SLICE: Enabling Greedy Routing in High Genus 3-D WSNs With General Topologies

Chen Wang, *Member, IEEE*, Hongbo Jiang, *Senior Member, IEEE*, Tianlong Yu, *Student Member, IEEE*, and John C. S. Lui, *Fellow, IEEE, ACM*

***Abstract*—In this paper, we propose a highly efficient scheme, SLICE (a scalable and low stretch routing scheme), enabling greedy routing for wireless sensor networks (WSNs) deployed in complex-connected 3-D settings, whose topologies are often theoretically modeled as high genus 3-D WSNs. Compared to previous 3-D greedy embedding techniques, SLICE improves both the robustness and applicability. 1) It achieves a smaller distance distortion and a lower routing stretch with guaranteed delivery. While it follows the basic idea to embed the surface network to a planar topology to enable greedy routing, the embedding method proposed in SLICE is novel. We first slice the surface network to a genus-0 open surface with exactly one boundary. Then, to achieve a lower distance distortion, we purposely propose a variation of the Ricci flow algorithm, by which this open surface is flattened not to a planar annulus, but to a planar convex polygon, resulting in a lower routing stretch. 2) This is the first work, to the best of our knowledge, that enables greedy routing in high genus 3-D WSNs with general topologies. SLICE not only works for high genus 3-D surface WSNs, but also can be easily adapted to more general cases: high genus 3-D surface networks with holes, and high genus 3-D volume networks. For a high genus 3-D surface network with holes, SLICE embeds it to a planar convex polygon with circular holes, where our proposed greedy routing variation can be applied. For a high genus 3-D volume network, SLICE embeds the inner nodes to a height structure attached to the convex polygon, and a variation of greedy routing scheme with guaranteed delivery is proposed in this structure. The effectiveness of SLICE is validated by extensive simulations.**

***Index Terms*—Greedy routing, embedding techniques, WSNs, high genus, complex-connected 3-D settings.**

Manuscript received June 11, 2014; revised January 18, 2015; accepted July 19, 2015; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor P.-J. Wan. Date of publication September 17, 2015; date of current version August 16, 2016. This work was supported in part by the National Natural Science Foundation of China under Grants 61271226, 61272410, 61202460, and 61471408; the National High-Tech R&D Program ("863" Program) of China under Grants 2014AA01A701 and 2015AA011303; the National Natural Science Foundation of Hubei Province under Grant 2014CFA040; the China Postdoctoral Science Foundation under Grant 2014M560608; the Fundamental Research Funds for the Central Universities under Grant 2015QN073; and the Science and Technology Plan Projects of Wuhan City under Grant 2015010101010022. Part of this work was presented at IEEE INFOCOM 2013. *(Corresponding author: Hongbo Jiang.)*

C. Wang and H. Jiang are with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: hongbojiang2004@gmail.com).

T. Yu was with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China. He is now with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 USA.

J. C. S. Lui is with the Computer Science and Engineering Department, The Chinese University of Hong Kong (CUHK), Hong Kong.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNET.2015.2464312

I. INTRODUCTION

AS THEIR scales dramatically increase and their topologies become much more complex [2], [3], wireless sensor networks (WSNs) with complex shapes are in an urgent need for scalable greedy routing as a critical enabling component. Examples of such networks include: 1) high genus 3-D closed surface WSNs [4], e.g., corridors of buildings as shown in Fig. 1(a) and (b); 2) high genus 3-D surface WSNs with holes, e.g., coal mine tunnels as shown in Fig. 1(d) and (e); 3) high genus 3-D volume WSNs, e.g., underwater networks as shown in Fig. 1(g) and (h). The WSNs deployed in these scenarios are often of complex-connected 3-D settings with nontrivial topologies (i.e., multiple handles) and can be collectively called high genus 3-D WSNs.

In this paper, we focus on schemes enabling greedy routing in high genus 3-D WSNs with general topologies. While there exist a series of previous studies exploring greedy routing for 2-D networks [5]–[7] and simple 3-D volume networks (with at most one inner boundary) [3], few of them can work for high genus 3-D WSNs. In the following, we first review previous work on greedy routing in 2-D/3-D scenarios before we present our ideas.

A. Related Work

Greedy routing forwards packets by selecting next hops that are progressively closer to the destination. It is appealing for its simplicity and scalability, as its routing decision is made with only local knowledge. However, a packet may get stuck in a *local minima*, where greedy forwarding cannot proceed. To deal with this, face routing [5], [8] exploits the fact that a concave void in a 2-D planar network is a face with a simple perimeter. When a local minima is encountered, the packet employs face routing to traverse along the perimeter, until greedy forwarding is achievable. In contrast to a 2-D planar network, the void in 3-D is not a 2-D face, and its perimeter becomes a surface, so there does not exist a deterministic localized algorithm that guarantees delivery in 3-D networks [3], [9], rendering face routing infeasible for 3-D WSNs due to an arbitrarily large number of possible paths to be explored. To tackle this issue, several approaches [10]–[12] are proposed to enable greedy routing with structure-based routing. For example, a recent proposal, multihop Delaunay triangulation (MDT) routing [12], utilizes a virtual Delaunay triangulation to aid greedy routing. When it is stuck at a local minima, a packet is forwarded via a virtual link to a multihop Delaunay triangulation neighbor closest to the destination. However, the construction and maintenance of MDT are not purely localized and require centralized operations.

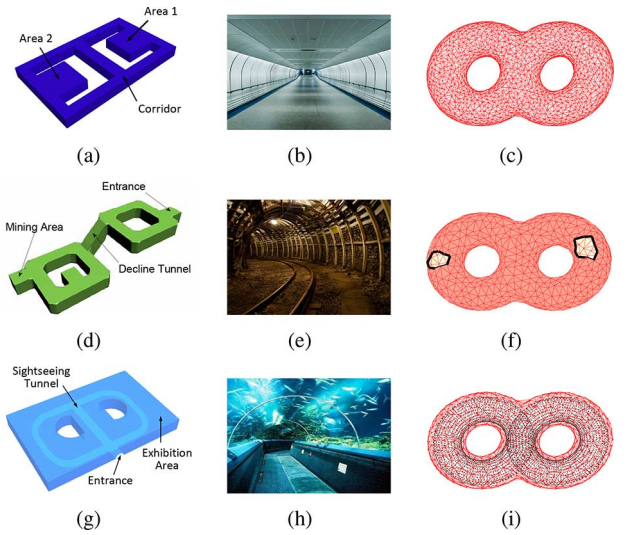


Fig. 1. Networks of (a), (b) corridors of buildings, homotopically equivalent to (c) a genus-2 3-D closed-surface network. (d), (e) Coal mine tunnels, homotopically equivalent to (f) a genus-2 3-D surface network with holes. (g), (h) Underwater aquariums, homotopically equivalent to (i) a genus-2 3-D volume network.

A theoretically sound solution to ensure the success of greedy routing in 3-D networks is greedy embedding [3], [13], which maps the original 3-D topology to a planar surface or to a virtual sphere where greedy routing can be applied. For instance, bubble routing algorithm proposed in [13] first decomposes a 3-D network into a set of hollow spherical cells (HSCs). While routing across HSCs is guided by a global routing table, greedy routing in an HSC is guaranteed by a continuous and one-to-one spherical mapping as well as a virtual tree structure established inside each HSC. Nevertheless, none of these greedy embedding algorithms can be extended to high genus 3-D surface WSNs [4]. The reason is that the surface $\mathbf{M}$ of the embedded topology should be continuously deformed to the surface $\mathbf{D}$ of the embedding topology, i.e., $\mathbf{M}$ and $\mathbf{D}$ should be in the same homotopy class. A high genus 3-D surface WSN [as shown in Fig. 1(c)], however, is apparently not homotopically equivalent to a 2-D planar surface or a 3-D sphere [14], and thus greedy embedding is not directly applicable for high genus 3-D surface WSNs.

Yu *et al.* [4] conducted a pioneer work on scalable routing in high genus 3-D surfaces. They first utilized graph embedding to decompose the network into genus-0 components (called pants in the original paper), and then enabled greedy routing following a two-level paradigm similar to that in [15]. However, one major concern of this approach is its centralized operations of finding the genus and decomposing the network, which makes the algorithm impractical for distributed sensor networks. Besides, the compact routing among components requires every node to maintain a routing table to all other components [4], possibly resulting in high storage overhead on individual nodes when the network grows large and becomes more complex.

The only distributed and scalable solution for routing in high genus 3-D surfaces was proposed in our previous work [1], where we proposed a distributed algorithm, SINUS, to slice the high genus surface to form one single genus-0 surface with two boundaries, which was then mapped to a planar annulus via the Ricci flow for delivery guaranteed greedy routing. Though it guarantees the packet delivery, this method may introduce a large distortion in distance metric]see Fig. 2(a)], when mapping the generated genus-0 surface to a planar annulus, as the planar annulus is not a good approximation in shape to preserve the distance metric on the genus-0 surface, as to be discussed in Section II-C.

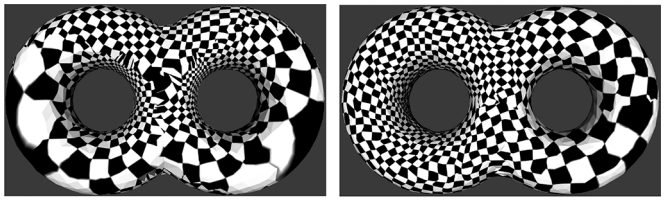


Fig. 2. (a) SINUS converted texture. (b) SLICE converted texture. Even distribution of the black and white rectangles indicates a small distance distortion.

In addition, both algorithms [1], [4] provide no solution for high genus 3-D surfaces with holes [e.g., Fig. 1(f)] or high genus 3-D volume WSNs [e.g., Fig. 1(i)], which are nontrivial topologies for complex-connected 3-D WSNs deployed in real settings.

B. Our Contributions

In this paper, we propose SLICE, a scalable and distributed routing algorithm with guaranteed delivery and low stretch for high genus 3-D WSNs. As a follow-up of our previous work [1], SLICE provides two more salient features: 1) It achieves a *smaller distance distortion* and thus a lower routing stretch while still guaranteeing a 100% packet delivery [see Fig. 2(b)]. 2) It is the first scheme, to the best of our knowledge, that enables greedy routing in high genus 3-D WSNs with *general topologies*: SLICE not only works for high genus 3-D surface WSNs, but also can be easily adapted to high genus 3-D surface networks with holes and high genus 3-D volume networks.

More specifically, we start with a genus-n surface network without holes by first extracting a maximum cut set (n cuts) based on the Reeb graph. Then, those cuts are connected leveraging the idea of Depth-First-Search (DFS). By doing so, a genus-0 open surface with exactly *one* boundary emerges. To achieve a low distance distortion, we purposely propose a variation of the Ricci flow algorithm, by which we flatten this genus-0 open surface into a planar *convex polygon*. Finally, greedy routing is allowed on the planar convex polygon via assigning nodes virtual coordinates.

SLICE can be further extended to more general cases: high genus 3-D surface networks with holes and high genus 3-D volume networks. For a high genus 3-D surface network with holes, SLICE embeds it to a planar convex polygon with *circular holes*, where our proposed greedy routing variation can be applied. For a high genus 3-D volume network, SLICE embeds the inner nodes to a *height structure* attached to the convex polygon, and a variation of the greedy routing scheme with guaranteed delivery is proposed based upon this structure. In summary, SLICE has the ability to improve the robustness and applicability as compared to previous studies.

The rest of the paper is organized as follows. We describe the theoretical foundation of our proposed algorithm in Section II. In Section III, we introduce our SLICE algorithm for WSNs on high genus 3-D surfaces. In Section IV, we extend SLICE to

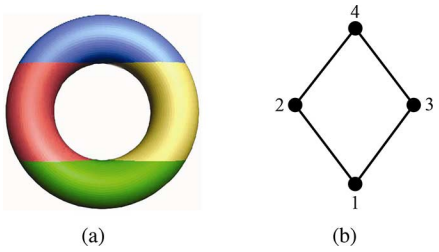


Fig. 3. (a) Genus-1 torus with contour lines. (b) Reeb graph of the genus-1 torus in (a).

more general topologies including high genus 3-D surfaces with holes and high genus 3-D volumes. In Section V, we discuss the time complexity, the message complexity, and the storage cost of SLICE. Simulation results are illustrated in Section VI. Finally, Section VII concludes the paper.

II. Preliminary

A. Cut and Genus

In the content of algebraic topology [16], a *cut* C is referred to as a nonintersecting closed simple curve on a connected and orientable surface $\mathbf{M}$. One notable property of a cut is its ability to *locally* disconnect the topology of $\mathbf{M}$. We call a cut set of $\mathbf{M}$ $\mathcal{C}_{\max} = \{C_1, C_2, \ldots, C_n\}$ a *maximum cut set*, if and only if: 1) every two cuts in $\mathcal{C}_{\max}$ belong to different homotopy classes, i.e., one cut cannot be smoothly deformed to another without leaving the surface; 2) $\mathbf{M} \setminus (C_1 \cup C_2 \cup \cdots \cup C_n)$ is connected; 3) $\mathbf{M} \setminus (C_1 \cup C_2 \cup \cdots \cup C_{n+1})$ is disconnected. Accordingly, the *genus* [16] of $\mathbf{M}$ is given by the cardinality of maximum cut set $\mathcal{C}_{\max}$, representing the maximum number of cuts on $\mathbf{M}$ without disconnecting $\mathbf{M}$. For example, Fig. 3(a) shows a genus-1 torus, while Fig. 1(c) shows a genus-2 surface. Generally, one cut C is able to slice a genus-n surface to genus-$(n-1)$.

Following [1] where the surface $\mathbf{M}$ is decomposed to a simple genus-0 topology using cuts, our idea is to exploit a similar method to achieve the sliced surface based on the Reeb graph of $\mathbf{M}$, and we hereby put forward a variation of the Ricci flow algorithm to embed it to a planar convex polygon, aiming at achieving a lower distance distortion.

B. Morse Function and Reeb Graph

A Morse function [17] is defined as a mapping $f : \mathbf{M} \to \mathbb{R}$, from a manifold $\mathbf{M}$ to a real number set $\mathbb{R}$. A typical Morse function on sensor networks is a mapping from a successive of contour lines to a real set $\{0, 1, \ldots, I\}$ of the height value [1], [18]. Specially, $f^{-1}(i)$ is called an *i-level set* of the Morse function. The structure of an arbitrary sensor network surface (without holes) can be explicitly represented by the Reeb graph that tracks the evolution of the connected components of the level sets $f^{-1}(\cdot)$.

The Reeb graph [19] has a mathematical foundation in the Morse theory. It can be obtained by contracting the connected components of the level sets $f^{-1}(\cdot)$ into points, based on the topological changes of the level sets. Specifically, a *point* of the Reeb graph corresponds to a connected component. *Nodes* of the Reeb graph are formed by the points that pass through *critical points* (namely, the minima, saddle, and maxima) of f, where the gradient of $f^{-1}(\cdot)$ will vanish when the number of the connected components of $f^{-1}(\cdot)$ increases or decreases. Its *arcs* are formed by the rest of the points, i.e., by the family of connected components that do not change the topology. Therefore, a Reeb graph is determined by the changes in the numbers of the connected components of the level sets (see Fig. 3 for instance).

Based on the Reeb graph, we turn to extracting a maximum cut set $\mathcal{C}_{\max}$ from $\mathbf{M}$. Our approach is motivated by the following theorem.

Theorem 1: The Reeb graph of a closed orientable genus-n 2-manifold has exactly n loops [20].

Theorem 1 implies that we can first identify all loops of the Reeb graph, thereby finding a cut for each loop. Specifically, a loop in a Reeb graph is associated with a node that ends the loop.

Definition 1: A region (arc of the Reeb graph) of $\mathbf{M}$ is a *loop-end region* if it is merged from two different regions by a loop-end node of the Reeb graph.

Another observation is that the Reeb graph is a connected graph and starts from exactly one node. That is, we have the following corollary.

Corollary 2: Each loop in the Reeb graph of $\mathbf{M}$ corresponds to one loop-end arc.

As such, in order to identify a cut for one loop, our method is to find the bisection in the loop-end region that disconnects this loop, and further achieving the sliced surface before it is embedded to a planar convex polygon.

C. Surface Ricci Flow and Its Variation

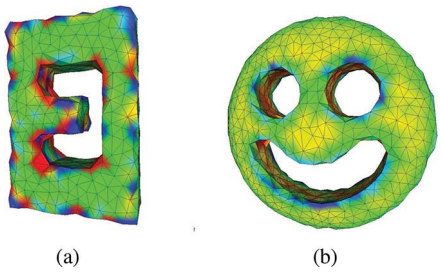
Ricci flow is an intrinsic geometric flow that deforms the metric of a Riemannian manifold. It was introduced by Richard Hamilton for general Riemannian manifolds in his seminal work [21] and has been utilized to prove the Poincare conjecture on 3-D manifolds [22]–[24].

1) Physical Intuition: Given a surface with a Riemannian metric, the metric induces the Gaussian curvature. When the metric is changed, the Gaussian curvature will be changed accordingly. Consider a deformation of the metric in the following way: At each point, the metric is locally scaled such that the scaling factor is proportional to the curvature at the point. After the deformation, the curvature will be changed. As the deformation process is repeated, both the metric and the curvature will evolve such that the curvature evolution is like a heat diffusion process. In the end, the Gaussian curvature is constant everywhere, and the limiting metric is *conformal* to the original one.

2) Surface Ricci Flow: Surface Ricci flow is a powerful tool to construct conformal Riemannian metrics, such that the metrics induce the user-defined Gaussian curvatures on the surface. Let $\mathbf{S}$ be a surface in $\mathbb{R}^3$, with a Riemannian metric g induced from the Euclidean metric of $\mathbb{R}^3$. The Gaussian curvature and the geodesic curvature of $\mathbf{S}$ are determined by the Riemannian metric, while the total curvature is a topological invariant.

Theorem 3: (Gauss–Bonnet) Suppose $\mathbf{S}$ is a compact 2-manifold with its boundary $\partial\mathbf{S}$. Then, the total curvature is given by $\int_{\mathbf{S}} K dA + \int_{\partial\mathbf{S}} k_g ds = 2\pi\chi(\mathbf{S})$, where K is the Gaussian curvature on interior points, k_g is the geodesic curvature on $\partial\mathbf{S}$, dA is the area element under the metric g, and $\chi(\mathbf{S})$ is the Euler characteristic number of $\mathbf{S}$.

The surface Ricci flow [25] deforms the metric of a Riemannian manifold in the manner analogous to the heat diffusion, by which the irregularities of the metric (see Fig. 4) can be smoothed out. In particular, the curvature evolution is the same as the heat diffusion on the surface: $\frac{K(t)}{dt} = -\Delta_{g(t)} K(t)$, where



(a) (b)

Fig. 4. Unevenly distributed curvature on 3-D surfaces depicted by color: (a) genus-1 corridor network; (b) genus-3 smile network.

$K(t)$ is the Gaussian curvature of the metric $g(t)$, $-\Delta_{g(t)}$ is the Laplace–Beltrami operator induced by $g(t)$, and t is the time parameter. Let $g(t) = e^{2u}g(0)$, then the surface Ricci flow can be simplified as $\frac{du(t)}{dt} = -2K(t)$. It has been proved that for a closed surface, if the total area of the surface is preserved during the flow, the Ricci flow will converge to a metric such that the Gaussian curvature is constant everywhere [21], [26].

3) Variation of Surface Ricci Flow: When directly applying the surface Ricci flow to the genus-0 3-D surface $\mathbf{S}$, the total curvature of $\mathbf{S}$, equal to $2\pi\chi(\mathbf{S})$, is evenly distributed on the boundary $\partial\mathbf{S}$. As a consequence, $\mathbf{S}$ is mapped to a disk $\mathbf{D}$, as is commonly used in [7] and [27]. Even though it guarantees the packet delivery for greedy routing in such disk $\mathbf{D}$, this plain method can introduce a large distortion in the distance metric when mapping $\mathbf{S}$ to a planar disk $\mathbf{D}$, especially when $\mathbf{S}$ is a 3-D surface generated by slicing a high-genus surface $\mathbf{M}$. The reason is that the plain method maps a 3-D open surface $\mathbf{S}$ of any shape to a disk $\mathbf{D}$, while for $\mathbf{S}$ obtained by slicing high-genus surface $\mathbf{M}$, the disk $\mathbf{D}$ is not a good approximation in shape of $\mathbf{S}$.

To better characterize the distance metric distortion, we have the following definition.

Definition 2: Suppose v_a and v_b are two nodes on the surface $\mathbf{S}$, $d_{ab\in\mathbf{S}}$ is the distance metric of v_a and v_b in $\mathbf{S}$, and $d_{ab\in\mathbf{D}}$ is the image distance metric of $d_{ab\in\mathbf{S}}$ in the disk $\mathbf{D}$, then the distance distortion in $\mathbf{D}$ is defined as

$$\triangle\mathcal{D} = \frac{d_{ab\in\mathbf{D}} - \overline{d_{ab\in\mathbf{D}}}}{\overline{d_{ab\in\mathbf{D}}}} = \frac{d_{ab\in\mathbf{D}}}{\overline{d_{ab\in\mathbf{D}}}} - 1 \qquad (1)$$

where $\overline{d_{ab\in\mathbf{D}}}$ is the straight-line (or shortest) distance, and $\frac{d_{ab\in\mathbf{D}}}{\overline{d_{ab\in\mathbf{D}}}}$ is the distance stretch in $\mathbf{D}$.

Since the deformation process of Ricci flow is conformal (angle-preserving) [28], [29], the value of the distance stretch is closely related to the shape changes from $\mathbf{S}$ to $\mathbf{D}$: the more shape changes, the larger distance stretch will be. Therefore, a larger distance stretch will induce a larger distance distortion.

The above point is further illustrated by the following example: In Fig. 5(a), a genus-1 torus is sliced open by a cut line in green and a slice line in blue. A, B, C and D are four corner angles generated by the intersection of the cut line and the slice line. Unfolding this sliced torus yields a 3-D open surface $\mathbf{S}$ that is close in shape to a rectangle in Fig. 5(b), with two distance metric $d_{ab\in\mathbf{S}}$ and $d_{ac\in\mathbf{S}}$ parallel to direction $D \to C$ and $D \to A$, respectively, and $d_{ab\in\mathbf{S}}$ equal to $d_{ac\in\mathbf{S}}$. Fig. 5(c) presents the case when $\mathbf{S}$ is mapped to a disk $\mathbf{D}$. Obviously, $d_{ac\in\mathbf{D}}$ becomes much shorter than $d_{ab\in\mathbf{D}}$ due to the distortion introduced by mapping $\mathbf{S}$ to $\mathbf{D}$. It is also observed

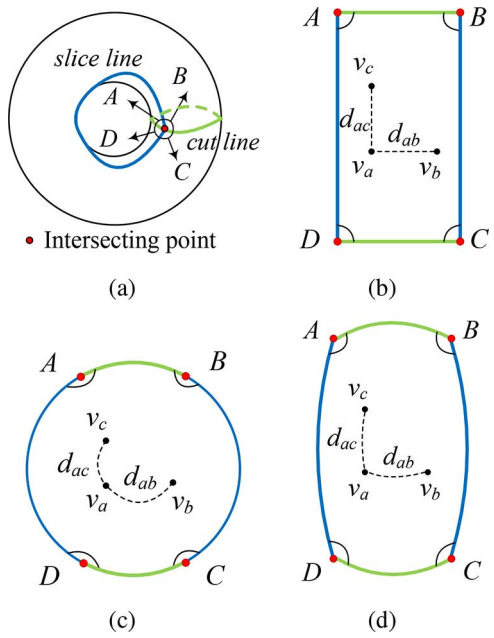


Fig. 5. (a) Genus-1 torus. (b) General shape of the sliced torus $\mathbf{S}$. (c) Embedding $\mathbf{S}$ to a planar disk $\mathbf{D}$. (d) Embedding $\mathbf{S}$ to a convex planar polygon $\mathbf{P}$.

that, if the corner angles A, B, C, and D are properly preserved in the mapping process, the distance metric distortion can be effectively relieved, as shown in Fig. 5(d), where $d_{ac\in\mathbf{P}}$ and $d_{ab\in\mathbf{P}}$ remains close in length.

Motivated by this observation, we proposed a variation of the Ricci flow, aiming at providing a mapping from a surface $\mathbf{S}$ to a convex planar polygon $\mathbf{P}$, so as to enable greedy routing on $\mathbf{P}$, as well as to preserve the distance metric on $\mathbf{S}$ as much as possible. As is observed, the sliced surface $\mathbf{M}$ is more like a polygon shape, which is a flexible shape that can serve as a better approximation of the genus-0 open surface $\mathbf{S}$. Also, the intersection angles at the corner points (angle A, B, C, and D in Fig. 5) are critical in preserving the shape of $\mathbf{S}$ in the mapping process. Therefore, in SLICE we propose a variation of the Ricci flow, by setting proper target curvatures for nodes, to map $\mathbf{S}$ to a polygon with its corner points' curvatures preserved to the greatest extent. We present the detail in Section III.

III. SLICE Algorithm

This section presents the SLICE algorithm that deals with a fundamental problem: embedding a network on a high genus 3-D surface into a planar surface as a whole to enable greedy routing, while preserving the distance metric of the original network to the greatest extent in order to reduce the distance distortion. The basic idea behind SLICE is as follows: Slice the genus-n surface $\mathbf{M}$ to a simpler open surface $\mathbf{S}$ for embedding, and leverage a variation of the Ricci flow to flatten $\mathbf{S}$ to a planar convex polygon $\mathbf{P}$. The preprocess of SLICE is to compute a triangulation from the original network via a simple distributed algorithm as in [30] and [31]. The triangulated structure, or *mesh* for short, forms a shape representation of the high genus 3-D surface, as shown in Fig. 1(c). For the ease of presentation, we still call the triangulated mesh as the high genus surface, denoted by $\mathbf{M}$ henceforth. Overall SLICE mainly consists of four steps.

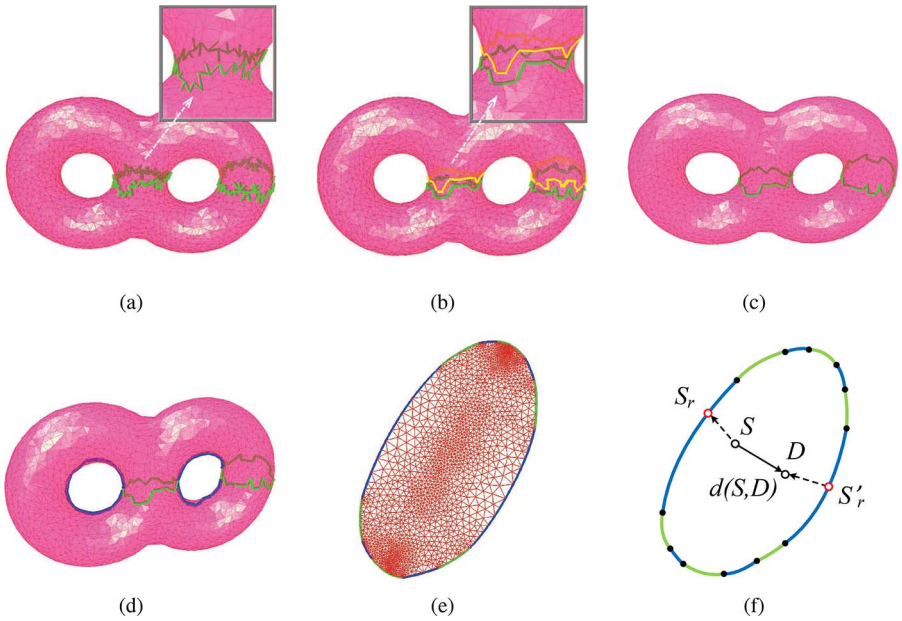


Fig. 6. (a) Cut pairs (green edges) identified on the genus-2 surface $\mathbf{M}$. (b) Disconnecting the cut pairs yields two curves that can be selected as the cut in each loop-end region. (c) Genus-0 surface $\mathbf{G}$ with n cuts; each loop-end region selects one cut. (d) Connecting the cuts and slicing $\mathbf{G}$ to $\mathbf{S}$, a genus-0 surface with exactly one boundary. (e) Convex polygon $\mathbf{P}$ with virtual coordinates. (f) SLICE realizes a variation of greedy routing.

Step 1: We first identify a maximum cut set $\mathcal{C}_{\max}$ of $\mathbf{M}$, which is used to slice $\mathbf{M}$ to a genus-0 surface $\mathbf{G}$ with n cuts. To this end, a flooding across $\mathbf{M}$ is initiated by an arbitrary node r, which implicitly constructs a Morse function by assigning each node a level index. Based on the Morse function, a Reeb graph $\mathbf{R}$ that starts at node r is also constructed in this process, by assigning each node a region ID. Thereafter, all the *loop-end* regions $I_m^i (i = 1, 2, \ldots, n)$ are identified, and next a bisection operation is performed to extract a cut C_i within each I_m^i. All the cuts C_i $(i = 1, 2, \ldots, n)$ form a maximum cut set $\mathcal{C}_{\max}$, slicing the genus-n surface $\mathbf{M}$ to a genus-0 surface $\mathbf{G}$ with n cuts, as illustrated in Fig. 6(c).

Step 2: We further slice $\mathbf{G}$ to an open surface $\mathbf{S}$—a genus-0 surface with exactly one boundary. To do so, we propose to perform a DFS analog that slices open $\mathbf{S}$ by connecting n cuts with $(2n-1)$ slice lines, as illustrated in Fig. 6(d). In this process, if a cut and a slice intersects at a node s_I, s_I then identifies itself to be an *intersection node*, as shown in Fig. 5(a) and (d).

Step 3: By applying a proposed variation of the Ricci flow, $\mathbf{S}$ is flattened into a convex polygon $\mathbf{P}$, with the vertices of the polygon being the *intersection nodes* and the edges of the polygon being the cuts and the slice lines [see Fig. 6(e)]. The packet delivery of greedy routing is guaranteed on such convex polygon $\mathbf{P}$. Moreover, the convex polygon $\mathbf{P}$ can well preserve the distance metric in the original high genus 3-D surface $\mathbf{M}$, enabling a low-stretch routing on $\mathbf{P}$. Every node in the network is then given a virtual coordinate in $\mathbf{P}$, as illustrated in Fig. 6(e).

Step 4: In the virtual coordinate system, it is observed that a source node s can route to a destination node d via two kinds of paths: 1) traversing only the interior nodes; 2) across the boundary $\partial\mathbf{P}$ [see Fig. 6(f)]. In SLICE, the source node s first estimates the distance of the two paths and then chooses a shorter one for routing, thereby achieving a low stretch and load balance greedy routing.

The whole process of SLICE, as an example, is shown in Fig. 6.

A. Extracting a Maximum Cut Set

For a genus-n surface $\mathbf{M}$, the first step is to find a maximum cut set $\mathcal{C}_{\max}$, aiming at slicing $\mathbf{M}$ to a genus-0 surface $\mathbf{G}$ based on the Morse theory and Reeb graph. To this end, we exploit the discrete geometry characteristics of $\mathbf{M}$.

In order to set up the Morse function of $\mathbf{M}$, an arbitrary root node r initiates a flooding across the whole network. By doing so, every node learns its hop distance i to r and records its level index as i. All the nodes with a level index of i belong to level-i of $\mathbf{M}$. A Morse function in discrete settings thus can be defined as a mapping $f(p) \to i$, where p is a node in level-i, and all the nodes in level-i are denoted by $f^{-1}(i)$. We denote the max hop count of nodes from r as I, that is, $i \leq I$. Also, after a message flooded from r reaches a node p, p records its parent from which it receives the message.

To construct a Reeb graph from the defined Morse function, a distributed algorithm similar to that in [18] is carried out, which involves two major steps.

First, the algorithm identifies the nodes in each connected component in $f^{-1}(i)$ (level-i) with a component ID. To this end, a randomly selected node q in $f^{-1}(i)$ claims itself as a landmark and floods within $f^{-1}(i)$ a message containing the node ID of q and its level index i. Then, by a landmark selection process similar to that in [32], each component selects a dominating landmark, denoted by $DM(\cdot)$, with the smallest ID. As such, the nodes in each component in $f^{-1}(i)$ are notified a component ID.

Second, all components should be transformed to *regions* (arcs) of the Reeb graph. This process starts from $f^{-1}(1)$ to $f^{-1}(I)$. We say a component P_i in $f^{-1}(i)$ is connected with a component P_{i+1} in $f^{-1}(i+1)$ if there exists in the component P_i a node p that has a neighbor p' in the component P_{i+1}. Then, the nodes p and p' will notify the dominating landmarks $DM(P_i)$ and $DM(P_{i+1})$ in the components P_i and P_{i+1}, respectively, of this connectivity. According to the Morse theory,

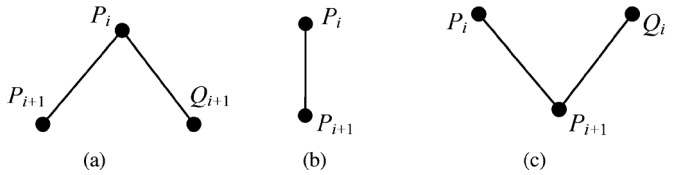


Fig. 7. P_i, Q_i are in level-i; P_{i+1}, Q_{i+1} are in level-$(i+1)$. (a) P_i is connected with P_{i+1} and Q_{i+1}. (b) P_i corresponds to component P_{i+1} only. (c) P_{i+1} is connected with P_i and Q_i.

there only exist three cases related to this connection, as shown in Fig. 7. The dominating landmarks will notify the nodes in the component P_{i+1} if P_{i+1} only corresponds to the component P_i. In this case, the nodes in P_{i+1} are assigned the same *region* ID as the nodes in component P_i; otherwise, the nodes in component P_{i+1} are assigned a new region ID. After this process, every node is notified with a region ID.

Having the Morse function and Reeb graph, all the *loop-end* regions can be identified directly: If the dominating landmark $DM(P_{i+1})$ in P_{i+1} is notified that P_{i+1} is connected with P_i and Q_i in level-i [as shown in Fig. 7(c)], then P_{i+1} and all other components in the same *region* are notified to be in a loop-end region.

Then, to extract the maximum cut set $\mathcal{C}_{\max}$, each loop-end region I_m^i performs a bisection operation to extract a cut C_i as follows. Suppose a loop-end region I_m^i consists of a set of components $\{P_h\}$ where $H_1 \leq h \leq H_2$, then the bisection operation in I_m^i is initiated from P_{H_1} to P_{H_2}. The nodes in P_{H_1} reset its region ID to its parent's region ID. Since P_{H_1} is the first component of I_m^i, it is the neighbor of two components P_i and Q_i [see Fig. 7(c)], and component P_i and Q_i belongs to region I_α and I_β, respectively. The nodes in P_{H_1} change their *region* ID to the region ID of I_α or I_β, respectively. Therefore, P_{H_1} is bisected and assigned to *region* I_α or I_β. This process is carried out from P_{H_1} to P_{H_2}. Consequently, loop-end region I_m^i is bisected and two *newly merged regions* I'_α and I'_β are generated.

Finally, we can obtain a simple cut curve constituted by a set of connected edges. In a continuous domain, disconnecting I'_α and I'_β will generate a simple curve, which is exactly a cut of $\mathbf{M}$. Distinctively, in discrete settings, disconnecting I'_α and I'_β will generate a pair of curves, as illustrated in Fig. 6(b). Either one of them is a cut for $\mathbf{M}$. So the cut C_i is obtained by disconnecting I'_α and I'_β as follows. 1) Each node p in loop-end region I_m will send a message to its neighbor p' in I_m; if p' has a different region ID with p, (p, p') is notified to be a cut pair. 2) By disconnecting all the cut pairs in loop-end region I_m, two cuts $C_{I'_\alpha}$ and $C_{I'_\beta}$ are identified, as shown in Fig. 6(b). Either one of them is selected to be the cut that disconnects I'_α and I'_β. Since there are n loop-end regions I_m in $\mathbf{M}$, there are n cuts identified correspondingly in $\mathbf{M}$.

B. Slicing the Topology Open

Given a genus-0 surface $\mathbf{G}$ with n cuts, which is still complex: It is not homotopically equivalent to any simple planar topology [33]; we propose to further slice $\mathbf{G}$ to $\mathbf{S}$—a genus-0 surface with exactly one boundary, which is homotopically equivalent to a planar convex polygon.

To this end, we propose to perform a DFS analog that slices open $\mathbf{S}$ by connecting n cuts with $(2n-1)$ slice lines, as illustrated in Fig. 6(d). The pipeline of this slicing process is given

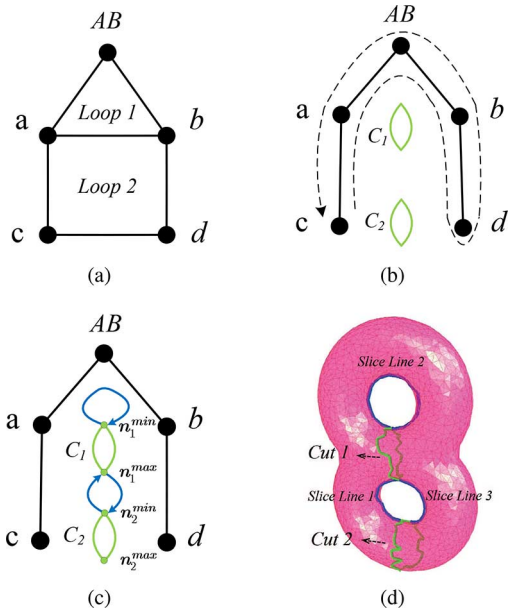


Fig. 8. (a) Connection graph of the Reeb components. (b) Tree structure $\mathbf{T}$. (c) Slicing process by the DFS analog. (d) Slicing result.

in Fig. 8. After the bisection operation, the Reeb graph of $\mathbf{M}$ in Fig. 8(d) can be depicted in Fig. 8(a), where an aforementioned *region* is represented by a node and the links displays the connections between the regions. In each region, one dominating landmark is selected (the process is quite similar to that of selecting a dominating landmark in a component as mentioned before). In Fig. 8(a), there are two loops—Loop 1 and Loop 2. Once a maximum cut set $C_{\max} = (C_1, C_2)$ with two cuts is identified, a and b are disconnected by cut C_1, while c and d are disconnected by cut C_2. As a consequence, Loop 1 and Loop 2 are disconnected, and the Reeb graph of $\mathbf{S}$ is transformed to a tree structure $\mathbf{T}$, as illustrated in Fig. 8(b). It is well known that, given such a tree structure $\mathbf{T}$ that contains many nodes (regions), it can be traversed in a manner analogous to a DFS as shown by the dashed line with arrow in Fig. 8(b). The arrow in Fig. 8(b) illustrates the search order: $c \to a \to AB \to b \to d$.

Before the DFS connecting process, a node with the minimum level index and a node with the maximum level index are identified in each cut C_i in a spontaneous manner: Every node in C_i floods a massage containing its own level index; when a node n_i receives a message with level index LV_m, it compares LV_m to its recorded maximum and minimum level index $LV_i^{\max}$ and $LV_i^{\min}$; if $LV_m > LV_i^{\max}$ or $LV_m < LV_i^{\min}$, node n_i floods the massage and updates its recorded maximum and minimum level index, otherwise the massage is dropped. For example, in Fig. 8(c), minimum-level node $n_1^{\min}$ and maximum-level node $n_1^{\max}$ are identified in cut C_1, and minimum-level node $n_2^{\min}$ and maximum-level node $n_2^{\max}$ are identified in cut C_2.

In practice, the DFS connecting process is initiated by the minimum-level node in the lowest cut in the tree structure $\mathbf{T}$. For example, in Fig. 8(c), it is node $n_2^{\min}$ in C_2 that initiates the clustering procedure. Then, $n_2^{\min}$ notifies the dominating landmark in the region c to start to traverse $\mathbf{T}$. In the process $c \to a$, $n_2^{\min}$ is connected with $n_1^{\max}$, the transverse $c \to a$ is finished, and *Slice Line 1* in Fig. 8(d) is notified. Hence, the dominating landmark in a continues to traverse $\mathbf{T}$. Then, *Slice Line 2* started from $n_1^{\min}$ slices along the route $a \to AB \to b$, eventually

reaching $n_1^{\min}$ again. Finally, *Slice Line 3* from $n_1^{\max}$ to $n_2^{\min}$ is generated in transverse $b \to d$. The local topology is then sliced open according to the slice lines, and a genus-0 surface $\mathbf{S}$ with exactly one boundary is extracted. The slice lines are depicted in blue in Fig. 8(d), and the result of $\mathbf{S}$ is given in Fig. 6(d).

C. Variation of Ricci Flow in Discrete Surface

Given a genus-0 surface $\mathbf{S} = (V, E, F)$ (where V, E, F represent the vertices, edges, and face) with exactly one boundary, a variation of the discrete surface Ricci flow is proposed to embed $\mathbf{S}$ to a convex polygon $\mathbf{P}$.

In discrete settings, the Riemannian metric on $\mathbf{S}$ can be simply defined by the edge lengths on $\mathbf{S}$: $l : E \to \mathbb{R}^+$, such that on each face f_{ijk}, the edge lengths satisfy the triangle inequality: $l_{ij} + l_{jk} > l_{ki}$. The discrete metric determines the corner angles on each face by the cosine law $l_{ij}^2 = l_{jk}^2 + l_{ki}^2 - 2l_{jk}l_{ki}cos\theta_{ij}$. Then, the discrete Gaussian curvature is defined as the angle deficient [29].

Definition 3: The discrete Gaussian curvature K_i is defined as the angle deficit on a mesh, where K_i on a vertex v_i satisfies

$$K_i = \begin{cases} 2\pi - \Sigma_{f_{ijk}\in F}\theta_i^{jk}, & v_i \notin \partial\mathbf{S} \\ \pi - \Sigma_{f_{ijk}\in F}\theta_i^{jk}, & v_i \in \partial\mathbf{S} \end{cases} \tag{2}$$

where θ_i^{jk} represents the corner angle attached to v_i in face $f_{ijk} \in F$, and $\partial\mathbf{S}$ is the boundary of $\mathbf{S}$.

The total Gaussian curvature is controlled by the topology of $\mathbf{S}$, following the Gauss–Bonnet theorem (Theorem 3) in the discrete setting as $\sum_{v_i \in V} K_i = 2\pi\chi(\mathbf{S})$.

To approximate the conformal deformation of metrics in discrete settings, circle packing metric was introduced in [34] and [35]. For a vertex $v_i \in V$, a circle with radius r_i is assigned. A function that assigns a radius r_i to each vertex v_i is denoted as: $\Gamma : V \to \mathbb{R}^+$. A weight function is also defined as $\Phi : E \to [0, \frac{\pi}{2}]$ by assigning a positive weight w_{ij} to each edge e_{ij}. The pair of the vertex radius and the edge weight function on $\mathbf{S}$, (Γ, Φ), is called a *circle packing metric* of $\mathbf{S}$.

Suppose v_i and v_j are the two vertices of edge $e_{ij} \in E$, and the two circles at v_i, v_j with radius r_i, r_j intersect with an acute ϕ_{ij}. Then, the length of e_{ij} is computed as $l_{ij}^2 = r_i^2 + r_j^2 + 2r_ir_jcos\phi_{ij}$.

Definition 4: Suppose $u_i = \log r_i$ for vertex v_i, then the discrete Ricci flow is defined as

$$\frac{du_i(t)}{dt} = (\bar{K}_i - K_i) \tag{3}$$

where K and $\bar{K}_i$ are the current and target Gaussian curvatures at v_i, and t is the evolving time [28].

To deform the initial circle packing metric $\mathbf{S}(\Gamma, \Phi)$ to flatten the surface $\mathbf{S}$ to a convex polygon $\mathbf{P}$, a variation of the discrete surface Ricci flow is proposed to preserve the angles at intersection points to the greatest extent by setting the target curvature $\bar{K}_i$ as

$$\bar{K}_i = \begin{cases} 0, & v_i \notin \partial\mathbf{S} \\ \frac{2\pi - \Sigma\theta_I}{L - L_I}, & v_i \in \partial\mathbf{S}, v_i \notin I \\ \theta_I, & v_i \in \partial\mathbf{S}, v_i \in I \end{cases} \tag{4}$$

where I is the set of intersection points, θ_I is the intersection angles at I, $\Sigma\theta_I$ is the sum of all intersection angles, L is the length of the boundary, and L_I is the length sum of all the boundary edges connected to the intersection points.

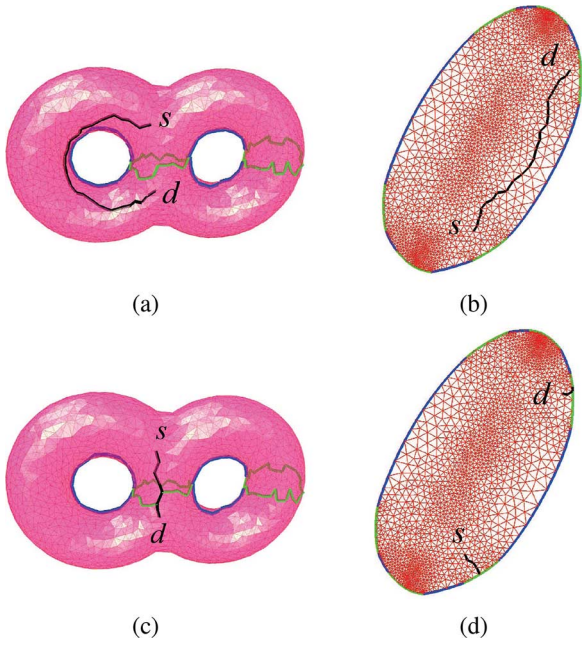


Fig. 9. (a) Greedy routing path in original topology. (b) Greedy routing path in the virtual coordinate system. (c) SLICE routing path in original topology. (d) SLICE routing path in the virtual coordinate system.

In this target curvature setting, the sum of boundary curvatures equals to $\frac{2\pi - \Sigma\theta_I}{L - L_I} \times (L - L_I) + \Sigma\theta_I = 2\pi$. Since the curvature sum of the inner nodes is 0, the discrete Gauss–Bonnet theorem holds.

Finally, every node p is assigned a virtual coordinate in the convex polygon, as shown in Fig. 6(e).

D. Routing Scheme

With the virtual coordinates, intuitively greedy routing can be applied directly on the convex polygon $\mathbf{P}$ with guaranteed delivery. However, there are still two problems. First, a number of local connections are disconnected by the cuts in the embedding process, possibly resulting in long paths between those nodes that are close to each other, as shown in Fig. 9(a); second, packets are more likely to travel through the central part of $\mathbf{P}$, potentially overloading the central nodes.

To solve the above problems, we propose a variation of greedy routing based on the following observation. In the embedding process, a line segment belonging to a cut or a slice line is sliced and embedded into two line segments, as illustrated in Fig. 6(d) and (e). Hence, a real node on a cut or a slice line is assigned with multiple virtual coordinates[1] in $\mathbf{P}$. Suppose a real node S_{real} on a cut or a slice line corresponds to two virtual coordinate (S_r, S_r') in $\mathbf{P}$, as illustrated in Fig. 6(f), and the closest boundary node to the source node s in $\mathbf{P}$ is S_r, then there are only two possible shortest routing paths between the source node s and the destination node d: 1) s routes the packet to d all by interior nodes; and 2) s first routes the packet to the node S_r on the boundary, and then to d from S_r', as shown in Fig. 6(f). Based on this observation, our routing scheme includes the following steps.

[1]An *intersection node* is associated with more than one cuts or slice lines.

First, the boundary nodes of $\mathbf{P}$, denoted by $\partial\mathbf{P}$, initiate flooding. In the process, every node s in $\mathbf{P}$ records a root node S_r, the closest node to s among all nodes on $\partial\mathbf{P}$. Also, the node s records an arbitrary S'_r such that S_r and S'_r are the virtual coordinates for the same real node S_{real}. It is worth noting that every node requires only maintaining a 2-tuple $((S_r, S'_r)$. That is, the storage overhead for local routing decisions is trivial.

Second, if we denote the Euclidean distance between two nodes by the virtual coordinate as $\texttt{dist}(\cdot,\cdot)$, the routing path is then determined as follows: In the virtual coordinate system, the length of the routing path $s \to d$ is estimated by $L_A = \texttt{dist}(s,d)$; the length of the routing path $s \to S_r \to S'_r \to d$ is estimated by $L_B = \texttt{dist}(s,S_r) + \texttt{dist}(S'_r,d)$. Therefore, by finding the minimum value in the set $\{L_A, L_B\}$, the source node s chooses the corresponding route to deliver the packet. Fig. 9 demonstrates an example of routing by SLICE.

IV. Adapting SLICE to High Genus 3-D Surfaces With Holes and High Genus 3-D Volumes

In Section III, SLICE provides a low-stretch delivery-guaranteed routing solution for WSNs on high genus 3-D closed surface (compact and without boundaries). However, in many real tunnel-shape WSNs scenarios, the sensors are not deployed on an ideal 3-D closed surfaces. First, the sensor network topology can be a complex 3-D surface with holes on it: There can be entrances or exits on the surface [Fig. 1(d) and (e)]; the sensors may be deployed only in part of the tunnel surface, rendering holes on the surface. Second, the sensor network topology can be a complex 3-D high genus volume: many applications require the sensors be deployed not only on the surface of the tunnel, but also in its volume, especially for WSN deployed under water or in the air to monitor water (salinity, flow, pressure) or atmosphere (CO_2, wind, humidity) conditions. To deal with the challenges in the above two WSNs scenarios, in this section, we extend SLICE to apply on a wider range of high genus WSNs deployed on surfaces with holes and in 3-D volumes.

A. Adapting SLICE to Surface Networks With Holes

In SLICE, the maximum cut set is generated by identifying the loop-end regions, which requires the network topology be a closed surface $\mathbf{M}$. In other words, $\mathbf{M}$ should be compact and without boundaries. However, as is demonstrated in Fig. 1(f), the network topology may be deployed on a 3-D high genus surface with holes, denoted as $\mathbf{M_h}$. In this case, it is more complex and difficult to slice and embed the network because the holes may change the Reeb graph identified in $\mathbf{M_h}$, and the method in Section III is no longer valid. For example, the Reeb graph of a common genus-1 torus is given in Fig. 3(b), with 4 Reeb components constituting one loop; the corresponding Reeb components are depicted by color in Fig. 3(a). When there are holes on this torus [see Fig. 10(a)], however, the Reeb graph changes to Fig. 10(b), where the number of Reeb components changes to 7, while the loop number changes to 2. In this case, with holes, the loop number of the Reeb graph is not equal to the number of its genus. Therefore, these holes will hinder the process of generating the maximum cut set for $\mathbf{M_h}$.

We tackle this problem on the premise condition that the hole is detected by some boundary detection algorithms, e.g., [36]–[38]. It is observed that the holes change the Reeb graph because the holes on a surface $\mathbf{M_h}$ render $\mathbf{M_h}$ to change from

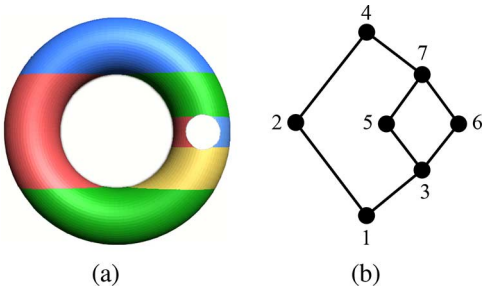


Fig. 10. (a) Color components for the Reeb graph in (b). (b) Reeb graph of a genus-1 torus with holes.

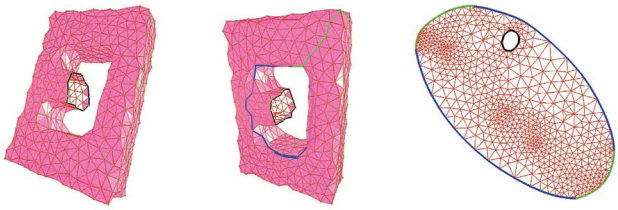


Fig. 11. (a) Genus-1 corridor $\mathbf{M_h}$ with a hole; the hole's boundary edges are colored in black. (b) Open surface $\mathbf{S_h}$ with multiple boundaries, with the cut (green line) and slice line (blue line). (c) Convex polygon $\mathbf{P_h}$, with the hole boundary (black line).

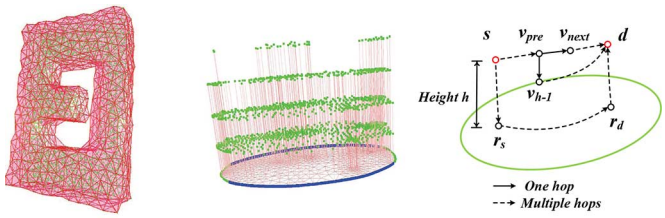
a closed surface to an open surface. Accordingly, we propose a solution by means of "regarding" a hole to be a polygon face of $\mathbf{M_h}$, so that $\mathbf{M_h}$ can still be regarded as a "closed" surface. This is realized by a local flooding constrained within the boundary nodes of each hole during the evolution of the components of the level sets $f^{-1}()$. Consequently, all components in a level set $f^{-1}(i)$ connected to the same hole $\mathbf{H_j}$ will notify each other that they are connected via multiple boundary edges of $\mathbf{H_j}$. Hence, the hole $\mathbf{H_j}$ does not separate any two components in a level set $f^{-1}(i)$. In this way, $\mathbf{M_h}$ will still maintain the Reeb graph of $\mathbf{M}$, and a correct maximum cut set can be generated for $\mathbf{M_h}$, as illustrated in Fig. 11(b). As a result, $\mathbf{M_h}$ is sliced to an open surface $\mathbf{S_h}$ with a set of holes. Accordingly, $\mathbf{S_h}$ has multiple boundaries: One major boundary $\partial\mathbf{S_h^m}$ is constituted by cuts and slice lines [green and blue lines in Fig. 11(b)]; other boundaries are hole boundaries, denoted by $\partial\mathbf{S_h^j}$ [the black line in Fig. 11(b)].

Next, we embed $\mathbf{S_h}$ to a convex polygon $\mathbf{P_h}$ with multiple circular holes, where greedy routing is still allowed. In the Ricci flow and embedding process, the boundary $\partial\mathbf{S_h^m}$ is mapped to the outer boundary of the convex polygon $\mathbf{P_h}$, and the hole boundary $\partial\mathbf{S_h^j}$ is mapped to the circular hole in $\mathbf{P_h}$. To this end, the target curvatures of the inner nodes and the nodes on major boundary $\partial\mathbf{S_h^m}$ are set according to (4). For a node v_i on a hole boundary $\partial\mathbf{S_h^j}$, its target curvature is set as

$$\bar{K}_i = -\frac{2\pi}{L_j}, \quad v_i \in \partial\mathbf{S_h^j} \tag{5}$$

where L_j is the length of $\partial\mathbf{S_h^j}$. Note that a hole boundary $\partial\mathbf{S_h^j}$ is inside the polygon $\mathbf{P_h}$ and is concave for $\mathbf{P_h}$, so the target curvature should be set as negative.

Through the above process, the high genus 3-D surface $\mathbf{M_h}$ with holes is embedded to a planar convex polygon with multiple circular holes on it [see Fig. 11(c)], where our proposed



(a) (b) (c)

Fig. 12. (a) High genus 3-D volume $\mathbf{V}$; its surface is depicted by the red triangulated mesh, and its inner nodes are in green. (b) 3-D height structure $\mathbf{H}$ attached to the convex polygon; the inner nodes are mapped into four levels in this case. (c) Variation of greedy routing for 3-D volume.

greedy routing variation can be applied to enable delivery guaranteed greedy routing.

B. Adapting SLICE to 3-D Volume Networks

Compared to embedding a surface, embedding a volume directly to a simple topology is more challenging, which usually requires tetrahedralization [39] and centralized operations [12]. In Section III, SLICE embeds the high genus 3-D surface to a convex planar polygon. Noting that the surface of a high genus 3-D volume $\mathbf{V}$ is accordingly a high genus 3-D surface $\mathbf{M}$, the virtual coordinates of the surface nodes on the convex planar polygon, generated by SLICE, can serve as a helpful auxiliary structure for routing in $\mathbf{V}$. Hence, SLICE assigns the inner nodes of a 3-D volume to a height structure attached to the surface nodes on the convex planar polygon, in order to provide a variation of greedy routing scheme with guaranteed delivery for high genus 3-D volume WSNs.

To this end, a scheme associating the inner nodes of $\mathbf{V}$ with its surface nodes on $\mathbf{M}$ is proposed, which attains a mapping for the inner nodes of $\mathbf{V}$ to a height structure $\mathbf{H}$. Specifically, a flooding similar to the process of setting up the Morse function in Section III is initiated from the surface nodes on $\mathbf{M}$. During the flooding process, every inner node v of the volume $\mathbf{V}$ records a hop-count distance h_v to v's nearest boundary node v_r on $\mathbf{M}$. h_v is denoted as the *height* of node v, while node v_r on $\mathbf{M}$ is called the *root* of node v. In this way, every inner node now corresponds to one surface node. Since the virtual coordinate of node v_r on $\mathbf{M}$ is on the planar convex polygon $\mathbf{P}$ (on $x-y$ plane), if we consider *height* h_v to be the z axis coordinate for node v, then inner nodes of volume $\mathbf{V}$ is mapped to a 3-D height structure $\mathbf{H}$, as illustrated in Fig. 12(b).

Suppose the source node and the destination node are denoted as s and d; a packet from s to d is denoted as pkt; the height of s and d are denoted as h_s and h_d; the *roots* of s and d are denoted as r_s and r_d; and their virtual coordinates on the polygon $\mathbf{P}$ are denoted as $\mathrm{Coord}(r_s)$ and $\mathrm{Coord}(r_d)$. Then, based on the height structure $\mathbf{H}$, a variation of greedy routing scheme with guaranteed delivery is provided to adapt SLICE for high genus 3-D volumes, with the following steps

Step 1 (Greedy): Suppose packet pkt has reached an intermediate node v_{pre} with height $h_{v_{\mathrm{pre}}}$, v_{pre} will then flood a massage to all its one-hop neighbors (in its one-hop neighbor set V_n). Suppose $v_n \in V_n$ is one of v_{pre}'s neighbors with height $h_{v_{\mathrm{pre}}}$: 1) If v_n is closest to d in set V_n, and v_n is closer to d than v_{pre}, i.e., $\mathtt{dist}(v_n, d) = min(\mathtt{dist}(V_n, d))$ and $\mathtt{dist}(v_n, d) < \mathtt{dist}(v_{\mathrm{pre}}, d)$, then v_n is selected as the next node in the routing path, denoted as v_{next}, while v_{pre} sends the packet pkt to v_{next}. 2) If no such v_{next} is identified, and $\mathrm{Coord}(r_{v_{\mathrm{pre}}}) \neq \mathrm{Coord}(r_{v_d})$, then we turn to Step 2. 3) if no such v_{next} is identified, and $\mathrm{Coord}(r_{v_{\mathrm{pre}}}) = \mathrm{Coord}(r_{v_d})$, then to Step 3.

Note that the distance $\mathtt{dist}(s, d)$ here is the Euclidean distance between $\mathrm{Coord}(r_s)$ and $\mathrm{Coord}(r_d)$ $(z = 0)$.

Step 2 (Descending Height): v_{pre} decreases its height $h_{v_{\mathrm{pre}}}$ by 1, and sends pkt to a node v_{h-1} with a coordinate equal to $\mathrm{Coord}(v_{\mathrm{pre}})$ and a *height* equal to $(h_{v_{\mathrm{pre}}} - 1)$. Set v_{h-1} as v_{pre} and turn to Step 1.

Step 3 (Reaching Destination): In this case, if $h_{v_{\mathrm{pre}}} = h_{v_d}$, then the destination d is reached. If $h_{v_{\mathrm{pre}}} > h_{v_d}$, then each time the height of v_{pre} decreases by 1 until $h_{v_{\mathrm{pre}}} = h_{v_d}$, so that destination d is reached. If $h_{v_{\mathrm{pre}}} < h_{v_d}$, then each time the height of v_{pre} increases by 1 until $h_{v_{\mathrm{pre}}} = h_{v_d}$, so that destination d is reached.

This process continues until the packet pkt reaches the destination d. The height structure $\mathbf{H}$ and the virtual coordinates on the convex polygon $\mathbf{P}$ guarantee the packet delivery in the routing process, as depicted in Fig. 12(c).

V. Discussions

In this section, we discuss the time complexity, the message complexity, and the storage cost of SLICE, which are important factors for realizing a scalable and distributed routing algorithm in WSNs. We also briefly discuss recent methods of how to cut a high genus surface to a topological disk.

A. Time Complexity and Message Complexity

For the time complexity and the message complexity, we have the following theorem.

Theorem 4: For a network with nodes roughly uniformly distributed over the sensing area, the time complexity and the message complexity of SLICE are $O(m)$ and $O(m{\cdot}n)$, respectively, where m is the network size, and n is the genus of the network.

Proof: As the nodes are roughly uniformly distributed in the network, the diameter of a 2-D network is $\sqrt{m}$, while the diameter of a 3-D volume network is $\sqrt{[3]m}$. Hence, the diameter of a 3-D surface network is between $\sqrt{[3]m}$ and $\sqrt{m}$.

SLICE includes the following steps.

Step 1) Generating a maximum cut set. This step includes: a flooding across 3-D high genus surface $\mathbf{M}$ initiated by an arbitrary node with $O(\sqrt{m})$ time complexity and $O(m)$ message complexity; the Morse function and Reeb graph construction, as well as all the loop-end regions' identification, with $O(\sqrt{m})$ time complexity and $O(m \cdot n)$ message complexity; all the bisection operations with $O(\sqrt{m} \cdot n)$ time complexity and $O(m \cdot n)$ message complexity.

Step 2) Slicing $\mathbf{M}$ to a genus-0 open surface $\mathbf{S}$. As the DFS analog takes no more than $(2m - 1)$ times, both the time and message complexity are $O(m)$.

Step 3) Embedding the surface to a convex polygon by the Ricci flow. Both the time and message complexity of this process are $O(m \cdot k)$, where k is the iteration times determined by the convergence time of the Ricci flow. Since the Ricci flow converges with an exponential speed [7], k is a small constant. To sum up, the time and message complexity of SLICE is $O(m)$ and $O(m \cdot n)$, respectively.

■

TABLE I
STORAGE COST COMPARISON

Samples	Genus-2 Torus	Genus-4 Torus	Genus-8 Torus
High-Genus	20	32	56
SLICE	16	16	16

B. Storage Cost

Storage cost is another aspect that is important for scalable and distributed routing algorithms in WSNs. The storage of a sensor should be small and of constant size to avoid dynamic memory allocation. For instance, TinyOS, an operating system running on many types of sensor nodes, does not provide dynamic memory allocation. Consequently, the number of messages for which a node may store the state needs to be determined at the compile time.

For the storage cost of SLICE, as any sensor delivering a packet only needs to store a 2-tuple <root coordinate 1, root coordinate 2>, it guarantees a small and constant size storage. The following is a comparison of the storage cost of the High-Genus algorithm [4] and our proposed SLICE.

In the High-Genus, a node needs to store its pant piece ID (1 bit), its virtual coordinate (x, y, z, 3 bits), 6 boundaries between its own and the adjacent pieces (6 bits), the hyperbolic distances from itself to the 6 boundaries (6 bits), and the most costly, the global adjacency graph of pant pieces, which includes at least $(6n-8)$ connections ($(6n-8)$ bits) for a genus-n network. For example, the High-Genus generates 4 components for a genus-2 torus, and the global adjacency graph includes 4 connections (4 bits).

On the contrary, SLICE only requires a node ID (1 bit), a virtual coordinate (x, y, z, 3 bits), the root coordinate 1, and root coordinate 2 (12 bits). The results of their storage comparisons are given in Table I.

C. Alternative Cutting Methods

In computational geometry and algebra topology communities, a basic tool to deal with high genus topological surfaces is the notion of decomposition. The most useful type of decomposition is the so called *cut graph* [40], whose removal transforms the high genus surface into a topological disk. The combination of the cut line and slice line of our cutting method to slice the surface network open is actually a cut graph of the network.

We also notice that there is an alternative cutting method in a recent proposal [41], which is distributed and connectivity-based. However, as [41] focuses on applying the cut graph for in-network information processing, storage, and retrieval, the performance as well as the theoretical foundation of the cut graph computing are not its concentrations. In addition, the output of the cut graph based cutting method in [41] is a cut graph along which the surface is cut open to a topological disk, so this method can not be used to identify the genus. For SLICE, there are two steps to cut open the surface: The first step is to find the cuts to obtain a genus-0 surface, and the second step is to connect the cuts to obtain the cut graph. During the first step, genus can be identified where it is, and this may be useful for the follow-up studies on high genus 3-D networks, e.g., a recent study [42].

VI. PERFORMANCE EVALUATIONS

We have implemented a simulator and conducted a series of simulations on various 3-D topologies. In the simulations, we presume a perfect link between a pair of nodes, as many recent connectivity-based algorithms [13], [18], [36]–[39] in WSNs did. In fact, when taking into account the unreliable link, SLICE can be robust against the packet loss by simply using an Automatic Repeat Query (ARQ) mechanism to achieve reliable data transmission: A node will (re)transmit a message by means of either unicast or broadcast to its neighbor until it receives an acknowledgment from its neighbor before the timeout. By doing so, there is minor impact on the routing performance, e.g., path selection and routing stretch, of SLICE, though unavoidable extra message cost due to retransmissions will be induced to compensate the unreliable communication.

Simulation results presented in Fig. 13 depict four 3-D topologies—a genus-1 corridor with 710 nodes, a genus-2 bowknot with 837 nodes, a genus-3 smile with 1102 nodes, and a genus-4 window with 5429 nodes, with an average degree from 8.92 to 10.01. It is observed that despite the variation in the scale and complexity of the networks, SLICE extracts appropriate maximum cut sets for the networks by which it embeds them to the planar convex polygons, which guarantees packet delivery between any pair of nodes. What is more, SLICE generates more perceptible "cuts" for large-scale sensor networks [see Fig. 13(g) and (h)], as they are closer to smooth surfaces.

The delivery rate of Greedy routing is 65% for Corridor, 76% for Bowknot, 70% for Smile, and 81% for Window, while SLICE constantly produces a delivery rate of 100%. Therefore, we focus on the following three factors for routing performance evaluation: distance distortion, routing stretch, and load balance. For comparisons, we also implemented two other routing algorithms for high genus 3-D surfaces—High-Genus in [4] and Random-Walk in [43], except for our previous algorithm SINUS in [1].

Distance Distortion: The distance distortion is an important factor that would influence the routing stretch for graph embedding algorithms. In this part, we evaluate the distance distortion of SLICE and compare it to SINUS. Note that High-Genus and Random-Walk are not comparable as they are not routing schemes using embedding techniques. To evaluate the distance distortion, we "glue" 50×50 uniformly distributed black and white rectangles on the embedded planar surface (the convex polygon in SLICE or the annulus in SINUS). Then, this texture is converted back to the original 3-D topology. If the black and white rectangles are also evenly distributed in the original topology, the distance distortion of the mapping algorithm is small; otherwise the distance distortion is large. Fig. 14 shows the results of this texture conversion for SLICE and SINUS.

It is observed that SLICE yields a more evenly distributed conversed texture result than SINUS, indicating a smaller distance distortion of SLICE. In the corner of Fig. 14(a) and (b) and the center of Fig. 14(c), there are large distance distortions of SINUS, while SLICE maintains a relatively even texture distribution. As mentioned before, the smaller distance distortion of SLICE benefits from the proposed variation of the Ricci flow, which approximates the original distance metric by embedding to a convex polygon, a better approximation in shape to preserve the distance metric.

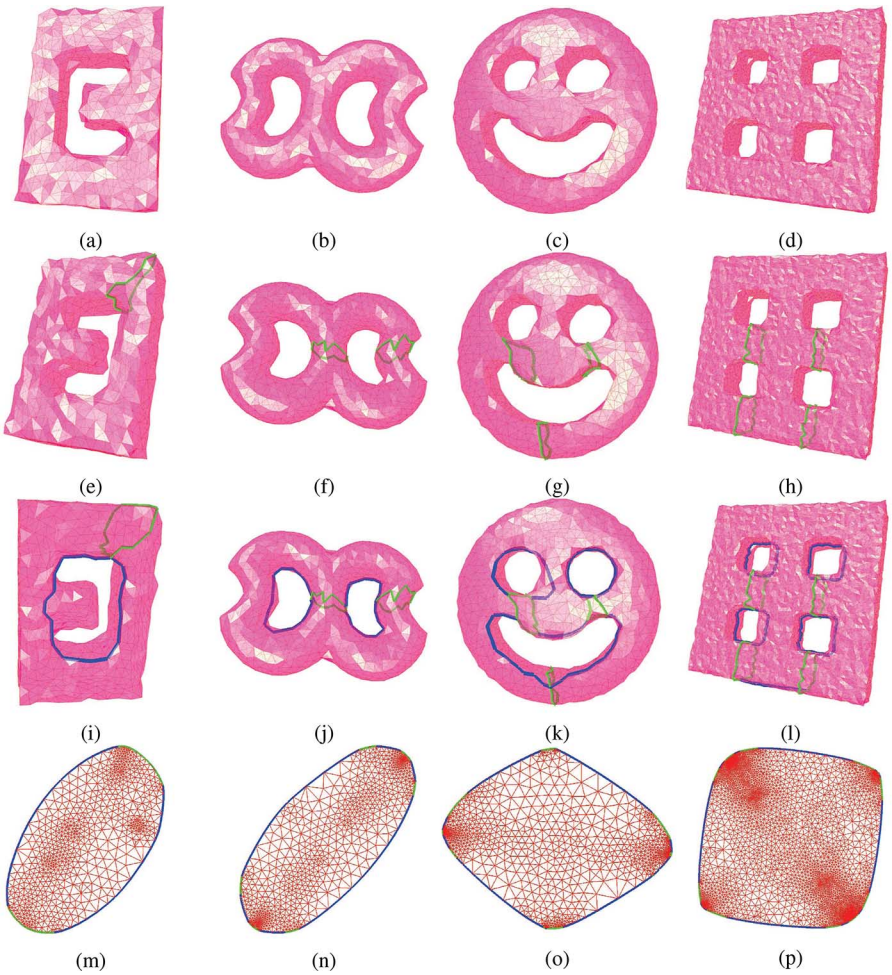


Fig. 13. Columns from left to right: (a) Genus-1 corridor network with 710 nodes; avg deg is 8.92. (b) Genus-2 bowknot network with 837 nodes; avg deg is 9.35. (c) Genus-3 smile network with 1102 nodes; avg deg is 10.01. (d) Genus-4 window network with 5429 nodes; avg deg is 9.74. Rows: 1) Original network. 2) Genus-0 surface $\mathbf{G}$ with n cuts. (3) Genus-0 open surface $\mathbf{S}$ with exactly one boundary. (4) Convex polygon $\mathbf{P}$.

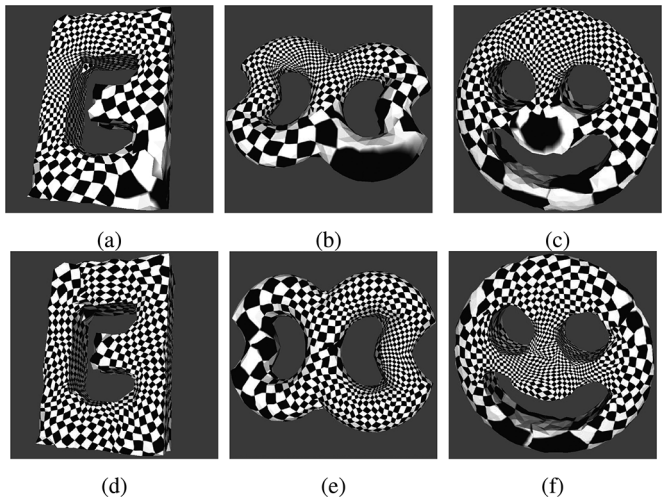


Fig. 14. (a)–(c) SINUS converted texture results. (d)–(f) SLICE converted texture results.

Routing Stretch: The routing stretch for a pair of nodes (s, d) is the ratio of the generated routing path length to the shortest path length between s and d. In our simulation, 10 000 pairs of nodes are randomly selected to calculate the average routing stretch for each network, as well as the worst-case stretch (i.e., the average stretch of the largest 5% stretches), shown in Fig. 15(a). It is observed that SLICE yields a much smaller average routing stretch [1.18, the average of the four topologies in Fig. 15(a)] than SINUS (1.30), High-Genus (1.86), and Random-Walk (1.94). This means SLICE has the highest routing efficiency, as the stretch is an indicator of routing overhead. For example, an average stretch factor of 1.86 in High-Genus indicates its overhead of 86% on the basis of the shortest path. Therefore, SLICE reduces the routing overhead by 12%, 68%, and 76%, in comparison to SINUS, High-Genus, and Random-Walk, respectively. Besides, SLICE has a much smaller routing stretch for Corridor with one hole (1.24) and Corridor volume (1.45), compared to Random-Walk (resp. 2.08 and 2.12), while SINUS and High-Genus do not work on surfaces with holes or in 3-D volumes.

It is noted that, for the topology of the genus-1 corridor, the routing stretch of High-Genus is close to SLICE. This is because for genus-1 surfaces, High-Genus maps the network into a standard genus-1 torus without introducing network decomposition. When decomposition is introduced for High-Genus, that is, when the topology is genus-2 or higher, the stretch of High-Genus increases significantly. The reason is that the adjacency graph of High-Genus does not consider the size of the components. When routing between different components, a

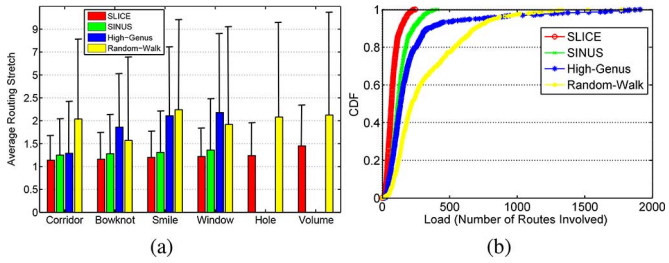


Fig. 15. (a) Routing stretch (Hole stands for Corridor with a hole in Fig. 11(a), Volume for Corridor volume in Fig. 12(a); SINUS and High-Genus do not work on these two topologies). (b) Load distribution.

packet may travel through a series of large components even if there exists a much shorter path in small components.

It is also observed that Random-Walk has larger stretches in Corridor and Smile than in Bowknot and Window, where Random-Walk has to traverse the nodes in the concave region. When the concave region is large (say Corridor and Smile), the stretch increases notably. In contrast to High-Genus and Random-Walk, SLICE keeps a small routing stretch despite the diversity in the genus or topology concavity since it embeds the topology as a whole with the concave region flattened, effectively reducing the distance distortion in the embedding process.

Note that SLICE is slightly superior to SINUS in terms of the average routing stretch because the distance distortions are not that large in most regions of the networks, and therefore the average routing stretches are close to 1.0 for both SINUS and SLICE. For the worst-case stretch, however, SLICE performs much better than SINUS, as SLICE has relatively smaller distance distortions in those regions where the distance distortions of SINUS are large, e.g., the corner of Fig. 14(a) and (b) and the center of Fig. 14(c).

Load Balance: For algorithms that enable greedy routing by embedding, such as [3] and [7], there exists a critical issue—the nodes on the embedded hole boundary tend to suffer a higher traffic load. In SLICE, the case that the packet is routed between nonneighbor boundary nodes has been considered. Therefore, SLICE achieves better load balance, and its boundary nodes are not overloaded. We simulated the traffic load of SLICE, SINUS, High-Genus, and Random-Walk, with randomly selected 10 000 routes from the four tested topologies. Fig. 15(b) shows the cumulative distribution function (CDF) of the load of nodes, which is measured by the number of routes involved.

It is observed that, in SLICE, all the nodes evolve a number of routes less than 280, and its CDF rapidly increases to 1, which means SLICE generates few overloaded nodes and the nodes involve relatively fewer routes in SLICE than in other three schemes. For Random-Walk, it has to traverse the concave region within a certain radius. Therefore, the nodes of the concave region attract a lot of traffic and may be overloaded. As for High-Genus, the boundary nodes in each region may be overloaded, since the six boundaries of a component [4] are concave, so routing within each region tends to route along these concave boundaries. For SINUS, it embeds a surface of any shape to an annulus, resulting in large distance distortion as well as high routing stretch. In contrast, SLICE achieves a better load balance with the concave region flattened and the distance metric well preserved by mapping the surface to a convex polygon.

VII. Conclusion

We have presented SLICE, a novel scalable and distributed routing algorithm with guaranteed delivery for sensor networks on high genus 3-D surfaces. By slicing the high genus surface to a simpler one for embedding, SLICE finally realizes a variation of greedy routing with a low stretch. The proposed algorithm is appealing as it has the best performance with respect to routing stretch and load balance for routing on high genus surfaces. Plus, SLICE can be easily adapted to high genus 3-D surface networks with holes and high genus 3-D volume networks. Finally, the proposed algorithm requires the connectivity information only. We have demonstrated the effectiveness of SLICE through extensive simulations.

One limitation of SLICE is that it relies on a triangular form of the original network. However, current triangulation algorithms for WSNs with mere connectivity information [30], [31] require that network nodes be relatively evenly distributed for a reasonable accuracy when the distance is estimated by hop count. When the network size is small, or the node density is quite nonuniform, the performance of network triangulation as well as SLICE will be affected. Thus, in terms of the future work, we plan to explore triangulation techniques with more general assumptions for our algorithm. We are also interested in how SLICE can incorporate with other routing algorithms to further offer a complete solution for 3-D WSNs with arbitrary shapes.

References

[1] T. Yu *et al.*, "SINUS: A scalable and distributed routing algorithm with guaranteed delivery for WSNs on high genus 3D surfaces," in *Proc. IEEE INFOCOM*, 2013, pp. 2175–2183.

[2] M. Li and Y. Liu, "Underground coal mine monitoring with wireless sensor networks," *Trans. Sensor Netw.*, vol. 5, no. 2, pp. 10:1–10:29, 2009.

[3] S. Xia, X. Yin, H. Wu, M. Jin, and X. D. Gu, "Deterministic greedy routing with guaranteed delivery in 3D wireless sensor networks," in *Proc. ACM MobiHoc*, 2011, pp. 1–10.

[4] X. Yu, X. Yin, W. Han, J. Gao, and X. Gu, "Scalable routing in 3D high genus sensor networks using graph embedding," in *Proc. IEEE INFOCOM*, 2012, pp. 2681–2685.

[5] B. Karp and H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proc. ACM MobiCom*, 2000, pp. 243–254.

[6] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica, "Geographic routing without location information," in *Proc. ACM MobiCom*, 2003, pp. 96–108.

[7] R. Sarkar, X. Yin, J. Gao, F. Luo, and X. D. Gu, "Greedy routing with guaranteed delivery using Ricci flows," in *Proc. IEEE IPSN*, 2009, pp. 121–132.

[8] P. Bose, P. Morin, I. Stojmenovi, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," in *Proc. ACM DIAL-M*, 1999, pp. 48–55.

[9] S. Durocher, D. Kirkpatrick, and L. Narayanan, "On routing with guaranteed delivery in three-dimensional ad hoc wireless networks," in *Proc. ICDCN*, 2008, pp. 546–557.

[10] C. Liu and J. Wu, "Efficient geometric routing in three dimensional ad hoc networks," in *Proc. IEEE INFOCOM*, 2009, pp. 2751–2755.

[11] J. Zhou, Y. Chen, B. Leong, and P. S. Sundaramoorthy, "Practical 3D geographic routing for wireless sensor networks," in *Proc. 8th ACM SenSys*, 2010, pp. 337–350.

[12] S. S. Lam and Q. Chen, "Geographic routing in *d*-dimensional spaces with guaranteed delivery and low stretch," *IEEE/ACM Trans. Netw.*, vol. 21, no. 2, pp. 663–677, Apr. 2013.

[13] S. Xia, M. Jin, H. Wu, and H. Zhou, "Bubble routing: A scalable algorithm with guaranteed delivery in 3D sensor networks," in *Proc. IEEE SECON*, 2012, pp. 245–253.

[14] A. Gray, E. Abbena, and S. Salamon, *Modern Differential Geometry of Curves and Surfaces With Mathematica*. London, U.K.: Chapman & Hall-CRC, 2006.

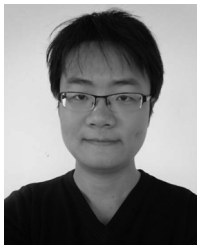
[15] Q. Fang, J. Gao, L. J. Guibas, V. de Silva, and L. Zhang, "GLIDER: Gradient landmark-based distributed routing for sensor networks," in *Proc. IEEE INFOCOM*, 2005, pp. 339–350.
[16] J. R. Munkres, *Elements of Algebraic Topology*. Reading, MA, USA: Addison-Wesley, 1984.
[17] A. Fomenko and T. Kunii, *Topological Modeling for Visualization*. New York, NY, USA: Springer-Verlag Telos, 1997.
[18] H. Jiang, T. Yu, C. Tian, G. Tan, and C. Wang, "CONSEL: Connectivity-based segmentation in large-scale 2D/3D sensor networks," in *Proc. IEEE INFOCOM*, 2012, pp. 2086–2094.
[19] J. Milnor, *Morse Theory*. Princeton, NJ, USA: Princeton Univ. Press, 1963.
[20] K. Cole-McLaughlin, H. Edelsbrunner, J. Harer, V. Natarajan, and V. Pascucci, "Loops in Reeb graphs of 2-manifolds," in *Proc. ACM SoCG*, 2003, pp. 344–350.
[21] R. S. Hamilton, "Three-manifolds with positive Ricci curvature," *J. Diff. Geometry*, vol. 17, no. 2, pp. 255–306, 1982.
[22] G. Perelman, "The entropy formula for the Ricci flow and its geometric applications," arXiv Preprint math/0211159, 2002.
[23] G. Perelman, "Ricci flow with surgery on three-manifolds," arXiv Preprint math/0303109, 2003.
[24] G. Perelman, "Finite extinction time for the solutions to the Ricci flow on certain three-manifolds," arXiv Preprint math/0307245, 2003.
[25] R. S. Hamilton, "The Ricci flow on surfaces," *Contemporary Math.*, vol. 71, no. 1, pp. 237–262, 1988.
[26] B. Chow and F. Luo, "Combinatorial Ricci flows on surfaces," *J. Diff. Geometry*, vol. 63, no. 1, pp. 55–68, 2003.
[27] S. Li, W. Zeng, D. Zhou, D. X. Gu, and J. Gao, "Compact conformal map for greedy routing in wireless mobile sensor networks," in *Proc. IEEE INFOCOM*, 2013, pp. 1409–1417.
[28] M. Jin, J. Kim, F. Luo, and X. Gu, "Discrete surface ricci flow," *IEEE Trans. Vis. Comput. Graphics*, vol. 14, no. 5, pp. 1030–1043, 2008.
[29] M. Jin, J. Kim, and X. D. Gu, *Discrete Surface Ricci Flow: Theory and Applications*. New York, NY, USA: Springer, 2007, pp. 209–232.
[30] S. Funke and N. Milosavljevic, "Network sketching or: "how much geometry hides in connectivity?—Part II"," in *Proc. ACM-SIAM SODA*, 2007, pp. 958–967.
[31] H. Zhou, H. Wu, S. Xia, M. Jin, and N. Ding, "A distributed triangulation algorithm for wireless sensor networks on 2D and 3D surface," in *Proc. IEEE INFOCOM*, 2011, pp. 1053–1061.
[32] A. Nguyen, N. Milosavljevic, Q. Fang, J. Gao, and L. J. Guibas, "Landmark selection and greedy landmark-descent routing for sensor networks," in *Proc. IEEE INFOCOM*, 2007, pp. 661–669.
[33] A. A. Kosinski, *Differential Manifolds*. New York, NY, USA: Dover, 2007.
[34] W. P. Thurston and J. W. Milnor, *The Geometry and Topology of Three-Manifolds*. Princeton, NJ, USA: Princeton Univ. Press, 1979.
[35] K. Stephenson, *Introduction to Circle Packing: The Theory of Discrete Analytic Functions*. Cambridge, U.K.: Cambridge Univ. Press, 2005.
[36] F. Li, C. Zhang, J. Luo, S.-Q. Xin, and Y. He, "LBDP: Localized boundary detection and parametrization for 3-D sensor networks," *IEEE/ACM Trans. Netw.*, vol. 22, no. 2, pp. 567–579, Apr. 2014.
[37] H. Jiang, S. Zhang, G. Tan, and C. Wang, "Connectivity-based boundary extraction of large-scale 3D sensor networks: Algorithm and applications," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 4, pp. 908–918, Apr. 2014.
[38] H. Zhou, H. Wu, and M. Jin, "A robust boundary detection algorithm based on connectivity only for 3D wireless sensor networks," in *Proc. IEEE INFOCOM*, 2012, pp. 1602–1610.
[39] Y. Zhao *et al.*, "Cut-and-sew: a distributed autonomous localization algorithm for 3D surface wireless sensor networks," in *Proc. ACM MobiHoc*, 2013, pp. 69–78.
[40] A. Hatcher, *Algebraic Topology*. Cambridge, U.K.: Cambridge Univ. Press, 2002.
[41] Y. Yang, M. Jin, Y. Zhao, and H. Wu, "Cut graph based information storage and retrieval in 3D sensor networks with general topology," in *Proc. IEEE INFOCOM*, 2013, pp. 465–469.
[42] C. Wang and H. Jiang, "SURF: A connectivity-based space filling curve construction algorithm in high genus 3D surface WSNs," in *Proc. IEEE INFOCOM*, 2015, pp. 981–989.
[43] R. Flury and R. Wattenhofer, "Randomized 3D geographic routing," in *Proc. IEEE INFOCOM*, 2008, pp. 834–842.

Chen Wang (M'13) received the B.S. and Ph.D. degrees in automation from Wuhan University, Wuhan, China, in 2008 and 2013, respectively.

He is currently a Postdoctoral Fellow with Huazhong University of Science and Technology, Wuhan, China. His recent research interests include wireless networking and communication protocols, especially geometric algorithms for wireless sensor networks.

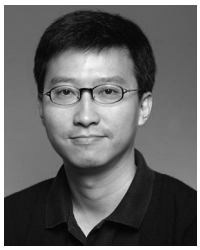
Hongbo Jiang (M'08–SM'14) received the B.S. and M.S. degrees from Huazhong University of Science and Technology, Wuhan, China, in 1999 and 2002, respectively, and the Ph.D. degree from Case Western Reserve University, Cleveland, OH, USA, in 2008, all in computer science.

After that, he joined the faculty of Huazhong University of Science and Technology, where he is now a Full Professor and the Dean of the Department of Communication Engineering. His research concerns computer networking, especially algorithms and protocols for wireless and mobile networks.



Tianlong Yu (S'11) received the B.S. degree in computer science and M.S. degree in electronics and information engineering from Huazhong University of Science and Technology, Wuhan, China, and is currently pursuing the Ph.D. degree in computer science at Carnegie Mellon University, Pittsburgh, PA, USA.

His current research areas include wireless networking.



John C. S. Lui (M'93–SM'02–F'10) was born in Hong Kong. He received the Ph.D. degree in computer science from the University of California, Los Angeles, CA, USA, in 1992.

He is currently the Choh-Ming Li Professor with the Department of Computer Science and Engineering, The Chinese University of Hong Kong (CUHK), Hong Kong. He was the Chairman of the department from 2005 to 2011. His current research interests are in communication networks, network/system security (e.g., cloud security, mobile security, etc.), network economics, network sciences (e.g., online social networks, information spreading, etc.), cloud computing, large-scale distributed systems, and performance evaluation theory.

Prof. Lui is an elected member of the IFIP WG 7.3, a Fellow of the Association for Computing Machinery (ACM), a Senior Research Fellow of the Croucher Foundation, and is currently the Chair of the ACM SIGMETRICS. He has been serving in the Editorial Board of the IEEE/ACM TRANSACTIONS ON NETWORKING, IEEE TRANSACTIONS ON COMPUTERS, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, *Performance Evaluation*, and the *International Journal of Network Security*. He received various departmental teaching awards and the CUHK Vice-Chancellors Exemplary Teaching Award. He is also a co-recipient of the Best Paper Award in the IFIP WG 7.3 Performance 2005, IEEE/IFIP NOMS 2006, and SIMPLEX 2013.