Editorial Manager(tm) for Algorithmica
Manuscript Draft

Manuscript Number:

Title: Additive Spanners and Distance and Routing Labeling Schemes for Hyperbolic Graphs

Article Type: Original Research

Keywords: algorithms; distance and
routing labeling schemes; additive spanners; δ-hyperbolic
graphs.

Corresponding Author: Dr. Feodor F. Dragan, Ph.D.

Corresponding Author's Institution: Kent State University

First Author: Victor Chepoi, PhD

Order of Authors: Victor Chepoi, PhD; Feodor F. Dragan, Ph.D.; Bertrand Estellon, PhD; Michel Habib, PhD; Yann Vaxes, PhD; Yang Xiang

Additive Spanners and Distance and Routing Labeling Schemes for Hyperbolic Graphs*

Victor Chepoi[1], Feodor F. Dragan[2], Bertrand Estellon[1,3], Michel Habib[3], Yann Vaxès[1], and Yang Xiang[2]

[1] Laboratoire d'Informatique Fondamentale, Faculté des Sciences de Luminy
Université de la Mediterranée, Marseille, F-13288, Marseille Cedex 9, FRANCE
chepoi@lif.univ-mrs.fr, estellon@lif.univ-mrs.fr, vaxes@lif.univ-mrs.fr
[2] Algorithmic Research Laboratory, Department of Computer Science
Kent State University, Kent, OH 44242, USA
dragan@cs.kent.edu, yxiang@cs.kent.edu
[3] LIAFA Case 7014, University Paris Diderot - Paris 7
F-75205 Paris Cedex 13, FRANCE
habib@liafa.jussieu.fr

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 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. For example, (a) it has been shown empirically that the internet topology embeds with better accuracy into a hyperbolic space than into an Euclidean space of comparable dimension, (b) 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. A connected graph $G = (V, E)$ equipped with standard graph metric d_G is δ-*hyperbolic* if the metric space (V, d_G) is δ-hyperbolic. In this paper, using our Layering Partition technique, we provide a simpler construction of distance approximating trees of δ-hyperbolic graphs on n vertices with an additive error $O(\delta \log n)$ and show that every n-vertex δ-hyperbolic graph has an additive $O(\delta \log n)$-spanner with at most $O(\delta n)$ edges. As a consequence, we show that the family of δ-hyperbolic graphs with n vertices enjoys an $O(\delta \log n)$-additive routing labeling scheme with $O(\delta \log^2 n)$ bit labels and $O(\log \delta)$ time routing protocol, and an easier constructable $O(\delta \log n)$-additive distance labeling scheme with $O(\log^2 n)$ bit labels and constant time distance decoder.

Keywords: algorithms, distance and routing labeling schemes, additive spanners, δ-hyperbolic graphs.

1 Introduction

This paper investigates distributed abilities of δ-hyperbolic graphs for the problems of distance computation and routing. Commonly, when we make a query concerning a pair of vertices in a graph (adjacency, distance, shortest route, etc.), we need to make a global access to the structure. 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 [47, 34].

We are mainly interested here in the distance and routing labeling schemes, introduced by Peleg (see, e.g., [47]). *Distance labeling schemes* (DLS, for short) are schemes that label

* This research was partly supported by the ANR grant BLAN06-1-138894 (project OPTICOMB).

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. *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 source vertex and the label of a destination, it is possible to compute efficiently the port number of the edge from the source 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. A *routing scheme* is a mechanism that can deliver packets of information from any vertex of the network to any other vertex.

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 $d_G(v, u) \leq f(L(v), L(u)) \leq s \cdot d_G(v, u) + r$. 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*, and $(s, 0)$-approximate labeling schemes are called s*-multiplicative labeling schemes*. The distance and routing labeling schemes, we propose for δ-hyperbolic graphs, are additive in nature.

In this paper, using our Layering Partition technique, we provide a simpler construction of distance approximating trees of δ-hyperbolic graphs on n vertices with an additive error $O(\delta \log n)$ and show that every n-vertex δ-hyperbolic graph has an additive $O(\delta \log n)$-spanner with at most $O(\delta n)$ edges. As a consequence, we show that the family of δ-hyperbolic graphs with n vertices enjoys an $O(\delta \log n)$-additive routing labeling scheme with $O(\delta \log^2 n)$ bit labels and $O(\log \delta)$ time routing protocol, and an easier constructable $O(\delta \log n)$-additive distance labeling scheme with $O(\log^2 n)$ bit labels and constant time distance decoder.

1.1 δ-Hyperbolicity

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 $G = (V, E)$ equipped with standard graph metric d_G is δ*-hyperbolic* if the metric space (V, d_G) is δ-hyperbolic. Every 4-point metric d (tree-realizable or not) has a canonical representation in the rectilinear plane, cf. [25]. In Fig. 1, the three distance sums are ordered from small to large, thus implying $\xi \leq \eta$. Then η is half the difference of the largest and the smallest sum, while ξ is half the largest minus the medium sum. Hence, a metric space (X, d) is δ-hyperbolic iff ξ does not exceed δ for any four points u, v, w, x of X. 0-Hyperbolic metric spaces are exactly the tree metrics. 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).

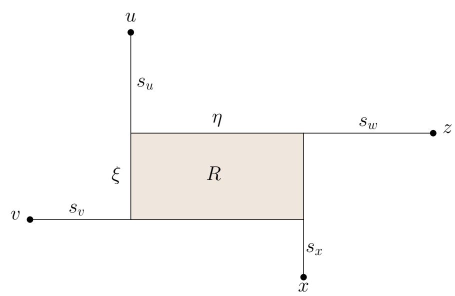

Fig. 1. Realization of a 4-point metric in the rectilinear plane.

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 [49] (see also [1]) 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 [1, 14, 16, 31, 43, 45]. Kleinberg showed [43] 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 [45] 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 [16] 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. [14] gave efficient algorithms for fast and accurate estimations of diameters and radii of δ-hyperbolic geodesic spaces and graphs.

In [7, 14, 44], some 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 factor 2 or 3 [10], from which it follows that those graphs have low δ-hyperbolicity (this result has been extended in [13] to all graphs in which the largest induced cycle is bounded by some constant δ; this result implies that those graphs are δ-hyperbolic). In general, the distance in a δ-hyperbolic space on n points can be approximated within an additive factor of $2\delta \log_2 n$ by a weighted tree metric [38, 37] and this approximation is sharp. For n-vertex δ-hyperbolic graphs G, in the conference paper [14] we described an alternative (linear time) construction of a tree approximating the distances of G with an additive error of $O(\delta \log_2 n)$. Since this construction is intensively used in the current paper, we present it here in more details.

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 [35], and that trees [5, 48], bounded tree-width graphs [35], distance-hereditary graphs [32], bounded clique-width graphs [18], non-positively curved plane graphs [15], 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 [35], implying that all the results mentioned above are tight as well, since all those graph families contain trees. Later, [33, 9] 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 [55]. 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 [30], it is proved that trees (and bounded tree-width graphs as well) enjoy a $(1+1/\log n)$-multiplicative DLS with labels of $O(\log n \log \log n)$ 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 [30]. Recently, the graphs with doubling dimension α have been considered, i.e., the graphs for which, for every r, each ball of radius $2r$ can be covered by at most 2^α balls of radius r. It generalizes Euclidian metrics and bounded growth graphs, and includes many realistic networks. After several successive improvements [39, 41, 46, 50, 52], the best scheme for them to date, due to Slivkins [51], is a $(1+\epsilon)$-multiplicative DLS with $O(\epsilon^{-O(\alpha)} \log n \log \log n)$ bit labels. This is optimal for bounded α, by combining the results of [46] and the lower bound of [30] for trees. Note also that planar graphs enjoy a $(1+\epsilon)$-multiplicative DLS with labels of $O(\epsilon^{-1} \log^3 n)$ bits (see [40, 53]). This has been generalized in [2] 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 [31]. 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 [31] needs $O(|V|^2)$ time). Note also that our distance approximating tree has n vertices while that of [31] may have about $O(n^2)$ vertices. Paper [31] 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 [36]). A better routing scheme is known for trees. In [27, 54], 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 and that can infer the distance between two vertices from their labels in constant time. A shortest path routing labeling schemes with $O(\log^2 n)$ bit labels are known for bounded tree-width graphs [35, 22] and non-positively curved plane graphs [15].

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 [17, 26, 54]. A 3-multiplicative RLS that uses labels of size $\tilde{O}(n^{2/3})$ was obtained in [17], and a 5-multiplicative RLS that uses labels of size $\tilde{O}(n^{1/2})$ was obtained in [26].[1] Later, authors of [54] 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 [36, 28]). 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 [53]. This has been generalized in [2] to graphs excluding a fixed minor with the same stretch and space bounds. Routing in graphs with doubling dimension α has been considered in [3, 12, 51, 52]. 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 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 [19, 22–24] and papers cited therein).

2 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 graph $G = (V, E)$ equipped with its standard distance d_G can be transformed into a geodesic (networklike) 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

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

$\alpha_z := (d(z,y)+d(z,x)-d(y,x))/2$ from 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. 2 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.*

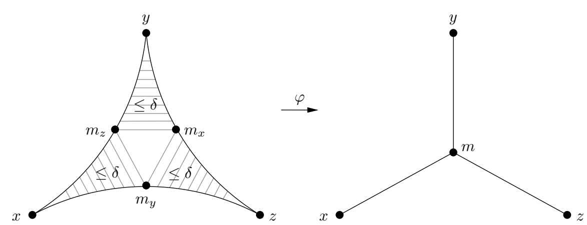

Fig. 2. A geodesic triangle $\Delta(x,y,z)$, the points m_x, m_y, m_z, and the tripod $\Upsilon(x',y',z')$

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. *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 $x,y \in X$.*

We conclude this section with a property of δ-hyperbolic graphs formulated and proven in several texts on Gromov hyperbolic spaces (in particular, in [11]) for all δ-hyperbolic spaces. This result is used in the proof of the fundamental property of δ-hyperbolic spaces established in [38] that geodesics in such spaces diverge at exponential rate; for a proof, see also [4, 11]. For a simple path ρ of a graph G, let $l(\rho)$ denote its length.

Proposition 2. *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 $1 + \delta \log_2 l(\rho)$.*

Proof. To explain why $\delta \log_2 l(\rho)$ occurs in this result, we sketch its (nice) proof, at the same time bringing it some computer science flavor (a detailed proof is given on p. 401 of [11]). Take the cycle constituted by the geodesic $[p, q]$ and the path ρ and "triangulate" it in the following way. If ρ consists of a single edge, then return this edge. Otherwise, pick the middle vertex r of the path ρ and include in the triangulation the geodesic triangle Δ having $[p, q]$ and two geodesic segments $[p, r]$ and $[r, q]$ as sides. Then recursively apply this algorithm twice to the geodesic segments $[p, r]$ and $[r, q]$ and the subpaths ρ' and ρ'' of ρ comprised between p and r, and r and q, respectively. The resulting triangulation $\mathcal{T}$ can be viewed as a binary tree rooted at Δ whose nodes are the triangles of $\mathcal{T}$ and two triangles are adjacent iff they share a common geodesic segment. Since the length of the current simple path is divided by 2 at each iteration, the number of levels h of this binary tree satisfies the inequality $l(\rho)/2^{h+1} < 1 \leq l(\rho)/2^h$.

For a vertex $x \in [p, q]$, the distance from x to one of the sides $[p, r]$ or $[r, q]$ of the geodesic triangle Δ is at most δ, because Δ is δ-thin. Suppose that $d_G(x, x') \leq \delta$ for a vertex $x' \in [p, r]$. Let Δ' be the geodesic triangle sharing the side $[p, r]$ with Δ. Repeating recursively the same operation on x' and Δ', we will construct a path from the initial vertex x to a vertex y of ρ consisting of at most h geodesic segments of length at most δ each. Hence $d_G(x, y) \leq 1 + \delta \log_2 l(\rho)$. □

3 Layering partitions

In this section, we describe the layering partitions of δ-hyperbolic graphs and establish their metric properties. These results will be used in the subsequent sections in the construction of sparse spanners and routing schemes.

Let $G = (V, E)$ be a 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* $L^i = \{u \in V : d(s, u) = i\}$, $i = 0, 1, 2, \ldots, r$. A *layering partition* $\mathcal{LP} = \{L^i_1, \ldots, L^i_{p_i} : i = 0, 1, 2, \ldots, r\}$ of G is a partition of each L^i into *clusters* $L^i_1, \ldots, L^i_{p_i}$ 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 [8, 21]). We continue by showing that if G is a graph with n vertices and with δ-thin geodesic triangles, then the diameters of clusters of a layering partition of G are bounded by a function of δ and $\log_2 n$. Set $\Lambda_n := 4 + 3\delta + 2\delta \log_2 n$.

Proposition 3. *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 $d_G(u, v) \leq \Lambda_n$.*

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 $d(u',x) \leq \delta$ for some vertex x of G lying on the geodesic segment $[u,v]$. By Proposition 2, the path ρ contains a vertex y such that $d_G(x,y) \leq \delta \log_2 l(\rho) + 1 \leq \delta \log_2 n + 1$. Thus $d_G(s,y) \leq d_G(s,u') + d_G(u',x) + d_G(x,y) \leq i - r + \delta + \delta \log_2 n + 1$. On the other hand, since y belongs to the path ρ, we must have $d_G(s,y) \geq i$. Thus $i \leq i - r + \delta + \delta \log_2 n + 1$, hence $2 + \delta + \delta \log_2 n = r \leq 1 + \delta + \delta \log_2 n$, a contradiction. □

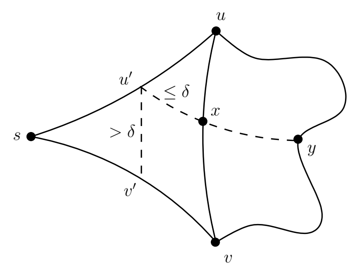

Fig. 3. To the proof of Proposition 3

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. 4). 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 $\mathcal{V}T$ be a shortest path tree spanning G and rooted at s. We call $\mathcal{V}T$ a *vertical spanning tree* of G. For integers $i \in \{1,2,\ldots,r\}$ and $0 \leq k \leq i$, and any vertex $v \in L^i$, let $f^k(v)$ be the kth ancestor of v in $\mathcal{V}T$, i.e., the vertex on the (v,s)-path of the vertical tree $\mathcal{V}T$ 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 $0 \leq k \leq i$, let $F^i_j(k)$ be the set of kth ancestors of vertices of L^i_j in $\mathcal{V}T$.

Proposition 4. *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 3, $d_G(u,v) \leq \Lambda_n$. 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.

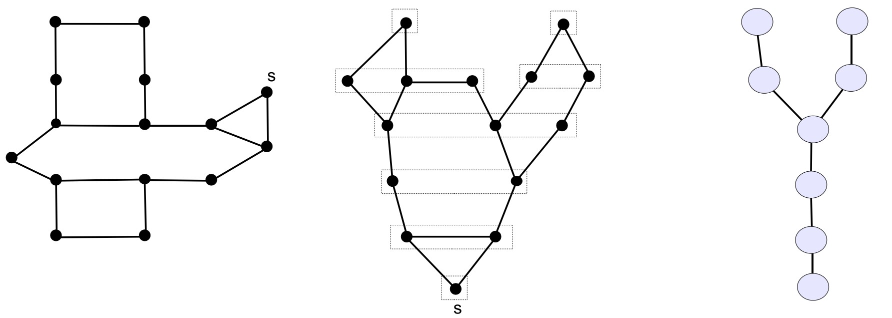

Fig. 4. A graph, its layering partition, and the tree Γ associated with that layering partition.

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

4 Distance labeling schemes and additive spanners

In this section, first 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 $|d_G(x, y) - d_{\mathcal{T}}(x, y)| \le \kappa$ 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 an easily constructable $O(\delta \log n)$-additive distance labeling scheme with $O(\log^2 n)$ bit labels for any δ-hyperbolic graph G with n vertices, i.e., we can assign to each vertex of G a label of size $O(\log^2 n)$-bits, such that, given the labels of any two vertices u and v of G, the distance $d_G(u, v)$ between them can be approximated within an additive error $O(\delta \log n)$ in constant time by merely inspecting their labels.

Let Γ be the layering tree defined by the layering partition $\mathcal{LP}$ of G. To construct the distance approximating tree $\mathcal{T} = (V, F)$ of G, for each cluster $C := L^i_j$ of $\mathcal{LP}$ 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. Clearly, $\mathcal{T}$ can be constructed in linear time, too.

Proposition 5. *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 3, if x, y belong to a common cluster, then $d_{\mathcal{T}}(x, y) = 2$ and $d_G(x, y) \leq \Lambda_n$. 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 $d_G(x, z) \geq i' - i$ and $d_G(z, y) \geq i'' - i$ for any vertex $z \in C$, we conclude that $d_G(x, y) \geq i' + i'' - 2i$. 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 $i' + i'' - 2i \leq d_G(x, y) \leq i' + i'' - 2i + \Lambda_n$. 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 $|d_G(x, y) - d_{\mathcal{T}}(x, y)| \leq \Lambda_n$. 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 [47, 30]), we obtain the following result.

Proposition 6. *The family of δ-hyperbolic graphs G with n vertices and m edges enjoys an $O(\delta \log n)$-additive distance labeling scheme with $O(\log^2 n)$ bit labels and constant time distance decoder. The labeling scheme is constructible 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 [47, 30], 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 5, we have $-2 \leq d_G(x, y) - d_{\mathcal{T}}(x, y) \leq 4(\Lambda_n - 3)$. Hence, the value $\bar{d}_G(u, v) := d_{\mathcal{T}}(u, v) + 4(\Lambda_n - 3)$ satisfies $0 \leq \bar{d}_G(u, v) - d_G(u, v) \leq 4(\Lambda_n - 3) + 2$. □

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. The graph H consists of a vertical spanning tree $\mathcal{V}T$ of G rooted at s and a set of *horizontal trees*, one such tree $\mathcal{H}T^i_j$ for each cluster L^i_j. From now on, set $\Lambda^* := 2(\Lambda_n - 3)$. If $i > \Lambda^*$, then the horizontal tree $\mathcal{H}T^i_j$ is a shortest path tree spanning in G the vertices of the set $F^i_j(\Lambda^*)$ and rooted at any vertex of $F^i_j(\Lambda^*)$. If $i \leq \Lambda^*$, then $\mathcal{H}T^i_j$ is just one node tree, i.e., $\mathcal{H}T^i_j := \{s\}$. Notice that, according to Propositions 1 and 4, the diameter of each set $F^i_j(\Lambda^*)$ is at most 4δ.

Lemma 1. *The graph H is an additive $O(\delta \log n)$-spanner of G.*

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

Let $u', v' \in L^i_j$ be the ancestors of u and v in the vertical tree $\mathcal{V}T$. If $i \leq \Lambda^*$, then the distance in H between u and v is at most $d_{\mathcal{V}T}(u,s) + d_{\mathcal{V}T}(v,s) \leq i' - i + \Lambda^* + i'' - i + \Lambda^* = i' + i'' - 2i + 4(\Lambda_n - 3)$. Hence, $d_H(u,v) - d_G(u,v) \leq 4(\Lambda_n - 3)$, 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 $d_H(u,v) \leq d_{\mathcal{V}T}(u,u'') + d_{\mathcal{H}T^i_j}(u'',v'') + d_{\mathcal{V}T}(v'',v) = i' - i + \Lambda^* + 8\delta + i'' - i + \Lambda^* = i' + i'' - 2i + 2\Lambda^* + 8\delta$ (by Proposition 4, both vertices u'' and v'' can be connected in $\mathcal{H}T^i_j$ to the root of $\mathcal{H}T^i_j$ by a path of length at most 4δ). Consequently, $d_H(u,v) - d_G(u,v) \leq 4(\Lambda_n - 3) + 8\delta = 4 + 20\delta + 8\delta \log_2 n$, and we are done in this case, too. □

Lemma 2. *The graph H has at most $(4\delta + 1)(n-1)$ edges.*

Proof. The vertical tree $\mathcal{V}T$ has $n-1$ edges. Every horizontal tree $\mathcal{H}T^i_j$ has at most $|L^i_j|$ leaves and so at most $4\delta |L^i_j|$ edges, except when $i \leq \Lambda^*$. In this latter case, $\mathcal{H}T^i_j$ contains no edges. The clusters $\{L^i_j : i = 1, \ldots, r, \quad j = 1, \ldots, p_i\}$ 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 7. *Every n-vertex δ-hyperbolic graph has an additive $O(\delta \log n)$-spanner with at most $O(\delta n)$ edges, constructible in polynomial time.*

5 Routing labeling scheme

To build a routing labeling scheme for a δ-hyperbolic graph G, we use the layering partition $\mathcal{LP} = \{L^i_1, \ldots, L^i_{p_i} : i = 0, 1, 2, \ldots, r\}$ of G, its layering tree Γ, and the vertical tree $\mathcal{V}T$ associated with Γ. We also use Proposition 3, Corollary 1, and a modification of the method proposed in [19] for routing in graphs with tree-length bounded by λ introduced in [21]. Additive 4λ-spanners with $O(\lambda n)$ edges for such graphs have been constructed in [20]. Note that any tree-length λ graph is λ-hyperbolic [14] but that the converse is not true.

As usually, we assume that the trees Γ and $\mathcal{V}T$ are rooted at $L^0 = \{s\}$ and s. Let again $f^k(v)$ be the kth ancestor of v in $\mathcal{V}T$, i.e., the vertex of the (v,s)-path of $\mathcal{V}T$ 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 $\mathcal{V}T$. As it was shown in [27, 54], one can assign to each vertex $v \in V$ a label $label(v)$ of size at most $O(\log n)$ bits, so that given $label(u)$ and $label(v)$ of two vertices of $\mathcal{V}T$, and nothing else, it is possible to determine in constant time, by a routing decision function $f(label(u), label(v))$, the port number at u of the first edge on the unique path of $\mathcal{V}T$ from u to v. Recall that $label(v)$ contains the port number from v to its father $f(v)$ in $\mathcal{V}T$ and this information can be extracted in constant time from $label(v)$.

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 $\log_2 |V(\Gamma)| \leq \log_2 n$.

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:

– 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 $\mathcal{V}T$ 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 $\mathcal{V}T$ of r_X and v, respectively, belonging to the cluster $Y := nca_\Gamma(C(v), X)$ and t is the smallest integer such that $d_G(f^t(r'_X), f^t(v')) \leq 4\delta$. 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* $P := (x_1, \ldots, x_l)$ we understand an ordered sequence of l triples. Each triple consists of the identification $id(x)$ (an integer from $\{1, \ldots, n\}$) 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 $\{1, \ldots, deg_G(x)\}$). 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

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

where

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

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 [30], 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 $[help(v, X_0), help(v, X_1), \ldots, help(v, X_h)]$ 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,

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

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 is constructible in $O(1)$ time. The routing path from u to v follows the pattern depicted in Fig. 5: the packet moves on the vertical tree $\mathcal{V}T$ until path $rescue_1$ is reached, then 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.

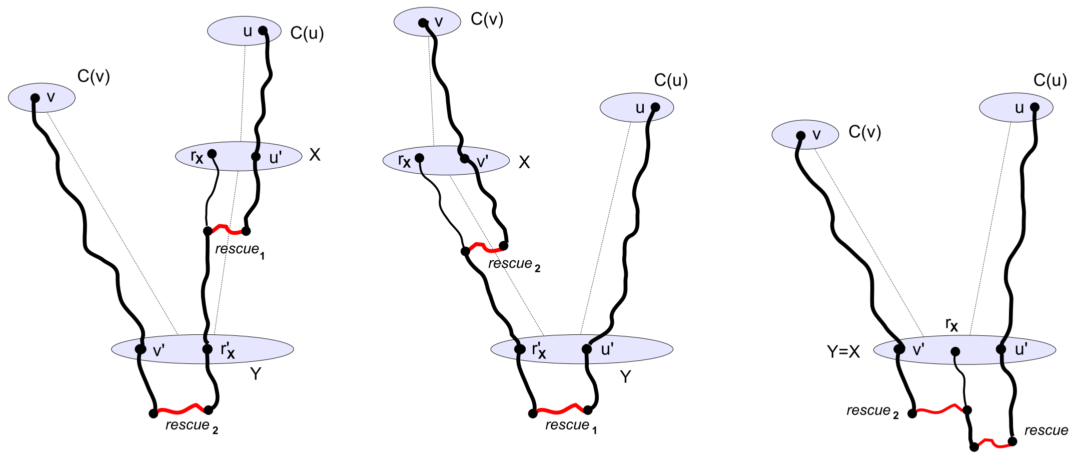

Fig. 5. The three possible locations of cluster X on the path of Γ between $C(u)$ and $C(v)$ (with respect to Y). The routing path induced by the scheme is indicated in all three cases. The paths $rescue_1$ and $rescue_2$ are shown in red. The black parts are paths from the spanning tree $\mathcal{V}T$ of G. Note that, we show rooted trees growing upward, so the roots are on bottom.

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. 5 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 $label(w)$ and $label(v)$. For this, we check if the port number returned by $f(label(v), label(w))$ is the port number of the father $f(v)$ of v. If w is an ancestor of v, then we return $f(label(w), label(v))$ (we advance in $\mathcal{V}T$). Assume now that w is not an ancestor of v. Then, using the binary search, we check in $O(\log \delta)$ 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 $label(w)$ and $label(r_X)$. If w is an ancestor of r_X, then we return $f(label(w), label(r_X))$, 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 $O(\log \delta)$ 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 1, 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 8. *The family of δ-hyperbolic graphs with n vertices enjoys an $O(\delta \log n)$-additive routing labeling scheme with $O(\delta \log^2 n)$ bit labels. Once computed by the sender in $O(1)$ time, headers of size $O(\delta \log n)$ bits never change. Moreover, the scheme is constructible in polynomial time and the routing decision takes $O(\log \delta)$ time per vertex.*

References

1. I. Abraham, M. Balakrishnan, F. Kuhn, D. Malkhi, V. Ramasubramanian, K. Talwar, Reconstructing approximate tree metrics, *Proceedings of the Twenty-Sixth Annual ACM Symposium on Principles of Distributed Computing (PODC 2007)*, Portland, Oregon, USA, August 12-15, 2007, ACM, pp. 43-52.
-yann
2. I. Abraham, C. Gavoille, Object location using path separators, *Proceedings of the Twenty-Fifth Annual ACM Symposium on Principles of Distributed Computing (PODC 2006)*, Denver, Colorado, USA, July 23-26, 2006, ACM, pp 188-197.
3. I. Abraham, C. Gavoille, A.V. Goldberg, D. Malkhi, Routing in Networks with Low Doubling Dimension, *ICDCS 2006*, p. 75 (see also http://dept-info.labri.fr/ gavoille/article/AGGM06).
4. J.M. Alonso, T. Brady, D. Cooper, V. Ferlini, M. Lustig, M. Mihalik, M. Shapiro, and H. Short, Notes on word hyperbolic groups, *Group Theory from a Geometrical Viewpoint, ICTP Trieste* 1990 (E. Ghys, A. Haefliger, and A. Verjovsky, eds.), World Scientific, 1991, pp. 3–63.
5. S. Alstrup, P. Bille, T. Rauhe, Labeling schemes for small distances in trees, *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2003)*, January 12-14, 2003, Baltimore, Maryland, USA, ACM/SIAM, pp. 689–698.
6. V. Alvarez, A. Portilla, J.M. Rodriguez, and E. Touris, Gromov hyperbolicity of Denjoy domains, *Geom. Dedicata* 121 (2006) 221-245.
7. H.-J. Bandelt and V. Chepoi, 1-Hyperbolic graphs, *SIAM J. Discr. Math.* 16 (2003) 323–334.
8. M. Bădoiu, P. Indyk, and A. Sidiropoulos, Approximation algorithms for embedding general metrics into trees, *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2007)*, New Orleans, Louisiana, USA, January 7-9, 2007, SIAM, pp. 512-521.
9. F. Bazzaro, C. Gavoille, Localized and Compact Data-Structure for Comparability Graphs, *ISAAC 2005*, pp. 1122–1131.
10. A. Brandstädt, V. Chepoi, and F. Dragan, Distance approximating trees for chordal and dually chordal graphs. *J. Algorithms* 30 (1999) 166–184.
11. M. Bridson, A. Haefliger, *Metric Spaces of Non-Positive Curvature,* Springer, Berlin, 1999
12. H. T.-H. Chan, A. Gupta, B. M. Maggs, and S. Zhou, On hierarchical routing in doubling metrics, *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2005)*, Vancouver, British Columbia, Canada, January 23-25, 2005. SIAM, pp. 762–771.
13. V. Chepoi and F. Dragan, A note on distance approximating trees in graphs, *Europ. J. Combin.* 21 (2000) 761–766.
14. V. Chepoi, F.F. Dragan, B. Estellon, M. Habib and Y. Vaxès, Diameters, centers, and approximating trees of δ-hyperbolic geodesic spaces and graphs, In *SoCG 2008*.
15. V. Chepoi, F. Dragan, Y. Vaxès, Distance and routing labeling schemes for non-positively curved plane graphs, *Journal of Algorithms,* 61 (2006), 60–88.
16. V. Chepoi and B. Estellon, Packing and covering δ-hyperbolic spaces by balls, In *APPROX-RANDOM 2007* pp. 59-73.
17. L. Cowen, Compact routing with minimum stretch, *SODA 1999*, pp. 255–260.
18. B. Courcelle and R. Vanicat, Query efficient implementation of graphs of bounded clique-width, *Discrete Applied Mathematics,* 131 (2003), 129–150.
19. Y. Dourisboure, Compact Routing Schemes for Bounded Tree-Length Graphs and for k-Chordal Graphs, *DISC 2004*, 365–378.
20. Y. Dourisboure, F.F. Dragan, C. Gavoille, and C. Yan, Spanners for bounded tree-length graphs, *Theor. Comput. Sci.* 383 (2007) 34–44.
21. Y. Dourisboure and C. Gavoille, Tree-decompositions with bags of small diameter, *Discr. Math.* 307 (2007) 208–229.

22. F.F. Dragan and C. Yan, Collective Tree Spanners in Graphs with Bounded Genus, Chordality, Tree-width, or Clique-width, *ISAAC 2005*, LNCS 3827, pp. 583–592.
23. F.F. Dragan, C. Yan and D.G. Corneil, Collective Tree Spanners and Routing in AT-free Related Graphs, *Journal of Graph Algorithms and Applications,* 10 (2006), 97–122.
24. F.F. Dragan, C. Yan and I. Lomonosov, Collective tree spanners of graphs, *SIAM J. Discrete Math.* 20 (2006), 241–260.
25. A.W.M. Dress, Trees, tight extensions of metric spaces, and the cohomological dimension of certain groups: a note on combinatorial properties of metric spaces, *Adv. Math. 53* (1984) 321–402.
26. T. Eilam, C. Gavoille, D. Peleg, Compact routing schemes with low stretch factor, *PODC 1998*, pp. 11–20.
27. P. Fraigniaud and C. Gavoille, Routing in Trees, *ICALP 2001*, 757–772.
28. C. Gavoille, M. Gengler, Space–efficiency of routing schemes of stretch factor three, *J. Parallel and Distr. Comput.*, 61 (2001), 679-687.
29. C. Gavoille, N. Hanusse, Compact Routing Tables for Graphs of Bounded Genus, *ICALP 1999*, LNCS 1644, pp. 351–360.
30. C. Gavoille, M. Katz, N.A. Katz, C. Paul, and D. Peleg, Approximate distance labeling schemes, In *ESA 2001* pp. 476-488.
31. C. Gavoille and O. Ly, Distance labeling in hyperbolic graphs, In *ISAAC 2005* pp. 171–179.
32. C. Gavoille, C. Paul, Distance labeling scheme and split decomposition, *Discrete Mathematics*, 273 (2003), 115–130.
33. C. Gavoille, C. Paul, Optimal Distance Labeling for Interval and Circular-Arc Graphs, *ESA 2003*, pp. 254–265.
34. C. Gavoille and D. Peleg, Compact and localized distributed data structures, *Distributed Computing,* 16 (2003), 111–120.
35. C. Gavoille, D. Peleg, S. Pérennès, and R. Raz, Distance labeling in graphs, *J. Algorithms,* 53 (2004), 85–112.
36. C. Gavoille, S. Pérennès, Memory requirements for routing in distributed networks, *PODC 1996*, pp. 125–133.
37. E. Ghys and P. de la Harpe eds., Les groupes hyperboliques d'après M. Gromov, *Progress in Mathematics* Vol. 83 Birkhäuser (1990).
38. M. Gromov, Hyperbolic Groups, In: *Essays in group theory* (S.M. Gersten ed.), MSRI Series 8 (1987) pp. 75–263.
39. A. Gupta, R. Krauthgamer, and J.R. Lee, Bounded geometries, fractals, and low-distortion embeddings, *FOCS 2003*, pp. 534–543.
40. A. Gupta, A. Kumar, and R. Rastogi, Traveling with a pez dispenser (or, routing issues in mpls), *SIAM Journal on Computing,* 34 (2005), 453–474.
41. S. Har-Peled, M. Mendel, Fast construction of nets in low dimensional metrics, and their applications, *Proceedings of the 21st ACM Symposium on Computational Geometry (SoCG 2005)*, Pisa, Italy, June 6-8, 2005, ACM, pp. 150–158.
42. A. Karlsson and G. Noskov, The Hilbert metric and Gromov hyperbolicity, *Ens. Math.* 48 (2002), 73-89.
43. R. Kleinberg, Geographic routing using hyperbolic space, In *INFOCOM 2007*, pp. 1902–1909.
44. J. Koolen and V. Moulton, Hyperbolic bridged graphs, *Europ. J. Combin.* 23 (2002) 683–699.
45. R. Krauthgamer and J.R. Lee, Algorithms on negatively curved spaces, In *FOCS 2006.*
46. M. Mendel and S. Har-Peled, Fast construction of nets in low dimensional metrics, and their applications, *SoCG 2005*, pp. 150–158.
47. D. Peleg, Distributed Computing: A Locality-Sensitive Approach, *SIAM Monographs on Discrete Math. Appl.*, SIAM, Philadelphia, 2000.
48. D. Peleg, Proximity-Preserving Labeling Schemes and Their Applications, *J. of Graph Theory*, 33 (2000), 167–176.
49. Y. Shavitt and T. Tankel, On internet embedding in hyperbolic spaces for overlay construction and distance estimation, In *INFOCOM 2004.*
50. A. Slivkins, Distributed approaches to triangulation and embedding, *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2005)*, Vancouver, British Columbia, Canada, January 23-25, 2005, SIAM, pp. 640-649.
51. A. Slivkins, Distance estimation and object location via rings of neighbors, *PODC 2005*, pp. 41–50.
52. K. Talwar, Bypassing the embedding: Algorithms for low dimensional metrics, *STOC 2004*, pp. 281–290.
53. M. Thorup, Compact oracles for reachability and approximate distances in planar digraphs, *Journal of the ACM,* 51 (2004), 993–1024.
54. M. Thorup and U. Zwick, Compact routing schemes, in *Proc. 13th Ann. ACM Symp. on Par. Alg. and Arch. (SPAA 2001)*, ACM 2001, pp. 1–10.
55. M. Thorup and U. Zwick, Approximate distance oracles, *Journal of the ACM,* 52 (2005), 1–24.