be a geodesic segment connecting the vertices u and v. Set $r := 2 + \delta + \delta \log_2 n$. On the sphere L^{i-r} pick two vertices u', v' of G such that u' lies on a geodesic segment $[s, u]$ between the root s and the vertex u while v' lies on a geodesic segment $[s, v]$ between s and v; see Fig. 3. Since $d_G(u, v) > 2\delta \log_2 n + 3\delta + 4$, we conclude that $d_G(u', v') > \delta$. Since the geodesic triangle formed by the geodesic segments $[s, u], [s, v], [u, v]$ is δ-thin, $d_G(s, u') = d_G(s, v')$, and $d_G(u', v') > \delta$, we conclude that $d(u', x) \le \delta$ for some vertex x of G lying on the geodesic segment $[u, v]$. By Lemma 1, the path ρ contains a vertex y such that $d_G(x, y) \le \delta \log_2 l(\rho) + 1 \le \delta \log_2 n + 1$. Thus $d_G(s, y) \le d_G(s, u') + d_G(u', x) + d_G(x, y) \le i - r + \delta + \delta \log_2 n + 1$. On the other hand, since y belongs to the path ρ, we must have $d_G(s, y) \ge i$. Thus $i \le i - r + \delta + \delta \log_2 n + 1$, hence $2 + \delta + \delta \log_2 n = r \le 1 + \delta + \delta \log_2 n$, a contradiction. □

Let Γ be a graph whose vertex set is the set of all clusters L^i_j in a layering partition $\mathcal{LP}$ of a graph G. Two vertices L^i_j and $L^{i'}_{j'}$ are adjacent in Γ if and only if there exist $u \in L^i_j$ and $v \in L^{i'}_{j'}$ such that u and v are adjacent in G (see Fig. 2). It is shown in [13] that Γ is a tree, called the *layering tree* of G, and that Γ is computable in linear time in the size of G.

Let $\mathcal{VT}$ be a shortest path tree spanning G and rooted at s. We call $\mathcal{VT}$ a *vertical spanning tree* of G. For integers $i \in \{1, 2, \ldots, r\}$ and $0 \le k \le i$, and any vertex $v \in L^i$, let $f^k(v)$ be the kth ancestor of v in $\mathcal{VT}$, i.e., the vertex on the (v, s)-path of the vertical tree $\mathcal{VT}$ located at distance k from v. Clearly, $f^k(v) \in L^{i-k}$ if $v \in L^i$. For any cluster L^i_j of the layering partition $\mathcal{LP}$ of G and any $0 \le k \le i$, let $F^i_j(k)$ be the set of kth ancestors of vertices of L^i_j in $\mathcal{VT}$.

Proposition 3 *Let L^i_j be a cluster of a layering partition of an n-vertex graph G with δ-thin geodesic triangles. Then* $d_G(x, y) \leq \delta$ *for every k such that* $\min\{\lceil \Lambda_n/2\rceil, i\} \leq k \leq i$ *and any* $x, y \in F^i_j(k)$.

Proof Consider arbitrary vertices $u, v \in L^i$ and set $\lambda := d_G(u, v)/2$. Denote by $[s, u]$ and $[s, v]$ the geodesic segments connecting in $\mathcal{V}T$ vertex s with u and v, respectively. Let also $[u, v]$ be any geodesic segment connecting u and v in G. Since $d_{\mathcal{V}T}(s, u) = d_{\mathcal{V}T}(s, v) = d_G(s, u) = d_G(s, v)$, for the geodesic triangle of G formed by the geodesic segments $[s, u]$, $[s, v]$ and $[u, v]$, we have $\alpha_u = \alpha_v = \lambda = i - \alpha_s$. All geodesic triangles of G are δ-thin, whence for any two vertices $a \in [s, u]$ and $b \in [s, v]$ with $d_G(a, s) = d_G(b, s) \leq \alpha_s$, the inequality $d_G(a, b) \leq \delta$ holds. Hence, $d_G(f^k(v), f^k(u)) \leq \delta$ whenever $\lceil\lambda\rceil \leq k \leq i$. Now, if both u, v belong to the same cluster $L^i_j \subseteq L^i$, then, by Proposition 2, $d_G(u, v) \leq \Lambda_n$. By the proof above, we get $d_G(f^k(v), f^k(u)) \leq \delta$ whenever $\lceil d_G(u, v)/2\rceil \leq k \leq i$. Consequently, $d_G(f^k(v), f^k(u)) \leq \delta$ for every k with $\min\{\lceil\Lambda_n/2\rceil, i\} \leq k \leq i$. □

Since geodesic triangles of a δ-hyperbolic graph G are 4δ-thin, the following corollary is immediate.

Corollary 1 *Let L^i_j be a cluster of a layering partition of an n-vertex δ-hyperbolic graph G. Then* $d_G(x, y) \leq 4\delta$ *for every k such that* $\min\{2(\Lambda_n - 3), i\} \leq k \leq i$ *and any* $x, y \in F^i_j(k)$.

4 Distance Approximating Trees and Distance Labeling Schemes

In this section, we present a simple method which constructs for any δ-hyperbolic graph $G = (V, E)$ with n vertices a distance $O(\delta \log n)$-approximating tree in optimal time $O(|E|)$. Recall that, a tree $\mathcal{T} = (V, F)$ is called a *distance κ-approximating tree* of a graph $G = (V, E)$ if $|d_G(x, y) - d_{\mathcal{T}}(x, y)| \leq \kappa$ for each pair of vertices $x, y \in V$ [10, 13]. Our result and the definition of a distance approximating tree are comparable with Theorem 1. The approximation of distances used in Theorem 1 is stronger because the mapping φ is non-expansive. On the other hand, distance approximating trees have the same set of vertices as G while the trees occurring in the theorem of Gromov may have Steiner points (in fact our construction can be easily modified to be non-expansive by accepting edges of length 1/2 and Steiner points). The error incurred by our result is slightly weaker (but of the same order), however the construction of our approximating tree $\mathcal{T}$ is simpler and can be done in linear $O(|E|)$ time while the construction in Theorem 1 needs $O(|V|^2)$ time. As a byproduct, we obtain also a $O(\delta \log n)$-additive distance labeling scheme with $O(\log^2 n)$ bit labels for the family of n-vertex δ-hyperbolic graphs (see Proposition 5).

Let Γ be the layering tree defined by the layering partition $\mathcal{L}P$ of G. To construct the distance approximating tree $\mathcal{T} = (V, F)$ of G, for each cluster $C := L^i_j$ of $\mathcal{L}P$ we select a vertex v_C of L^{i-1} which is adjacent in G with at least one vertex of C and make v_C adjacent in $\mathcal{T}$ to all vertices of C. Since Γ is a tree, $\mathcal{T}$ is a tree as well.

Since the layering partition $\mathcal{LP}$ of G can be constructed in linear time, the tree $\mathcal{T}$ can be constructed in linear $O(|E|)$ time, too (see [10, 13] for details).

Recall that $\Lambda_n := 4 + 3\delta + 2\delta\log_2 n$.

Proposition 4 *The tree $\mathcal{T} = (V, F)$ is a distance Λ_n-approximating tree for an n-vertex graph $G = (V, E)$ with δ-thin geodesic triangles. In particular, $\mathcal{T} = (V, F)$ is a distance $4(\Lambda_n - 3)$-approximating tree for a δ-hyperbolic graph G.*

Proof It can be easily seen that the tree $\mathcal{T}$ preserves the distances to the root s, i.e., $d_{\mathcal{T}}(x, s) = d_G(x, s)$ for any $x \in V$. From Proposition 2, if x, y belong to a common cluster, then $d_{\mathcal{T}}(x, y) = 2$ and $d_G(x, y) \le \Lambda_n$. Now, suppose that x and y belong to different clusters of Γ, say $x \in C' := L^{i'}_{j'}$ and $y \in C'' := L^{i''}_{j''}$. Let $C := L^i_j$ be the cluster which is the nearest common ancestor of C' and C'' in the tree Γ. By definition of clusters, any path of G connecting the vertices x and y will traverse the clusters lying on the unique path $P(C', C'')$ of the tree Γ between C' and C''. In particular, any shortest (x, y)-path will intersect the cluster C. Since $d_G(x, z) \ge i' - i$ and $d_G(z, y) \ge i'' - i$ for any vertex $z \in C$, we conclude that $d_G(x, y) \ge i' + i'' - 2i$. On the other hand, any (x, y)-path of G, sharing a single vertex with each cluster (except C) of the path $P(C', C'')$ and intersecting the cluster C in a shortest path, has length at most $i' + i'' - 2i + \Lambda_n$, thus $i' + i'' - 2i \le d_G(x, y) \le i' + i'' - 2i + \Lambda_n$. Now, notice that $d_{\mathcal{T}}(x, y) = i' + i'' - 2i + 2$ or $d_{\mathcal{T}}(x, y) = i' + i'' - 2i$ if the two clusters of $P(C', C'')$ incident to C have the same neighbor in $\mathcal{T}$. In both cases, we conclude that $|d_G(x, y) - d_{\mathcal{T}}(x, y)| \le \Lambda_n$. Now, since geodesic triangles of a δ-hyperbolic graph G are 4δ-thin, the second assertion is immediate. □

By using edges of length $\frac{1}{2}$ and Steiner points, the tree T can easily be transformed into a tree $T_{\frac{1}{2}}$ which has the same approximating performances and satisfies the non-expansive property. For this, for each cluster $C := L^i_j$ we introduce a Steiner point w_C, and add an edge of length $\frac{1}{2}$ between any vertex of C and w_C and an edge of length $\frac{1}{2}$ between w_C and the vertex v_C defined above.

Now, using a known result on distance labeling schemes for trees (see [35, 51]), we obtain the following result.

Proposition 5 *The family of δ-hyperbolic graphs G with n vertices and m edges admits an $O(\delta\log n)$-additive distance labeling scheme with $O(\log^2 n)$ bit labels and constant time distance decoder. The labeling scheme can be constructed in $O(m + n\log n)$ time.*

Proof Let $\mathcal{T} = (V, F)$ be a distance $4(\Lambda_n - 3)$-approximating tree of a δ-hyperbolic graph $G = (V, E)$ constructed above. We know that tree $\mathcal{T}$ can be constructed in linear $O(m)$ time for G. By [35, 51], there is a function labeling in $O(n\log n)$ total time the vertices of an n-vertex tree $\mathcal{T}$ with labels of up to $O(\log^2 n)$ bits such that given the labels of any two vertices v, u of $\mathcal{T}$, it is possible to compute in constant time the (exact) distance $d_{\mathcal{T}}(v, u)$, by merely inspecting the labels of u and v. By the proof of Proposition 4, we have $-2 \le d_G(x, y) - d_{\mathcal{T}}(x, y) \le 4(\Lambda_n - 3)$. Hence, the value $\bar{d}_G(u,v) := d_{\mathcal{T}}(u,v) + 4(\Lambda_n - 3)$ satisfies $0 \le \bar{d}_G(u,v) - d_G(u,v) \le 4(\Lambda_n - 3) + 2$. □

Note that one can get a similar result by just using a distance labeling scheme constructed for a Gromov's original (weighted) tree (see Theorem 1) as it was done in [30]. The additive error incurred by our result is slightly weaker (but of the same order), however the construction of our approximating tree $\mathcal{T}$, and therefore of our labeling scheme, is faster and simpler.

5 Additive Spanners

We continue with a simple method which constructs for any δ-hyperbolic graph $G = (V, E)$ with n vertices an additive $O(\delta \log n)$-spanner H with $O(\delta n)$ edges, i.e., a spanning subgraph H of G with at most $O(\delta n)$ edges such that $d_H(u,v) - d_G(u,v) \le O(\delta \log n)$ holds for any $u, v \in V$. Recall that, without loss of generality, we assumed that $\delta \ge 1/2$.

Let $\mathcal{LP} = \{L_1^i, \dots, L_{p_i}^i : i = 0, 1, 2, \dots, r\}$ be a layering partition of G where $L^0 = \{s\}$ and $r := \max\{d_G(s,x) : x \in V\}$ (see Sect. 3 for construction and notations). The graph H consists of a vertical spanning tree $\mathcal{VT}$ of G rooted at s and a set of *horizontal trees*, one such tree $\mathcal{HT}_j^i$ for each cluster L_j^i. From now on, set $\Lambda^* := 2(\Lambda_n - 3)$. If $i > \Lambda^*$, then the horizontal tree $\mathcal{HT}_j^i$ is a shortest path tree spanning in G the vertices of the set $F_j^i(\Lambda^*)$ and rooted at any vertex of $F_j^i(\Lambda^*)$. If $i \le \Lambda^*$, then $\mathcal{HT}_j^i$ is just one node tree, i.e., $\mathcal{HT}_j^i := \{s\}$. Notice that, according to Propositions 1 and 3, the diameter of each set $F_j^i(\Lambda^*)$ is at most 4δ.

Lemma 2 *The graph H is an additive $O(\delta \log n)$-spanner of G.*

Proof Let u, v be two vertices of G, and let $L_{j'}^{i'}, L_{j''}^{i''}$ be the clusters of G containing u and v, respectively. Let L_j^i be the cluster which is the nearest common ancestor of $L_{j'}^{i'}$ and $L_{j''}^{i''}$ in the layering tree Γ. Every path of G from u to v must intersect the cluster L_j^i. Since $d_G(u,z) \ge i' - i$ and $d_G(z,v) \ge i'' - i$ for any vertex $z \in L_j^i$, we conclude that $d_G(u,v) \ge i' + i'' - 2i$.

Let $u', v' \in L_j^i$ be the ancestors of u and v in the vertical tree $\mathcal{VT}$. If $i \le \Lambda^*$, then the distance in H between u and v is at most $d_{\mathcal{VT}}(u,s) + d_{\mathcal{VT}}(v,s) \le i' - i + \Lambda^* + i'' - i + \Lambda^* = i' + i'' - 2i + 4(\Lambda_n - 3)$. Hence, $d_H(u,v) - d_G(u,v) \le 4(\Lambda_n - 3)$, and we are done in this case. Assume now that $i > \Lambda^*$. Consider the vertices $u'' := f^{\Lambda^*}(u')$ and $v'' := f^{\Lambda^*}(v')$. We have $d_H(u,v) \le d_{\mathcal{VT}}(u,u'') + d_{\mathcal{HT}_j^i}(u'',v'') + d_{\mathcal{VT}}(v'',v) = i' - i + \Lambda^* + 8\delta + i'' - i + \Lambda^* = i' + i'' - 2i + 2\Lambda^* + 8\delta$ (by Proposition 3, both vertices u'' and v'' can be connected in $\mathcal{HT}_j^i$ to the root of $\mathcal{HT}_j^i$ by a path of length at most 4δ). Consequently, $d_H(u,v) - d_G(u,v) \le 4(\Lambda_n - 3) + 8\delta = 4 + 20\delta + 8\delta \log_2 n$, and we are done in this case, too. □

Lemma 3 *The graph H has at most $(4\delta + 1)(n - 1)$ edges.*

Proof The vertical tree $\mathcal{VT}$ has $n-1$ edges. Every horizontal tree $\mathcal{HT}^i_j$ has at most $|L^i_j|$ leaves and so at most $4\delta|L^i_j|$ edges, except when $i \le \Lambda^*$. In this latter case, $\mathcal{HT}^i_j$ contains no edges. The clusters $\{L^i_j : i = 1, \dots, r, \quad j = 1, \dots, p_i\}$ of G are disjoint, so the total number of edges of H is at most $n-1+4\delta(n-1) = (4\delta+1)(n-1)$. □

Thus, we proved the following result.

Proposition 6 *Every n-vertex δ-hyperbolic graph with $\delta \ge 1/2$ has an additive $O(\delta \log n)$-spanner with at most $O(\delta n)$ edges, and such a spanner can be constructed in polynomial time.*

6 Routing Labeling Scheme

To build a routing labeling scheme for an unweighted δ-hyperbolic graph G, we use the layering partition $\mathcal{LP} = \{L^i_1, \dots, L^i_{p_i} : i = 0, 1, 2, \dots, r\}$ of G, its layering tree Γ, and the vertical tree $\mathcal{VT}$ associated with Γ (see Sect. 3 and Sect. 5 for definitions). We also use Proposition 2, Proposition 3, Corollary 1, and a modification of the method proposed in [20] for routing in graphs with tree-length bounded by λ introduced in [21]. Our Proposition 3 is essential in obtaining $O(\delta \log^2 n)$-bit routing label size.

As in Sect. 3, we assume that the trees Γ and $\mathcal{VT}$ are rooted at $L^0 = \{s\}$ and s. Let again $f^k(v)$ be the kth ancestor of v in $\mathcal{VT}$, i.e., the vertex of the (v, s)-path of $\mathcal{VT}$ at distance k from v. For simplicity, we will use $f(v)$ for $f^1(v)$. To get routing labels for vertices of G, first we construct in $O(n)$ time a routing labeling scheme for the vertical tree $\mathcal{VT}$. As it was shown in [27, 58], one can assign to each vertex $v \in V$ a label *treelabel*(v) of size at most $O(\log n)$ bits, so that given *treelabel*(u) and *treelabel*(v) of two vertices of $\mathcal{VT}$, and nothing else, it is possible to determine in constant time, by a routing decision function $f($*treelabel*(u), *treelabel*$(v))$, the port number at u of the first edge on the unique path of $\mathcal{VT}$ from u to v. Recall that *treelabel*(v) contains the port number from v to its father $f(v)$ in $\mathcal{VT}$ and this information can be extracted in constant time from *treelabel*(v).

Then, for the layering tree Γ, we build in $O(n \log n)$ time a hierarchical tree $\mathcal{H}$ as follows. Find a centroid node M of Γ and let it to be the root of $\mathcal{H}$. (Recall that a *centroid node* of a tree T with p nodes is a node such that any subtree of T not containing it has at most $p/2$ nodes; a centroid node of a tree can be found in linear time). For each subtree of $\Gamma \setminus \{M\}$ construct a hierarchical tree recursively, and build $\mathcal{H}$ by connecting M to the roots of those trees. Clearly, the height of $\mathcal{H}$ is at most $\log_2 |V(\Gamma)| \le \log_2 n$.

In each cluster C of the layering partition $\mathcal{LP}$ we pick an arbitrary vertex r_C and call it *the center* of C. For each vertex v of G, let $C(v)$ denote the unique cluster of $\mathcal{LP}$ containing v. For each vertex $v \in V$ and for each cluster X which is an ancestor of $C(v)$ in $\mathcal{H}$, the label *Label*(v) of v in G will store a full description of the following shortest path *path*(v, X) of G (see Fig. 4 for an illustration):

– If X is also an ancestor of $C(v)$ in Γ, then $path(v, X)$ is a shortest path of G between the vertices $f^k(r_X)$ and $f^k(v')$, where v' is the ancestor in $\mathcal{VT}$ of v belonging to the cluster X and k is the smallest integer such that $d_G(f^k(r_X), f^k(v')) \leq 4\delta$ (by Corollary 1, such k exists). Let also $level(v, X) := d_G(s, f^k(r_X)) = d_G(s, f^k(v'))$.
– Otherwise, $path(v, X)$ is a shortest path of G between the vertices $f^t(r'_X)$ and $f^t(v')$, where r'_X and v' are the ancestors in $\mathcal{VT}$ of r_X and v, respectively, belonging to the cluster $Y := nca_\Gamma(C(v), X)$ and t is the smallest integer such that $d_G(f^t(r'_X), f^t(v')) \leq 4\delta$. Here, $nca_\Gamma(C(v), X)$ is the nearest common ancestor of $C(v)$ and X in rooted tree Γ. Set also, in this case, $level(v, X) := d_G(s, f^t(r'_X)) = d_G(s, f^t(v'))$.

Under the *full description of a path* $P := (x_1, \dots, x_l)$ we understand an ordered sequence of l triples. Each triple consists of the identification $id(x)$ (an integer from $\{1, \dots, n\}$) of a vertex x of P, the port number from x to the next vertex in P and the port number from x to the previous vertex in P (integers from $\{1, \dots, deg_G(x)\}$). For the end-vertices of the path, missing entries are nil. We assume that the sequence is ordered with respect to $id(\cdot)$s. Clearly, since the height of $\mathcal{H}$ is at most $\log_2 n$, each label $Label(v)$, $v \in V$, will store the descriptions of at most $\log_2 n$ such short, of length $\leq 4\delta$, paths. The routing label of a vertex $v \in V$ is

$$Label(v) := (id(v), treelabel(v), depthlabel(v), [help(v, X_0), help(v, X_1), \dots, help(v, X_h)]),$$

where

$$help(v, X_j) := [path(v, X_j), level(v, X_j), treelabel(r_{X_j})].$$

Here X_j is the ancestor of $C(v)$ in $\mathcal{H}$ at depth j and r_{X_j} is the center of X_j. The label $depthlabel(v)$ allows to compute in constant time, together with $depthlabel(u)$ of some other vertex u, the depth in the hierarchical tree $\mathcal{H}$ of $nca_\mathcal{H}(C(v), C(u))$. According to [35], the nodes of $\mathcal{H}$ can be assigned labels $depthlabel(X)$ of size $O(\log n)$ bits in such a way that the depth in $\mathcal{H}$ of $nca_\mathcal{H}(X, Y)$ can be computed in constant time given $depthlabel(X)$ and $depthlabel(Y)$. This part of $Label(v)$ will be useful in identifying an appropriate part of string $[help(v, X_0), help(v, X_1), \dots, help(v, X_h)]$ to be used in the routing decision. Summarizing, we conclude that the label $Label(v)$ of each vertex v of G consists of at most $O(\delta \log^2 n)$ bits.

Assume now that a vertex u wants to send a message to an arbitrary vertex v. First u creates a header h_{uv} of the message. For this, it extracts from $Label(u)$ and $Label(v)$ the parts $depthlabel(u)$ and $depthlabel(v)$ and uses them to compute in constant time the depth l in $\mathcal{H}$ of $nca_\mathcal{H}(C(u), C(v))$. Then,

$$h_{uv} := [treelabel(v), treelabel(r_{X_l}), rescue_1, level_1, rescue_2, level_2],$$

where $rescue_1 := path(u, X_l)$, $level_1 := level(u, X_l)$ and $rescue_2 := path(v, X_l)$, $level_2 := level(v, X_l)$. Clearly, h_{uv} consists of at most $O(\delta \log n)$ bits and can be computed in $O(1)$ time. The routing path from u to v follows the pattern depicted in Fig. 4: the packet moves on the vertical tree $\mathcal{VT}$ until path $rescue_1$ is reached, then

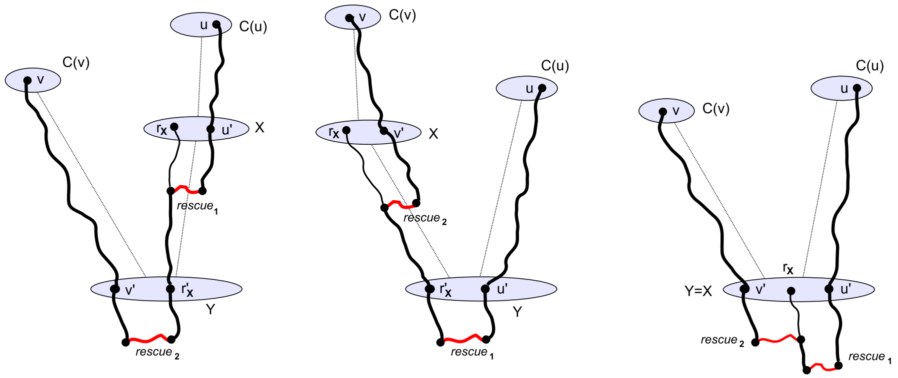

Fig. 4 The three possible locations of cluster X on the path of Γ between $C(u)$ and $C(v)$ (with respect to Y). The routing path induced by the scheme is indicated in all three cases. The horizontal parts are paths $rescue_1$ and $rescue_2$. The vertical parts are paths from the spanning tree $\mathcal{V}T$ of G. Note that, we show rooted trees growing upward, so the roots are on *bottom*

moves on $rescue_1$, then again on $\mathcal{V}T$ until path $rescue_2$ is reached, then moves on $rescue_2$, and then on $\mathcal{V}T$ until the destination vertex v is reached.

More precisely, let $X := nca_{\mathcal{H}}(C(u), C(v))$ and $Y := nca_{\Gamma}(C(u), C(v))$. By construction of $\mathcal{H}$ from Γ, we infer that X belongs to the unique path of Γ connecting $C(u)$ with $C(v)$. There are three possible locations of X on that path: X is between Y and $C(u)$, X is between Y and $C(v)$, or $X = Y$ (see Fig. 4 for an illustration). The routing algorithm proceeds as follows. Suppose that a packet with header h_{uv} is at a vertex w (initially, $w = u$). If $id(w) = id(v)$, then we are done. Otherwise, we check if w is an ancestor of v. This can be done in $O(1)$ time by using $treelabel(w)$ and $treelabel(v)$. For this, we check if the port number returned by $f(treelabel(v), treelabel(w))$ is the port number of the father $f(v)$ of v. If w is an ancestor of v, then we return $f(treelabel(w), treelabel(v))$ (we advance in $\mathcal{V}T$). Assume now that w is not an ancestor of v. Then, using the binary search, we check in $O(\log_2(4\delta))$ time if $id(w)$ belongs to the path $rescue_2$. If yes, then we extract the appropriate port number associated with w in $rescue_2$ (we advance in the path $rescue_2$). If no, then we check if w is an ancestor of r_X using $treelabel(w)$ and $treelabel(r_X)$. If w is an ancestor of r_X, then we return $f(treelabel(w), treelabel(r_X))$, if $level_1 < level_2$, and return port number between w and its father $f(w)$, otherwise (in both cases we advance in $\mathcal{V}T$). If w is not an ancestor of r_X (recall also that it is not an ancestor of v and it is not on the path $rescue_2$), then, using binary search we check in $O(\log_2(4\delta))$ time if $id(w)$ belongs to the path $rescue_1$. If yes, then we extract the appropriate port number associated with w in $rescue_1$ (we advance in the path $rescue_1$). Otherwise (w is an ancestor of u), we return the port number between w and its parent $f(w)$ (we advance in $\mathcal{V}T$). For each vertex w on the routing path, the decision where to go from w towards v takes $O(\log\delta)$ time in the worst case (i.e., if the binary search in $rescue_2$ or/and in $rescue_1$ is involved; otherwise, it would take only $O(1)$ time). Similarly to the proof of Lemma 2, we can show that the length of the path traveled by any packet from u to v is at most $d_G(u,v) + 4(\Delta_n - 3) + 8\delta = d_G(u,v) + 4 + 20\delta + 8\delta\log_2 n$.

Summarizing, we can formulate the main result of this section.

Proposition 7 *The family of δ-hyperbolic graphs with n vertices and $\delta \geq 1/2$ admits an $O(\delta \log n)$-additive routing labeling scheme with $O(\delta \log^2 n)$ bit labels. Once computed by the sender in $O(\delta)$ time, headers of size $O(\delta \log n)$ bits never change. Moreover, the scheme can be constructed in polynomial time and the routing decision takes $O(\log_2(4\delta))$ time per vertex.*

7 Conclusion

In this paper, for unweighted n-vertex δ-hyperbolic graphs with $\delta \geq 1/2$, we designed sparse spanners and compact routing and distance labeling schemes, all with an additive error $O(\delta \log n)$. It would be interesting to know if similar results can be obtained for δ-hyperbolic graphs with arbitrary edge weights and if the upper bounds obtained are optimal. With respect to lower bounds, we know only the following. Since graphs with tree-length λ are λ-hyperbolic (see [17]), from the lower bounds obtained in [22] for tree-length λ graphs, we conclude that there are δ-hyperbolic graphs for which every multiplicative δ-spanner (and thus every additive $(\delta - 1)$-spanner) must have $\Omega(n^{1+1/\Theta(\delta)})$ edges. Additionally, as we have mentioned earlier, the authors of [30] gave some lower bound results for DLS on δ-hyperbolic graphs. It was shown in [30] that the distance label size $O(\log^2 n)$ is optimal up to some constant for every additive error up to n^ϵ and that any s-multiplicative DLS using labels of any polylogarithmic size requires $s = \Omega(\log \log n)$. It would be interesting to know if similar lower bound results known for DLS can be obtained also for RLS on δ-hyperbolic graphs.

Acknowledgement We would like to thank the referees for many useful suggestions and remarks.

References

1. Abraham, I., Gavoille, C.: Object location using path separators. In: Proceedings of the Twenty-Fifth Annual ACM Symposium on Principles of Distributed Computing (PODC 2006), Denver, Colorado, USA, 23–26 July 2006, pp. 188–197. ACM, New York (2006)
2. Abraham, I., Gavoille, C., Goldberg, A.V., Malkhi, D.: Routing in networks with low doubling dimension. In: Proceedings of 26th International Conference on Distributed Computing Systems (ICDCS 2006), p. 75. IEEE Comput. Soc., Los Alamitos (2006)
3. Abraham, I., Balakrishnan, M., Kuhn, F., Malkhi, D., Ramasubramanian, V., Talwar, K.: Reconstructing approximate tree metrics. In: Proceedings of the Twenty-Sixth Annual ACM Symposium on Principles of Distributed Computing (PODC 2007), Portland, Oregon, USA, 12–15 August 2007, pp. 43–52. ACM, New York (2007)
4. Alonso, J.M., Brady, T., Cooper, D., Ferlini, V., Lustig, M., Mihalik, M., Shapiro, M., Short, H.: Notes on word hyperbolic groups. In: Ghys, E., Haefliger, A., Verjovsky, A. (eds.) Group Theory from a Geometrical Viewpoint, ICTP Trieste 1990, pp. 3–63. World Scientific, Singapore (1991),
5. Alstrup, S., Bille, P., Rauhe, T.: Labeling schemes for small distances in trees. SIAM J. Discrete Math. **19**, 448–462 (2005)
6. Alvarez, V., Portilla, A., Rodriguez, J.M., Touris, E.: Gromov hyperbolicity of Denjoy domains. Geom. Dedic. **121**, 221–245 (2006)

7. Bădoiu, M., Indyk, P., Sidiropoulos, A.: Approximation algorithms for embedding general metrics into trees. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2007), New Orleans, Louisiana, USA, 7–9 January 2007, pp. 512–521. SIAM, Philadelphia (2007)
8. Bandelt, H.-J., Chepoi, V.: 1-Hyperbolic graphs. SIAM J. Discrete Math. **16**, 323–334 (2003)
9. Bazzaro, F., Gavoille, C.: Localized and compact data-structure for comparability graphs. Discrete Math. **309**, 3465–3484 (2009)
10. Brandstädt, A., Chepoi, V., Dragan, F.: Distance approximating trees for chordal and dually chordal graphs. J. Algorithms **30**, 166–184 (1999)
11. Bridson, M., Haefliger, A.: Metric Spaces of Non-Positive Curvature. Springer, Berlin (1999)
12. Chan, H.T.-H., Gupta, A., Maggs, B.M., Zhou, S.: On hierarchical routing in doubling metrics. In: Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2005), Vancouver, British Columbia, Canada, 23–25 January 2005, pp. 762–771. SIAM, Philadelphia (2005)
13. Chepoi, V., Dragan, F.: A note on distance approximating trees in graphs. Eur. J. Comb. **21**, 761–766 (2000)
14. Chepoi, V., Estellon, B.: Packing and covering δ-hyperbolic spaces by balls. In: Proceedings of Approximation, Randomization, and Combinatorial Optimization: Algorithms and Techniques, 10th International Workshop, APPROX 2007, and 11th International Workshop, RANDOM 2007, Princeton, NJ, USA, 20–22 August 2007. Lecture Notes in Computer Science, vol. 4627, pp. 59–73. Springer, Berlin (2007)
15. Chepoi, V., Dragan, F.F., Yan, C.: Additive sparse spanners for graphs with bounded length of largest induced cycle. Theor. Comput. Sci. **347**, 54–75 (2005)
16. Chepoi, V., Dragan, F., Vaxès, Y.: Distance and routing labeling schemes for non-positively curved plane graphs. J. Algorithms **61**, 60–88 (2006)
17. Chepoi, V., Dragan, F.F., Estellon, B., Habib, M., Vaxès, Y.: Diameters, centers, and approximating trees of δ-hyperbolic geodesic spaces and graphs. In: Proceedings of the 24th Annual ACM Symposium on Computational Geometry (SoCG 2008), 9–11 June 2008, College Park, Maryland, USA, pp. 59–68 (2008)
18. Courcelle, B., Vanicat, R.: Query efficient implementation of graphs of bounded clique-width. Discrete Appl. Math. **131**, 129–150 (2003)
19. Cowen, L.: Compact Routing with Minimum Stretch. J. Algorithms **38**, 170–183 (2001)
20. Dourisboure, Y.: Compact routing schemes for generalised chordal graphs. J. Graph Algorithms Appl. **9**, 277–297 (2005)
21. Dourisboure, Y., Gavoille, C.: Tree-decompositions with bags of small diameter. Discrete Math. **307**, 208–229 (2007)
22. Dourisboure, Y., Dragan, F.F., Gavoille, C., Yan, C.: Spanners for bounded tree-length graphs. Theor. Comput. Sci. **383**, 34–44 (2007)
23. Dragan, F.F., Yan, C.: Collective tree spanners in graphs with bounded parameters. Algorithmica **57**, 22–43 (2010)
24. Dragan, F.F., Yan, C., Corneil, D.G.: Collective tree spanners and routing in AT-free related graphs. J. Graph Algorithms Appl. **10**, 97–122 (2006)
25. Dragan, F.F., Yan, C., Lomonosov, I.: Collective tree spanners of graphs. SIAM J. Discrete Math. **20**, 241–260 (2006)
26. Eilam, T., Gavoille, C., Peleg, D.: Compact routing schemes with low stretch factor. J. Algorithms **46**, 97–114 (2003)
27. Fraigniaud, P., Gavoille, C.: Routing in trees. In: Proceedings of the 28th International Colloquium on Automata, Languages and Programming (ICALP 2001), Crete, Greece, 8–12 July 2001. Lecture Notes in Computer Science, vol. 2076, pp. 757–772. Springer, Berlin (2001)
28. Gavoille, C., Gengler, M.: Space-efficiency of routing schemes of stretch factor three. J. Parallel Distrib. Comput. **61**, 679–687 (2001)
29. Gavoille, C., Hanusse, N.: Compact routing tables for graphs of bounded genus. In: Proceedings of the 26th International Colloquium on Automata, Languages and Programming (ICALP'99), Prague, Czech Republic, 11–15 July 1999. Lecture Notes in Computer Science, vol. 1644, pp. 351–360. Springer, Berlin (1999)
30. Gavoille, C., Ly, O.: Distance labeling in hyperbolic graphs. In: Proceedings of 16th International Symposium on Algorithms and Computation (ISAAC 2005), Sanya, Hainan, China, 19–21 December 2005. Lecture Notes in Computer Science, vol. 3827, pp. 171–179. Springer, Berlin (2005)
31. Gavoille, C., Paul, C.: Distance labeling scheme and split decomposition. Discrete Math. **273**, 115–130 (2003)

32. Gavoille, C., Paul, C.: Optimal distance labeling for interval graphs and related graph families. SIAM J. Discrete Math. **22**, 1239–1258 (2008)
33. Gavoille, C., Peleg, D.: Compact and localized distributed data structures. Distrib. Comput. **16**, 111–120 (2003)
34. Gavoille, C., Pérennès, S.: Memory requirements for routing in distributed networks. In: Proceedings of the Fifteenth Annual ACM Symposium on Principles of Distributed Computing (PODC 1996), Philadelphia, Pennsylvania, USA, 23–26 May 1996, pp. 125–133. ACM, New York (1996)
35. Gavoille, C., Katz, M., Katz, N.A., Paul, C., Peleg, D.: Approximate distance labeling schemes. In: Proceedings of the 9th Annual European Symposium on Algorithms (ESA 2001), Aarhus, Denmark, 28–31 August 2001. Lecture Notes in Computer Science, vol. 2161, pp. 476–488. Springer, Berlin (2001)
36. Gavoille, C., Peleg, D., Pérennès, S., Raz, R.: Distance labeling in graphs. J. Algorithms **53**, 85–112 (2004)
37. Ghys, E., de la Harpe, P. (eds.): Les Groupes Hyperboliques d'Après M. Gromov. Progress in Mathematics, vol. 83. Birkhäuser, Basel (1990)
38. Gromov, M.: Hyperbolic Groups. In: Gersten, S.M. (ed.) Essays in Group Theory. MSRI Series, vol. 8, pp. 75–263 (1987)
39. Gupta, A., Krauthgamer, R., Lee, J.R.: Bounded geometries, fractals, and low-distortion embeddings. In: Proceedings of the 44th Symposium on Foundations of Computer Science (FOCS 2003), Cambridge, MA, USA, 11–14 October 2003, pp. 534–543. IEEE Comput. Soc., Los Alamitos (2003)
40. Gupta, A., Kumar, A., Rastogi, R.: Traveling with a pez dispenser (or, routing issues in mpls). SIAM J. Comput. **34**, 453–474 (2005)
41. Har-Peled, S., Mendel, M.: Fast construction of nets in low dimensional metrics, and their applications, fast construction of nets in low-dimensional metrics and their applications. SIAM J. Comput. **35**, 1148–1184 (2006)
42. Karlsson, A., Noskov, G.: The Hilbert metric and Gromov hyperbolicity. Enseign. Math. **48**, 73–89 (2002)
43. Klein, P., Plotkin, S.A., Rao, S.: Excluded minors, network decomposition, and multicommodity flow. In: Proceedings of the 25th Annual ACM Symposium on Theory of Computing (STOC 1993), pp. 682–690. ACM, New York (1993)
44. Kleinberg, R.: Geographic routing using hyperbolic space. In: Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM 2007), Anchorage, Alaska, USA, 6–12 May 2007, pp. 1902–1909. IEEE Press, New York (2007)
45. Konjevod, G., Richa, A.W., Xia, D.: Optimal-stretch name-independent compact routing in doubling metrics. In: Proceedings of the 25th Annual ACM Symposium on Principles of Distributed Computing (PODC 2006), Denver, CO, USA, 23–26 July 2006, pp. 198–207. ACM, New York (2006)
46. Konjevod, G., Richa, A.W., Xia, D.: Optimal scale-free compact routing schemes in networks of low doubling dimension. In: Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2007), New Orleans, Louisiana, USA, 7–9 January 2007, pp. 939–948. SIAM, Philadelphia (2007)
47. Konjevod, G., Richa, A., Xia, D., Yu, H.: Compact routing with slack in low doubling dimension. In: Proceedings of the 26th Annual ACM Symposium on Principles of Distributed Computing (PODC 2007), Portland, Oregon, USA, 12–15 August 2007, pp. 71–80. ACM, New York (2007)
48. Konjevod, G., Richa, A.W., Xia, D.: Dynamic routing and location services in metrics of low doubling dimension. In: Proceedings of the 22nd International Symposium on Distributed Computing (DISC 2008), Arcachon, France, 22–24 September 2008, Lecture Notes in Computer Science, vol. 5218, pp. 379–393. Springer, Berlin (2008)
49. Koolen, J., Moulton, V.: Hyperbolic bridged graphs. Eur. J. Comb. **23**, 683–699 (2002)
50. Krauthgamer, R., Lee, J.R.: Algorithms on negatively curved spaces. In: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), Berkeley, California, USA, 21–24 October 2006, pp. 119–132. IEEE Comput. Soc., Los Alamitos (2006)
51. Peleg, D.: Distributed Computing: A Locality-Sensitive Approach. SIAM Monographs on Discrete Math. Appl. SIAM, Philadelphia (2000)
52. Peleg, D.: Proximity-preserving labeling schemes and their applications. J. Graph Theory **33**, 167–176 (2000)
53. Shavitt, Y., Tankel, T.: Hyperbolic embedding of internet graph for distance estimation and overlay construction. IEEE/ACM Trans. Netw. **16**, 25–36 (2008)

54. Slivkins, A.: Distributed approaches to triangulation and embedding. In: Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2005), Vancouver, British Columbia, Canada, 23–25 January 2005, pp. 640–649. SIAM, New York (2005)
55. Slivkins, A.: Distance estimation and object location via rings of neighbors. Distrib. Comput. **19**, 313–333 (2007)
56. Talwar, K.: Bypassing the embedding: algorithms for low dimensional metrics. In: Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC 2004), Chicago, IL, USA, 13–16 June 2004, pp. 281–290. ACM, New York (2004)
57. Thorup, M.: Compact oracles for reachability and approximate distances in planar digraphs. J. Assoc. Comput. Mach. **51**, 993–1024 (2004)
58. Thorup, M., Zwick, U.: Compact routing schemes. In: Proceedings of the Thirteenth Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA 2001), Heraklion, Crete, Greece, 4–6 July 2001, pp. 1–10. ACM, New York (2001)
59. Thorup, M., Zwick, U.: Approximate distance oracles. J. Assoc. Comput. Mach. **52**, 1–24 (2005)