Deterministic Greedy Routing with Guaranteed Delivery in 3D Wireless Sensor Networks

Su Xia
Center for Advanced Computer Studies
University of Louisiana Lafayette, LA
sxx1110@cacs.louisiana.edu

Xiaotian Yin
Mathematics Department
Harvard University
Cambridge, MA
xyin@math.harvard.edu

Hongyi Wu
Center for Advanced Computer Studies
University of Louisiana Lafayette, LA
wu@cacs.louisiana.edu

Miao Jin
Center for Advanced Computer Studies
University of Louisiana Lafayette, LA
mjin@cacs.louisiana.edu

Xianfeng David Gu
Department of Computer Science
Stony Brook University
Stony Brook, NY
gu@cs.sunysb.edu

ABSTRACT

With both computational complexity and storage space bounded by a small constant, greedy routing is recognized as an appealing approach to support scalable routing in wireless sensor networks. However, significant challenges have been encountered in extending greedy routing from 2D to 3D space. In this research we develop decentralized solutions to achieve greedy routing in 3D sensor networks. Our proposed approach is based on a unit tetrahedron cell (UTC) mesh structure. We propose a distributed algorithm to realize volumetric harmonic mapping of the UTC mesh under spherical boundary condition. It is a one-to-one map that yields virtual coordinates for each node in the network. Since a boundary has been mapped to a sphere, node-based greedy routing is always successful thereon. At the same time, we exploit the UTC mesh to develop a face-based greedy routing algorithm, and prove its success at internal nodes. To deliver a data packet to its destination, face-based and node-based greedy routing algorithms are employed alternately at internal and boundary UTCs, respectively. As far as we know, this is the first work that realizes truly deterministic greedy routing with constant-bounded storage and computation in 3D wireless sensor networks.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols—*Routing Protocols*

General Terms

Algorithms, Design, Theory

Mobihoc '11 Paris, France

Keywords

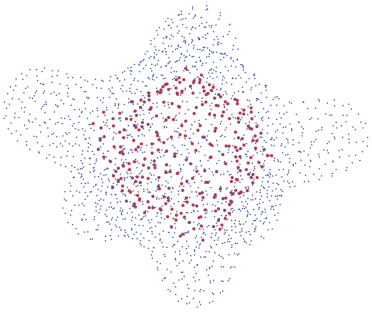
3D sensor networks, harmonic mapping, routing

1. INTRODUCTION

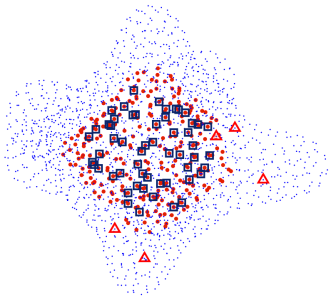
With both of its computation complexity and storage space bounded by a small constant, greedy routing is known for its scalability to large networks with stringent resource constraints on individual nodes. Under most greedy routing algorithms, a node makes its routing decision by standard distance calculation based on a small set of local coordinates only. Such salient property is imperatively needed in the emerging 3D sensor network [1–12], where the problem in routing scalability is greatly exacerbated in comparison with its 2D counterpart, due to dramatically increased sensor nodes in order to cover a 3D space.

The conventional greedy routing algorithms are node-based [13, 14]. More specifically, a node always forwards a packet to one of its neighbors, which is the closest to the destination of the packet. However, such greedy forwarding is not always achievable. A node is called a local minimum if it is not the destination but closer to the destination than all of its neighbors. Clearly, greedy routing fails at the local minimum. Such local minimums may appear at either boundary or internal nodes (as highlighted in red in Fig. 1(b)). A node on a boundary, especially a concave boundary, usually becomes a local minimum when the source and destination nodes are located on two sides of the boundary. Although it seems anti-intuitive, an internal node can be a local minimum too, due to local concavity under random deployment of sensor nodes.

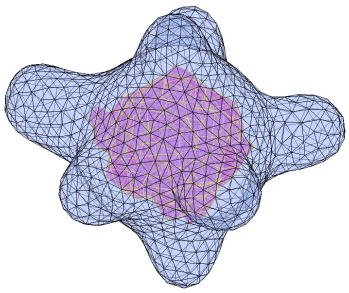
Various approaches have been developed to address the problem of local minimum in 2D networks, with primary focus on boundaries. For example, face routing and its alternatives and enhancements [13–20] exploit the fact that a concave void in a 2D planar network is a face with a simple line boundary. Thus when a packet reaches a local minimum on a boundary, it employs a local deterministic algorithm to search the boundary in either clockwise or counter-clockwise direction as shown in Fig. 2(a), until greedy forwarding is achievable. In a 3D network, however, a void is no longer a face. Its boundary becomes a surface, yielding an arbitrarily large number of possible paths to be explored (see Fig. 2(b)) and thus rendering face routing infeasible. On the other hand, greedy embedding [21–27] provides theoretically sound solutions to en-



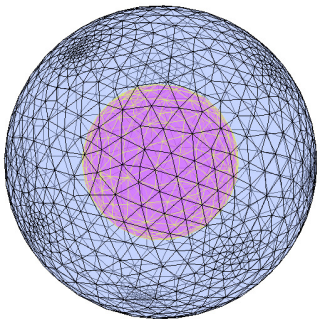
(a) A 3D sensor network (Network model 1).



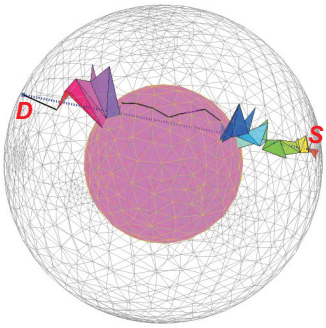
(b) Local minimums in nodal greedy routing.



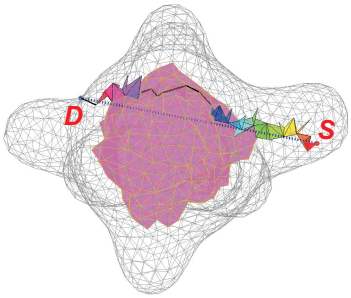
(c) Unit tetrahedron cells (UTCs).



(d) Volumetric Harmonic mapping.



(e) A greedy routing path in mapped domain.



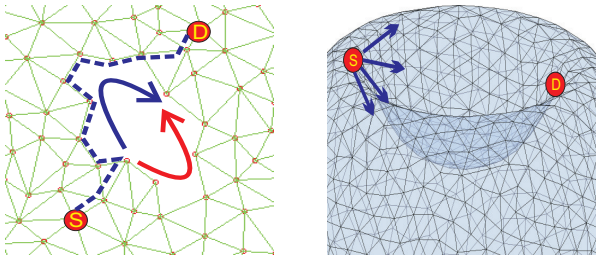
(f) A greedy routing path in original network.

Figure 1: Illustration of the proposed greedy routing protocol. (a) A 3D sensor network that has irregular outer and inner boundaries and consists of about 2,000 nodes. This is one of the network models used on our simulation (see Fig. 6 for other network models). The nodes on the inner boundary are highlighted in red. (b) The nodes that are local minimums under node-based greedy routing are highlighted as blue squares and red triangles, for boundary nodes and internal nodes, respectively. (c) The established unit tetrahedron cells (UTCs). (d) The result after volumetric Harmonic mapping, with both outer and inner boundaries mapped to spheres. (e) A greedy routing path shown in virtual coordinates created by volumetric Harmonic mapping. (f) The greedy routing path shown in the original network.

sure the success of greedy routing. Unfortunately, none of the greedy embedding algorithms in literatures can be extended from 2D to 3D general networks. The challenge of greedy routing in 3D networks is further revealed in [28], which proves that there does not exist a deterministic algorithm that can guarantee delivery based on local information only in 3D networks.

In view of the above challenges, several approaches have been proposed for recovery from local minimums in 3D networks. First, mapping and projection are introduced to reduce routing complexity in a 3D space. For example, a 3D network is projected to a 2D plane in [4,5] in order to apply face routing. However, face routing on the projected plane does not ensure a packet to move out of a void in the original 3D network. A different projection scheme is proposed in [7] for load balancing, which does not guarantee delivery either. Second, guarantee delivery can be achieved at the cost of more (non-constant-bounded) storage space. For example, a convex hull-based tree structure is introduced in GDSTR-3D [8]. A packet is greedily forwarded to its destination. If a local minimum is reached, GDSTR-3D switches to forwarding the packet along the edges of a spanning tree, guiding the packet to escape from the local minimum. GDSTR-3D offers deterministic routing. However, each node must maintain a set of convex hulls, and thus requires a storage space proportional to network size and some nodes (such as the roots of trees) are heavily loaded (see Fig. 3). Finally, local searching may be employed to jump out a local minimum. It is proposed in [3] to construct hulls to partition a network into subspaces, limiting the recovery to search a subspace only. Separately, the Random-Walk algorithm is proposed in [6] where random walk is employed on a local spherical structure to escape from voids when a local minimum is reached. However, such attempts for randomized recovery of local minimums are non-deterministic and often lead to high overhead or long delay. Among all routing algorithms discussed in literatures for 3D sensor networks, Random-Walk [6] is the sole truly greedy routing scheme with constant-bounded storage and computation complexity.

Our proposed solution is based on a unit tetrahedron cell (UTC) mesh structure. We propose a distributed algorithm to realize volumetric harmonic mapping of the UTC mesh under spherical boundary condition. It is a one-to-one map that yields virtual coordinates for each node in the network. Since a boundary has been mapped to a sphere, node-based greedy routing is always successful thereon. At the same time, we exploits the UTC mesh to develop a face-based greedy routing algorithm, and prove its success at internal nodes. To route a data packet to its destination, face-based and node-based greedy routing algorithms are employed alternately at internal and boundary UTCs, respectively. As far as we know, *this is the first work that realizes truly deterministic greedy routing with constant-bounded storage and computation in 3D sensor networks*.



(a) (b)

Figure 2: Comparison of face routing in 2D and 3D networks. Node S has a shorter distance to Destination D than all of its neighbors, and thus is a local minimum. (a) Face routing is successful in a 2D planar network because a concave void is a face with a simple line boundary, and thus a local deterministic algorithm can be employed to search the boundary in either clockwise or counter-clockwise direction as shown by the blue and red lines. (b) In a 3D network, a void is no longer a face. Its boundary becomes a surface, yielding an arbitrarily large number of possible paths to be explored (as indicated by the arrows). Thus face routing fails.

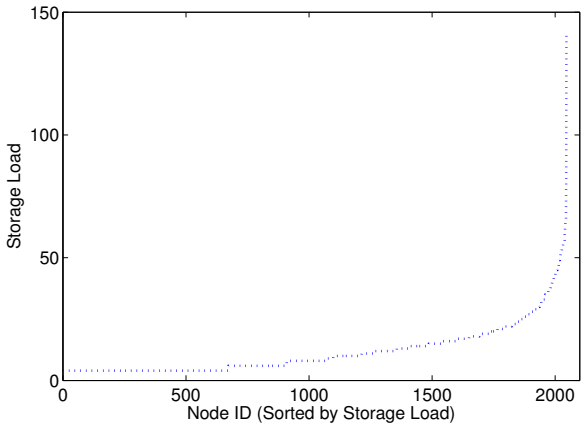


Figure 3: Storage load distribution in GDSTR-3D, where the storage load of a node is measured by the number of entries in its local convex hull table.

Each node simply stores the virtual coordinates of itself and its neighbors to make routing decisions.

The rest of this paper is organized as follows: Sec. 2 introduces the construction of UTC and related definitions. Sec. 3 proposes the face-based greedy routing algorithm. Sec. 4 elaborates the distributed volumetric harmonic mapping algorithm that yields virtual coordinates to support global end-to-end greedy routing. Sec. 5 presents our simulation results. Finally, Sec. 6 concludes the paper.

2. CONSTRUCTION OF UNIT TETRAHEDRON CELLS

We represent a wireless sensor network by a graph $G(V,E)$, where the vertices (V) denote the sensor nodes and the edges (E) indicate the communication links in the network.

Definition 1. *A* unit tetrahedron cell (UTC) *is a tetrahedron formed by four network nodes, which does not intersect with any other tetrahedrons.*

We let $UTC(A,B,C,D)$ denote the UTC formed by Nodes A,B,C and D, which includes four faces, i.e., $Face(A,B,C)$, $Face\ (A,B,D)$, $Face(A,C,D)$ and $Face(B,C,D)$. The union of all UTCs, called a *UTC mesh* (see Fig 1(c)), represent the network.

A simple algorithm is employed to create a UTC mesh, which starts from any arbitrary tetrahedron that contains its own vertex nodes only. By removing all edges that intersect this tetrahedron, the algorithm yields the first UTC, denoted by $UTC(A,B,C,D)$. Next the algorithm expands it to form other UTCs. Based on each face of $UTC(A,B,C,D)$, such as $Face(A,B,C)$, the algorithm looks for the common neighbors of Nodes A, B and C. Let E be such a common neighbor. Nodes A, B, C and E form a valid UTC only if it neither overlaps with any existing UTCs nor contains any other nodes. Multiple such nodes like E may exist, and the algorithm arbitrarily chooses one of them to form the new UTC. The algorithm repeats the above procedure until no new UTC can be formed.

Here we have assumed no degenerated edges or nodes exist in the network, and any internal hole (as small as a unit cube) has been identified by [29] to ensure the successful establishment of the UTC mesh. Moreover, we assume a node can create a local coordinates system by using local distance information estimated via standard methods [30]. Multiple available schemes are available for creating such a local coordinates system [31–34]. Our implementation adopts [34] for its efficiency of filtering noises in distance measurement and its tolerance of distance errors.

Definition 2. *A* Delaunay unit tetrahedron cell (DUTC) *is a UTC whose circumsphere contains no other nodes except its vertices.*

For example, $UTC(A,B,C,D)$ shown in Fig. 4(a) is a DUTC, since its circumsphere contains no nodes except A,B,C and D. On the other hand, Fig. 4(b) illustrates $UTC(A,B,C,D)$ that is not a DUTC because Node E is inside its circumsphere. Similarly, neither is $UTC(E,B,C,D)$ a DUTC. Note that the UTCs constructed by the algorithm introduced above are not necessarily DUTCs.

Definition 3. *A face is a* boundary face *if it is contained in one UTC only.*

Definition 4. *A UTC is a* boundary UTC *if it contains at least one boundary face. A non-boundary UTC is call an* internal UTC.

Definition 5. *A* hole *of a network is formed by a closed surface that consists of boundary faces. The outer boundary of the network is considered as a special hole.*

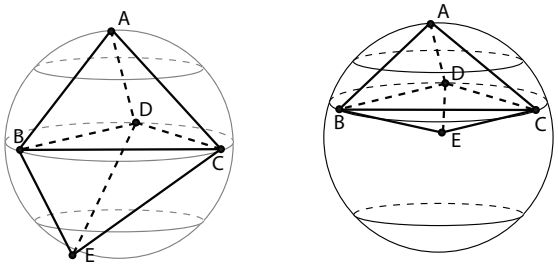
For example, a set of boundary faces are highlighted in magenta in Fig 1(c), which together form the surface of the hole.

Definition 6. *Two UTCs are neighbors if and only if they share a face.*

Apparently, a UTC has at most 4 neighboring $UTCs$. Similarly, we have

Definition 7. *Two faces are neighbors if and only if they share an edge.*

Definition 8. *Node i is greedily reachable to Node j if a packet can be greedily routed from the former to the latter based on a metric that is kept locally and consumes constant storage space.*



(a) DUTC. (b) Non-DUTC.

Figure 4: Illustration of Delaunay unit tetrahedron cell (DUTC), under an arbitrary (non-unit disk graph (UDG)) communication model.

Our objective is to enable greedy routing from any source to any destination in a given sensor network. More specifically, we aim to map an arbitrary 3D sensor network to a *greedily reachable network* as defined below:

Definition 9. *A network is called a* greedily reachable network *if every two nodes in the network are* greedily reachable *to each other.*

Based on the above definitions, we next discuss how to enable a greedily reachable network. We first examine a simple case, i.e., greedy routing at internal nodes, which appears trivial but is anti-intuitively unachievable by straightforward application of the node-based greedy routing algorithm. Then we introduce our proposed approach based on volumetric harmonic mapping for global end-to-end greedy routing.

3. ROUTING AT INTERNAL UTCS: FACE-BASED GREEDY ROUTING

As discussed in Sec. 1 and demonstrated in Fig. 1(b), the node-based greedy routing scheme ensures success at neither boundary nor internal nodes in a 3D wireless sensor network. We focus on the latter in this section.

We first show that node-based greedy routing is not always successful even under the UTC structure, for example, in a case as simple as across two neighboring UTCs. More specifically, let's consider two UTCs, $UTC(A,B,C,D)$ and $UTC(B,C,D,E)$, which share $Face(B,C,D)$. If they are DUTCs as illustrated in Fig. 4(a), Nodes A and E are greedily reachable to each other. However, building DUTCs is expensive and often impractical in sensor networks. The UTCs constructed by the algorithm introduced in Sec. 2 are not necessarily DUTCs. For a UTC shown in Fig. 4(b) for example, l_{EA} can be shorter than l_{BA}, l_{CA}, and l_{DA} (where l_{ij} denotes the distance between Nodes i and j), thus resulting in a failure in node-based greedy routing from E to A. Obviously, the UTC mesh is a special structure of a 3D sensor network. Since the node-based greedy routing is not always successful under UTCs, it offers no guarantee of data delivery in a general 3D sensor network either.

To this end, we propose a face-based greedy routing algorithm. Let's consider a data packet that is to be delivered from Source S to Destination D. Similar to conventional node-based greedy routing algorithms, we assume the data packet contains the IDs and coordinates of S and D. Note that, the coordinates are not necessarily based on GPS. Instead, they can be virtual coordinates, e.g., produced by our proposed volumetric harmonic mapping algorithm as to be discussed later in Sec. 4.

Node S first computes a line segment between S and D, which is denoted by Γ. Clearly, Γ passes through a set of UTCs between S and D, and intersects with a sequence of faces, denoted by $\Phi = \{Face(A_i,B_i,C_i) | 1 \le i \le k\}$, where k is the total number of such faces (see Fig. 5). The distance from $Face(A_i,B_i,C_i)$ to Destination D is defined as the distance between D and the intersection point of Γ and $Face(A_i,B_i,C_i)$. As to be proved in Lemma 1, $Face(A_i,B_i,C_i)$ and $Face(A_{i+1},B_{i+1},C_{i+1})$ must be neighboring faces, and thus share an edge. Let's denote the shared edge as τ_i.

Under the proposed face-based greedy routing algorithm, data packets are forwarded from $Face(A_1,B_1,C_1)$ to $Face\ (A_k,B_k,C_k)$. Each intermediate node only needs to calculate the next face in Φ. For example, Node S can easily determine $Face(A_1,B_1,C_1)$, because the latter must be one of the faces in the UTCs that contain the former. Therefore, Node S can check which of them intersects with Γ, with a computation time bounded by a small constant. $Face(A_2,B_2,C_2)$ is determined similarly. Thus the packet is routed from S to one of the end nodes of Edge τ_1, i.e., the edge shared by $Face(A_1,B_1,C_1)$ and $Face(A_2,B_2,C_2)$. The above process repeats at each intermediate node, until the packet arrives at $Face(A_k,B_k,C_k)$ that contains Destination D, or it fails to find the next face in Φ based on locally available information.

Next we prove that the proposed face-based greedy routing algorithm is always successful at internal nodes.

Lemma 1. *The face-based greedy routing does not fail at a non-boundary UTC.*

PROOF. Γ intersects with a sequence of faces, i.e., Φ. Since Γ is a straight line segment, it is obvious that $Face(A_{i+1}, B_{i+1}, C_{i+1})$ must be closer to the destination compared with $Face(A_i,B_i,C_i)$ for $i < k$, as illustrated in Fig. 5. To prove the lemma, we only need to show that $Face(A_i,B_i,C_i)$ and $Face(A_{i+1},B_{i+1},C_{i+1})$ are neighboring faces and thus a routing decision can be made by using local information only, if $Face(A_i,B_i,C_i)$ is a non-boundary face.

Γ penetrates through a set of UTCs. According to Definitions 3 and 6, a non-boundary face is always shared by two UTCs. Thus when Γ intersects with a non-boundary face, it can be consider as exiting from the current UTC or entering into the next UTC.

Let's consider that Γ enters a UTC when it intersects with a non-boundary face, e.g., $Face(A_i,B_i,C_i)$. According to Definition 1, Γ does not meet any faces inside the UTC. Thus the next face it meets, i.e., $Face(A_{i+1},B_{i+1},C_{i+1})$, must be another face of the same UTC, as along as $Face(A_i,B_i,C_i)$ is not a boundary face. Since any two faces of a tetrahedron share an edge, $Face(A_i,B_i,C_i)$ and $Face(A_{i+1},B_{i+1},C_{i+1})$ must be neighboring faces according to Definition 7. As a result, routing from the former to the latter can be achieved by using local information only. The lemma is thus proven. □

Lemma 1 shows that greedy routing via Φ always advances data packets toward the destination at internal UTCs, which is in a sharp contrast to node-based greedy routing where local minimum exists at internal nodes (as demonstrated in Fig. 1(b)).

4. GLOBAL END-TO-END GREEDY ROUTING: VOLUMETRIC HARMONIC MAPPING (VHM)

The face-based greedy routing algorithm supports greedy data forwarding at internal UTCs. However, as depicted in Fig. 1, it may fail at boundaries, which are generally complex and concave. This naturally motivates us to map a boundary to a sphere, yielding virtual coordinates for boundary nodes such that any two nodes on

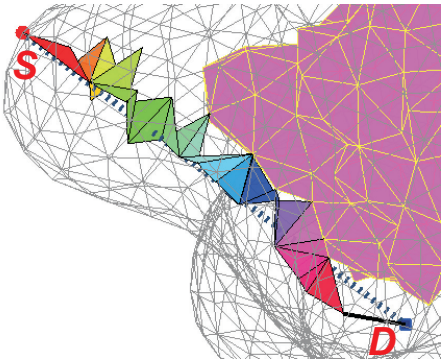


Figure 5: A packet is routed through a sequence of faces under face-based greedy routing.

a boundary are greedily reachable to each other. But note that it is insufficient to map boundaries only, because the virtual coordinates for boundary nodes would become inconsistent with the coordinates of the internal nodes. As a result, greedy routing fails when a routing path involves both boundary nodes and internal nodes. More specifically, although greedy routing is supported between any two nodes on a boundary based on their virtual coordinates, a node cannot identify the correct target on the boundary, in order to advance the packet to its destination. To this end, we propose a distributed algorithm to realize volumetric harmonic mapping (VHM) under spherical boundary condition. It is a one-to-one map that yields virtual coordinates for each node in the entire 3D wireless sensor network to enable global end-to-end greedy routing.

4.1 Theoretical Insights

First, we briefly introduce the necessary theoretical background knowledge that provides useful insights and underlies our proposed algorithm.

4.1.1 Volumetric Embedding

The volumetric embedding is the process of computing a map between the original volumetric data and a *canonical domain* in $\mathbb{R}^3$.

For the purpose of computation, a volume is usually modeled as point clouds or a piecewise linear tetrahedral mesh:

$$\mathbb{M} = (\mathbb{T}, \mathbb{F}, \mathbb{E}, \mathbb{V}, \mathbb{C}), \tag{1}$$

where $\mathbb{T}$, $\mathbb{F}$, $\mathbb{E}$ and $\mathbb{V}$ are the sets of tetrahedra, triangular faces, edges and vertices in the mesh, while $\mathbb{C}$ describes the connectivity among them.

Volumetric embedding is to assign a set of 3D coordinates to every vertex in the volumetric data. Note that although the mapping function is by definition restricted on vertices, it can be extended through out the whole tetrahedral mesh piecewisely. More specifically, the function value for an arbitrary point in the volume is defined as the interpolation of the values on the four vertices of the enclosing tetrahedron, inducing a piecewise-linear map from the original volumetric mesh $\mathbb{M}$ to a canonical domain $\mathbb{N}$. The domain $\mathbb{N}$ is a subset of $\mathbb{R}^3$, and should ideally have a regular shape. In our case, the canonical domain is a solid ball in order to support greedy routing.

4.1.2 Volumetric Harmonic Function

Our goal is to construct virtual coordinates for a 3D sensor network to support successful greedy routing. The virtual coordinates must be one-to-one correspondent to the sensor nodes. To this end, we resort to volumetric harmonic mapping (VHM) under spherical boundary condition.

In general, a function f is *harmonic* if it satisfies the Laplace's equation $\triangle f = 0$. If Dirichlet boundary condition is imposed on this partial differential equation, a harmonic function is the solution of the Dirichlet's problem.

The same concepts can be well formulated on volumes in a discrete setting. To this end, we first introduce the definition of edge weight.

Definition 10. *For Edge e_{ij} which connects Vertices v_i and v_j, its edge weight k_{ij} is a real value determined as follows. Suppose Edge e_{ij} is shared by t adjacent tetrahedra. Then it lies against t dihedral angles $\{\theta_m | 1 \le m \le t\}$. The weight of e_{ij} is defined as*

$$k_{ij} = \frac{1}{t} \sum_{m=1}^{t} l_m \cot \theta_m, \tag{2}$$

where l_m is the length of edge to which e_{ij} is against in the UTC mesh.

Based on edge weight, we next define the piecewise Laplacian under discrete setting.

Definition 11. *The piecewise Laplacian is the linear operator $\triangle_{PL} f = 0$ on the space of piecewise linear functions. Let's define a map $f : \mathbb{T} \to \mathbb{R}^3$, where $f = (f_0, f_1, f_2)$. f_0, f_1, and f_2 are corresponding to three dimensions, and each of them is a real valued function defined over the vertices of the UTC mesh. The piecewise Laplacian of f is:*

$$\triangle_{PL} f = (\triangle_{PL} f_0, \triangle_{PL} f_1, \triangle_{PL} f_2), \tag{3}$$

where $\triangle_{PL} f_m = \sum_{e_{ij} \in \mathbb{E}} k_{ij} (f_m(v_j) - f_m(v_i))$ for $m = 0, 1, 2$.

Our goal is to find f such that $\triangle_{PL} f = 0$, i.e., the volumetric harmonic function. It is equivalent to minimize the following volumetric harmonic energy.

Definition 12. *The volumetric harmonic energy of f is:*

$$E(f) = \sum_{m=0}^{2} E(f_m), \tag{4}$$

where $E(f_m) = \sum_{e_{ij} \in \mathbb{E}} k_{ij} ||f_m(v_j) - f_m(v_i)||^2$.

If f minimizes the volumetric harmonic energy, then it satisfies the condition $\triangle_{PL} f = 0$, i.e., f is harmonic.

4.1.3 Spherical Harmonic Function

Similar to that in a smooth setting, we can impose Dirichlet boundary conditions on the discrete volumetric harmonic function, by fixing the value of f on certain vertices $v_i \in \mathbb{V}_c$, where $\mathbb{V}_c$ is the set of constraint vertices. It is important to control boundary conditions in this work. Particularly, spherical boundaries are desired to support greedy routing.

The spherical harmonic function maps a closed topologically spherical surface (i.e., a surface with no holes) to a sphere. It is similar to the volumetric harmonic function. They share the same harmonic energy as defined in Eqn. (4) but differ in how to assign weight k_{ij}. For a topologically spherical surface, an edge is shared by two faces only. For example, given Edge e_{ij} shared by triangle faces f_{ijk} and f_{jil}, its weight is defined as

$$k_{ij} = \frac{1}{2} (\cot \theta_l + \cot \theta_k), \tag{5}$$

where $\theta_l = \angle v_i v_l v_j$ and $\theta_k = \angle v_i v_k v_j$.

4.2 Distributed Mapping Algorithm

Based on the theory introduced above, we now propose a practical distributed algorithm to realize volumetric harmonic mapping under spherical boundary condition.

Let's first consider a solid sensor network with a possibly complex and concave external boundary but no internal holes (see Figs. 6(a)-6(c) for examples). A UTC mesh is established as discussed in Sec. 2 (as shown in Figs. 6(f)-6(h)). We construct a volumetric harmonic map with the heat flow method such that the entire UTC mesh is homeomorphically (one-to-one) mapped to a solid tetrahedra ball in $\mathbb{R}^3$ (as illustrated in Figs. 6(k)-6(m)). The proposed algorithm follows two steps, as outlined below sequentially.

4.2.1 Distributed Spherical Harmonic Map

First we map the boundary of the 3D volume homeomorphically (one-to-one) to a sphere by using spherical harmonic map. The boundary nodes of a 3D sensor network are identified as the nodes on boundary faces. Each boundary node is associated with a 3-vector metric, i.e., $u_i = (u_i^0, u_i^1, u_i^2)$ for Node i, representing its 3D virtual coordinates. It is initialized by random coordinates on a unit sphere or by the normalized normal of Node i in order to accelerate convergence of the algorithm. Then, the algorithm goes through an iterative procedure. During the n-th iteration, Node i computes its current spherical harmonic energy:

$$E_i^n = \sum_{j=1}^{N_i} k_{ij}(u_i^{n-1} - u_j^{n-1})^2, \tag{6}$$

where N_i is the degree of Node i and k_{ij} is defined earlier in Eqn. (5). Node i then updates its u_i along the negative of the gradient direction of its energy:

$$u_i^n = u_i^{n-1} - \gamma \nabla E_i^n, \tag{7}$$

where γ is a small constant (which is set to 0.1 in our simulations). Next, u_i is normalized such that it is always on the unit sphere. The above process repeats, until the difference between E_i^n and E_i^{n-1} is less than a small constant δ (e.g., $\delta = 10^{-6}$) for all nodes in the network. The final u_i serves as the virtual coordinates of Node i. The algorithm is distributed, where a node only needs to communicate with its one-hop neighbors in each iteration. Moreover, its convergence is guaranteed [35].

4.2.2 Distributed Volumetric Harmonic Map

By now, we have arrived at a spherical harmonic mapping, which maps the network boundary one-to-one to a sphere. Next we apply volumetric harmonic map by minimizing the volumetric harmonic energy under the computed spherical boundary condition. More specifically, if a node is on the boundary, it simply keeps it current virtual coordinates obtained above. On the other hand, a non-boundary node, e.g, Node i, determines its 3D virtual coordinates via volumetric harmonic map. Similar to the spherical harmonic mapping discussed above, Node i is associated with a 3-vector metric, i.e., u_i, which represents its volumetric virtual coordinates. Node i iteratively calculates E_i and u_i according to Eqn. (6) and (7). But note that, the edge weight (i.e., k_{ij}) is now computed according to Definition 10, instead of Eqn. (5). When E_i differs by less than a small constant δ between two iterations for all nodes in the network, the volumetric harmonic mapping algorithm terminates, yielding the virtual coordinates for every internal node. Example of the result after volumetric harmonic mapping are given in Figs. 6(k)-6(m). Again the algorithm is distributed, where a node only needs to communicate with its one-hop neighbors in each iteration. The proof of its convergence can be found in Appendix.

4.2.3 Further Discussions

The above discussions are for a solid 3D sensor network without inner holes. If there is a hole inside (see Figs. 1(a), 6(d) and 6(e) for examples), the boundary condition has been changed. Two boundary surfaces will be detected, one outside and the other inside. The same spherical harmonic mapping algorithm is applied to map them to two unit spheres, respectively. Then, the boundary nodes perform simple local calculations to align the inner sphere to the outer sphere. Specifically, the nodes on the inner boundary scale their coordinates to reduce the radius of the inner sphere to r', which is constant less than one. Next, a node on the outer boundary with its virtual coordinates most close to (0,0,1) finds its closest node on the inner boundary based on the UTC mesh. The two nodes and the center of the spheres (i.e., (0,0,0)) form an angle, denoted as θ_0 (which is calculated according to virtual coordinates). θ_0 is broadcasted to all nodes on the inner boundary, which subsequently apply a rotation matrix with Angle θ_0 on their virtual coordinates. Therefore the inner sphere is aligned with the outer sphere with respect to this pair of nodes. Then, another node on the outer boundary with its virtual coordinates most close to (0,1,0) repeats the above process to initiate the second round of rotation. After two rotations, the inner sphere and the outer sphere are approximately aligned, setting the spherical boundary conditions. Finally the volumetric harmonic map introduced above is applied to produce 3D virtual coordinates for each internal node in the network. Examples of such mapping results are illustrated in Figs. 1(d), 6(n) and 6(o).

For a network with more than one inner holes, a trivial clustering algorithm is employed to segment the network into clusters, each centering at an inner hole. The algorithm discussed above is applied in each cluster to create virtual coordinates. Greedy routing is thus supported inside a cluster. However, routing across clusters must rely on gateways or global coordinates alignment, which remains challenging under the constraints of constant-bounded storage and computation at individual node. We will address it in our future work.

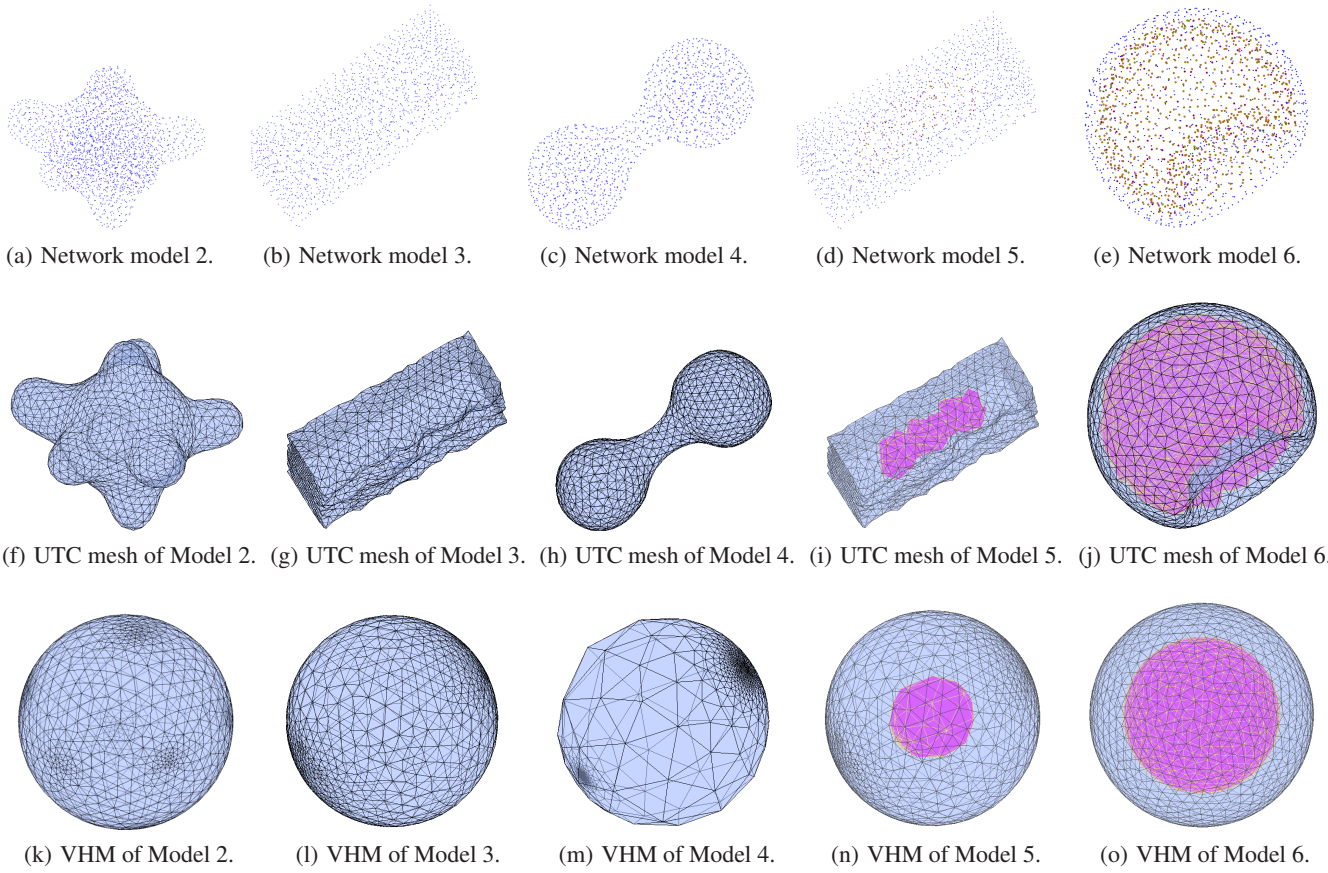
Note that the harmonic mapping of a shape in 2D is guaranteed a diffeomorphism, if the boundary of the 2D shape is mapped to a convex planar curve and the mapping is homeomorphism. However, for 3D cases, even if the image of the boundary is convex, diffeomorphism is not theoretically guaranteed for harmonic mapping, although we have not found a single non-diffeomorphism case for networks without or with one hole in our extensive experimental tests.

Finally, the proposed algorithm based on volumetric mapping may become less efficient under extreme conditions. For example, a higher stretch factor is expected if the original network is extremely narrow or thin. Nevertheless, the proposed algorithm still ensures successful greedy routing.

4.2.4 Summary of the Routing Algorithm

The above mapping algorithm is executed during network initialization. After mapping, each node has its own virtual coordinates in a 3D space. Since a boundary has been mapped to a sphere, node-based greedy routing is always successful thereon. At the same time, the UTC mesh remains valid under the virtual coordinates. Thus successful greedy routing at internal nodes is achieved by face-based greedy routing. To route a data packet to its destination, face-based and node-based greedy routing algorithms are employed alternately at internal and boundary UTCs, respectively.

An example is given in Fig. 1(f), where a data packet is delivered from S to D. Node S first identifies a sequence of faces Φ that intersects with the line segment between S and D. If the next face is reachable according to local information, the packet is forwarded



(a) Network model 2. (b) Network model 3. (c) Network model 4. (d) Network model 5. (e) Network model 6.

(f) UTC mesh of Model 2. (g) UTC mesh of Model 3. (h) UTC mesh of Model 4. (i) UTC mesh of Model 5. (j) UTC mesh of Model 6.

(k) VHM of Model 2. (l) VHM of Model 3. (m) VHM of Model 4. (n) VHM of Model 5. (o) VHM of Model 6.

Figure 6: 3D network models and mapping results, where the first row shows original networks, the second row illustrates the established UTC mesh structures with only boundary UTCs for conciseness, and the third row depicts the results after VHM. The inner boundary is highlighted in the magenta.

accordingly by face-based greedy routing. When the packet fails to find the next face toward Node *D*, it must arrive at a boundary, which has been mapped to a sphere. Thus node-based greedy routing is applied to move the packet across the void. Whenever *D* becomes reachable reachable, face-based greedy routing is applied again. The above process continues until the packet reaches its destination.

5. APPLICATIONS AND SIMULATIONS

We have implemented the face-based greedy routing algorithm and the volumetric harmonic mapping (VHM) algorithm introduced above, in order to achieve highly efficient peer-to-peer greedy routing in 3D sensor networks. Moreover, we further apply the proposed routing algorithm in in-network data centric storage and retrieval. The simulation results are presented below sequentially.

5.1 Peer-to-Peer Greedy Routing

Various 3D sensor networks in different sizes (ranging from 1,000 to 2,500) and shapes are simulated in this work. In addition to the network model shown in Fig. 1, Fig. 6 illustrates several other examples, where the first row shows original networks, the second row illustrates the established UTC mesh structures, and the third row depicts the results after volumetric harmonic mapping. Sensor nodes are randomly distributed. The radio transmission range is around 0.11, resulting in an average nodal degree between 16 to 30. Note that such nodal degree is moderate in 3D although it appears high for 2D networks. The boundaries are detected as discussed in previous section. For example, the inner boundary is highlighted in magenta in Figs. 6(i) and 6(j).

5.1.1 Stretch Factor

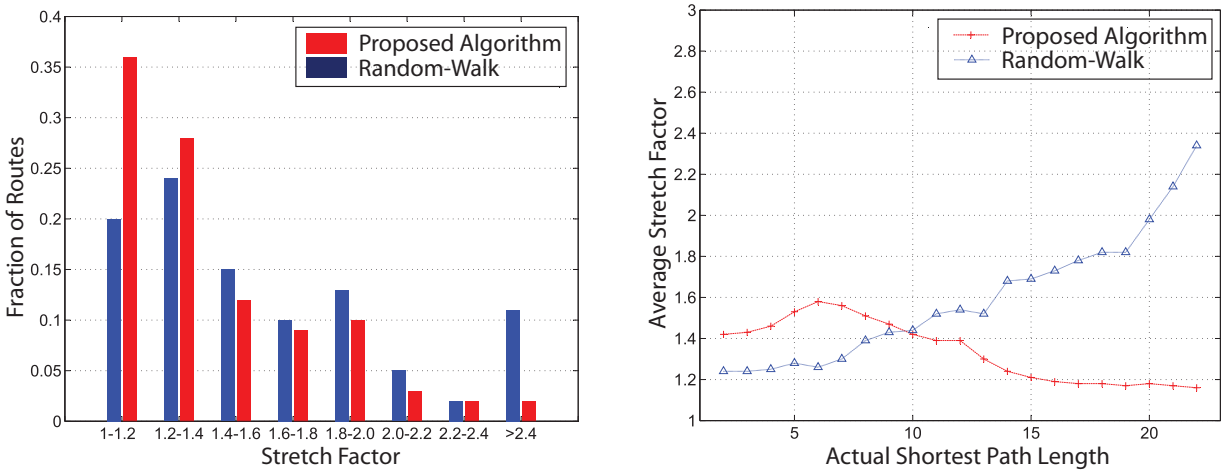
As proven in previous sections, the proposed scheme guarantees successful data delivery between any pair of nodes. Therefore we focus on stretch factor in performance evaluation. The stretch factor of a route is the ratio of the actual path length to the shortest path length. We randomly select 10,000 pairs of nodes to calculate the average stretch factor for each network model.

While many greedy routing algorithms have been proposed for wireless sensor networks, few of them can be applied in a 3D setting. Moreover, we only focus on truly greedy routing algorithms with constant-bounded storage and computation complexity in 3D sensor networks in this research. Therefore, Random-Walk [6] is the sole comparable scheme, as discussed in Sec. 1. Under Random-Walk, a packet is greedily advanced to its destination. If a local minimum is reached, it escapes from the local minimum by random walk on a local spherical structure. Note that Random-Walk does not ensure deterministic routing results.

The average stretch factors of Random-Walk and our proposed algorithm in different networks are summarized in Table 1. As can be seen, the proposed scheme exhibits stable stretch factor and out-

Table 1: Comparison of stretch factors.

	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Overall Average
Proposed Algorithm	1.63	1.63	1.66	1.61	1.62	1.44	1.59
Random-Walk [6]	1.83	1.70	1.73	1.84	1.89	2.12	1.85



(a) Stretch factor distribution.

(b) Impact of path length.

Figure 7: (a) More routes under the proposed scheme have low stretch factors than Random-Walk does. (b) With the increase of the actual path length between source and destination, the stretch factor decreases under our proposed scheme, while the stretch factor of Random-Walk grows noticeably.

performs Random-Walk in all network models. In a contrast, the performance of Random Walk heavily depends on the size of the hole, experiencing a higher stretch factor under the network with a bigger hole.

Fig. 7 illustrates the distribution of stretch factor based on Network model 6. We observe that most routing paths under our proposed scheme have low stretch factor. For example, 70% of routes have their stretch factor lower than 1.4. On the other hand, the distribution under Random-Walk has a considerable shift to the right side. Particularly, there are about 20% routes experiencing a stretch factor of 2.0 or higher (which means a routing path at least as twice long as the shortest path).

It is also an interesting observation from Fig. 7(b) that, with the increase of the actual path length between source and destination, the stretch factor decreases under our proposed scheme. This is in a sharp contrast to Random-Walk, where two far-separated nodes are likely located on the opposite side of the hole and accordingly conventional node-based greedy routing between them may fail, leading to a long random walk path.

5.1.2 Load Distribution

We have also calculated the traffic load distribution among the sensor nodes, as illustrated in Fig. 8. The traffic load under our proposed scheme is well balanced, with more than 40% of the nodes involved in less than 30 routes and around 60% in less than 50 routes. Random-Walk performs greedy routing first, and then randomly searches along the boundaries when a dead-end is reached. Thus, the nodes near boundaries (especially inner boundaries) usually experience heavy load.

5.2 Data Storage and Retrieval

Our scheme guarantees greedy and stateless peer-to-peer routing in a 3D sensor network. Another possible application of the proposed scheme is for data centric networking which supports in-network data storage and query. Traditionally, the underlying network used for data storage and retrieval needs to store a great amount of routing informations. With the proposed techniques, the whole network consists of tetrahedrons and is mapped into a unit sphere in a 3-D space, which supports greedy peer-to-peer routing. Therefore we propose to uniformly map a datum to a point inside of the unit sphere and let the tetrahedron which contains the point to store the datum. Both data insertion and retrieval are naturally supported by greedy routing.

5.2.1 Where to store the data

Given a datum, finding the location to store it is the key issue in data centric networking. To this end, we exploit the result of volumetric harmonic mapping, which maps the original network with an irregular shape to a ball. We adopt the polar coordinate system, where a point in a unit sphere can be represented by (ρ, α, β), where ρ is the distance from this point to the origin, α is the angle formed by the line connecting this point and the origin with X-axis, and β is the angle between the line and Y-axis. ρ is bounded to $0 \leq \rho \leq 1$ for a network without hole, or $r' \leq \rho \leq 1$ for a network with a hole, where r' is the radius of the inner sphere in volumetric harmonic mapping.

First, we map a datum (possibly with multiple attributes) to a series of bits by using the method introduced in [36–38]. We select the $(3k+1)$-th bits (where k ranges from 0 to the largest value that depends on the length of the bit series), and concatenate them to a binary string, which is further normalized to yield ρ. Similarly, $(3k+2)$-th and $(3k+3)$-th bits are used to determine α and β, respectively. Since the unit ball (or hollow ball) is the volumetric harmonic mapping of the tetrahedron mesh of the original network,

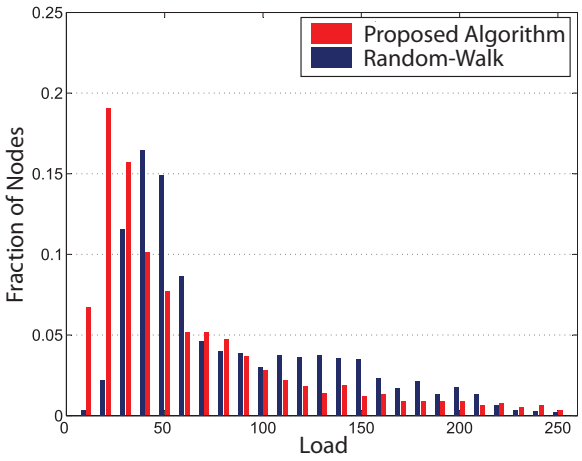


Figure 8: Load distribution in routing.

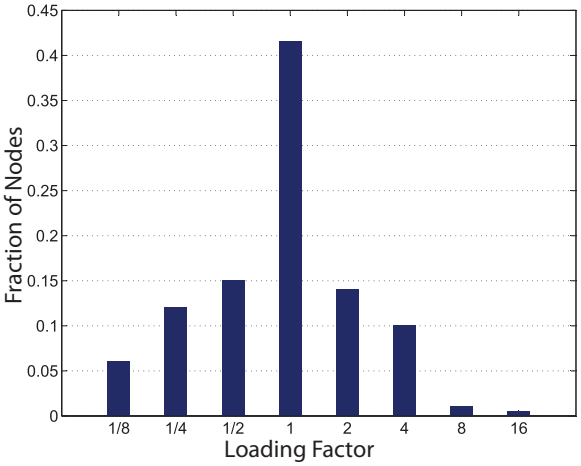


Figure 9: Distribution of loading factor.

a point (ρ, α, β) must be within a tetrahedron. Thus, we simply select one of the four vertex of this tetrahedron to store the datum.

5.2.2 *Route data and query*

Once a datum is mapped to a point location (ρ, α, β), it is routed toward the location by following our proposed greedy routing algorithm. In each hop of routing, it checks if (ρ, α, β) is inside the current tetrahedron. If it is true, the routing terminates and the datum is stored by one of the tetrahedron's vertices that has the lowest load. Query and retrieval of data can be realized similarly. The data being queried is mapped to a point location, and the query is routed to the corresponding tetrahedron that contains the requested data.

5.2.3 *Performance*

We have carried out simulations to insert 10,000 data generated by a set of randomly chosen nodes. Each data has a value ranging from 0 to 100. Fig. 9 shows the distribution of loading factor, i.e., the ratio of the actual load to the ideal load, where the ideal load is achieved when the data are evenly distributed over all nodes in the network. As can be seen, the loading factor is nicely distributed, where more than 40% nodes enjoy a perfect loading factor of one, signifying a well-balanced traffic among sensor nodes.

6. CONCLUSION

Viewing significant challenges encountered in extending greedy routing from 2D to 3D space, we have investigated decentralized solutions to achieve greedy routing in 3D sensor networks. Our proposed approach is based on a unit tetrahedron cell (UTC) mesh structure. We have proposed a distributed algorithm to realize volumetric harmonic mapping of the UTC mesh under spherical boundary condition. It is a one-to-one map that yields virtual coordinates for each node in the network. Since a boundary has been mapped to a sphere, node-based greedy routing is always successful thereon. At the same time, we have exploited the UTC mesh to develop a face-based greedy routing algorithm, and proved its success at internal nodes. To route a data packet to its destination, face-based and node-based greedy routing algorithms are employed alternately at internal and boundary UTCs, respectively. To our best knowledge, this is the first work that realizes deterministic greedy routing with constant-bounded storage and computation in 3D sensor networks.

7. ACKNOWLEDGEMENTS

S. Xia, H. Wu and M. Jin are partially supported by NSF CNS-1018306. X. Yin and X. Gu are partially supported by NSF CNS-1016829. Also, we are grateful to the anonymous reviewers and our shepherd Lakshmi Subramanian for their constructive feedback.

8. REFERENCES

[1] X. Bai, C. Zhang, D. Xuan, J. Teng, and W. Jia, "Low-Connectivity and Full-Coverage Three Dimensional Networks," in *Proc. of MobiHOC*, pp. 145–154, 2009.

[2] X. Bai, C. Zhang, D. Xuan, and W. Jia, "Full-Coverage and K-Connectivity (K=14, 6) Three Dimensional Networks," in *Proc. of INFOCOM*, pp. 388–396, 2009.

[3] C. Liu and J. Wu, "Efficient Geometric Routing in Three Dimensional Ad Hoc Networks," in *Proc. of INFOCOM*, pp. 2751–2755, 2009.

[4] T. F. G. Kao and J. Opatmy, "Position-Based Routing on 3D Geometric Graphs in Mobile Ad Hoc Networks," in *Proc. of The 17th Canadian Conference on Computational Geometry*, pp. 88–91, 2005.

[5] J. Opatrny, A. Abdallah, and T. Fevens, "Randomized 3D Position-based Routing Algorithms for Ad-hoc Networks," in *Proc. of Third Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services*, pp. 1–8, 2006.

[6] R. Flury and R. Wattenhofer, "Randomized 3D Geographic Routing," in *Proc. of INFOCOM*, pp. 834–842, 2008.

[7] F. Li, S. Chen, Y. Wang, and J. Chen, "Load Balancing Routing in Three Dimensional Wireless Networks," in *Proc. of ICC*, pp. 3073–3077, 2008.

[8] J. Zhou, Y. Chen, B. Leong, and P. Sundar, "Practical 3D Geographic Routing for Wireless Sensor Networks," in *Proc. of SenSys*, pp. 337–350, 2010.

[9] D. Pompili, T. Melodia, and I. F. Akyildiz, "Routing Algorithms for Delay-insensitive and Delay-sensitive Applications in Underwater Sensor Networks," in *Proc. of MobiCom*, pp. 298–309, 2006.

[10] W. Cheng, A. Y. Teymorian, L. Ma, X. Cheng, X. Lu, and Z. Lu, "Underwater localization in sparse 3d acoustic sensor networks," in *Proc. of INFOCOM*, pp. 798–806, 2008.

[11] J. Allred, A. B. Hasan, S. Panichsakul, W. Pisano, P. Gray, J. Huang, R. Han, D. Lawrence, and K. Mohseni,

"SensorFlock: An Airborne Wireless Sensor Network of Micro-Air Vehicles," in *Proc. of SenSys*, pp. 117–129, 2007.

[12] J.-H. Cui, J. Kong, M. Gerla, and S. Zhou, "Challenges: Building Scalable Mobile Underwater Wireless Sensor Networks for Aquatic Applications," *IEEE Network*, vol. 20, no. 3, pp. 12–18, 2006.

[13] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with Guaranteed Delivery in Ad Hoc Wireless Networks," in *Proc. of Third Workshop Discrete Algorithms and Methods for Mobile Computing and Communications*, pp. 48–55, 1999.

[14] B. Karp and H. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," in *Proc. of MobiCom*, pp. 1–12, 2001.

[15] E. Kranakis, H. Singh, and J. Urrutia, "Compass Routing on Geometric Networks," in *Proc. of Canadian Conference on Computational Geometry (CCCG)*, pp. 51–54, 1999.

[16] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger, "Geometric Ad-hoc Routing: Theory and Practice," in *Proc. of The 22nd ACM Symposium on the Principles of Distributed Computing*, pp. 63–72, 2003.

[17] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Worst-case Optimal and Average-case Efficient Geometric Ad-hoc Routing," in *Proc. of MobiHOC*, pp. 267–278, 2003.

[18] B. L. S. Mitra and B. Liskov, "Path Vector Face Routing: Geographic Routing with Local Face Information," in *Proc. of ICNP*, pp. 147–158, 2005.

[19] H. Frey and I. Stojmenovic, "On Delivery Guarantees of Face and Combined Greedy-face Routing in Ad Hoc and Sensor Networks," in *Proc. of MobiCom*, pp. 390–401, 2006.

[20] G. Tan, M. Bertier, and A.-M. Kermarrec, "Visibility-Graph-based Shortest-Path Geographic Routing in Sensor Networks," in *Proc. of INFOCOM*, pp. 1719–1727, 2009.

[21] C. Papadimitriou and D. Ratajczak, "On A Conjecture Related to Geometric Routing," *Theoretical Computer Science*, vol. 344, no. 1, pp. 3–14, 2005.

[22] P. Angelini, F. Frati, and L. Grilli, "An Algorithm to Construct Greedy Drawings of Triangulations," in *Proc. of The 16th International Symposium on Graph Drawing*, pp. 26–37, 2008.

[23] T. Leighton and A. Moitra, "Some Results on Greedy Embeddings in Metric Spaces," in *Proc. of The 49th IEEE Annual Symposium on Foundations of Computer Science*, pp. 337–346, 2008.

[24] R. Kleinberg, "Geographic Routing Using Hyperbolic Space," in *Proc. of INFOCOM*, pp. 1902–1909, 2007.

[25] A. Cvetkovski and M. Crovella, "Hyperbolic Embedding and Routing for Dynamic Graphs," in *Proc. of INFOCOM*, pp. 1647–1655, 2009.

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

[27] R. Flury, S. Pemmaraju, and R. Wattenhofer, "Greedy Routing with Bounded Stretch," in *Proc. of INFOCOM*, pp. 1737–1745, 2009.

[28] S. Durocher, D. Kirkpatrick, and L. Narayanan, "On Routing with Guaranteed Delivery in Three-Dimensional Ad Hoc Wireless Networks," in *Proc. of International Conference on Distributed Computing and Networking*, pp. 546–557, 2008.

[29] H. Zhou and et. al., "Localized Algorithm for Precise Boundary Detection in 3D Wireless Networks," in *Proc. of ICDCS*, pp. 744–753, 2010.

[30] Z. Zhong and T. He, "MSP: Multi-Sequence Positioning of Wireless Sensor Nodes," in *Proc. of SenSys*, pp. 15–28, 2007.

[31] G. Giorgetti, S. Gupta, and G. Manes, "Wireless Localization Using Self-Organizing Maps," in *Proc. of IPSN*, pp. 293 – 302, 2007.

[32] L. Li and T. Kunz, "Localization Applying An Efficient Neural Network Mapping," in *Proc. of The Int'l Conference on Autonomic Computing and Communication Systems*, pp. 1–9, 2007.

[33] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz, "Localization from Mere Connectivity," in *Proc. of MobiHOC*, pp. 201–212, 2003.

[34] Y. Shang and W. Ruml, "Improved MDS-based Localization," in *Proc. of INFOCOM*, pp. 2640–2651, 2004.

[35] X. Gu, Y. Wang, T. F. Chan, P. M. Thompson, and S.-T. Yau, "Genus Zero Surface Conformal Mapping and Its Application to Brain Surface Mapping," *IEEE Transaction on Medical Imaging*, vol. 23, no. 8, pp. 949–958, 2004.

[36] J. Li, J. Jannotti, D. S. J. De Couto, D. R. Karger, and R. Morris, "A Scalable Location Service for Geographic Ad Hoc Routing," in *Proc. of MobiCom*, pp. 120–130, 2000.

[37] X. Li, Y. J. Kim, R. Govindan, and W. Hong, "Multi-dimensional Range Queries in Sensor Networks," in *Proc. of SenSys*, pp. 63–75, 2003.

[38] C. Yu-Chi, S. I-Fang, and L. Chiang, "Supporting Multi-Dimensional Range Query for Sensor Networks," in *Proc. of ICDCS*, pp. 35–35, 2007.

[39] L. C. Evans, *Partial Differential Equations*. American Mathematical Society, 2010.

Appendix

Lemma 2. *The convergence of the proposed discrete volumetric harmonic mapping algorithm is guaranteed.*

PROOF. The proposed algorithm is to compute the harmonic map f of a giving volumetric ball M, which is a simply connected 3-manifold with a single boundary. The boundary surface of the volume is a topological sphere.

The algorithm is equivalent to use heat diffusion to solve a Laplace equation with Dirichlet boundary condition on a volume:

$$\begin{cases} \Delta f = 0 & \text{in } M \\ f = g & \text{on } \partial M, \end{cases}$$

where g map ∂M to S^2 in our algorithm.

According to elliptic PDE theory [39], the solution is the minimizer of the harmonic energy,

$$E(f) = \int_M < \nabla f, \nabla f > .$$

Its definition in discrete case is given by Eqn. (4).

If the boundary is smooth, the energy is convex [39]. Since we have mapped the boundary surface continuously to a unit sphere, the solution exists and unique. The proposed algorithm uses gradient descent method to minimize the energy, which is exactly the heat flow. Therefore, the iteration process is guaranteed to converge. If M has a hole inside, it is still solving the Laplace equation with smooth Dirichlet boundary conditions. Its convergence is still guaranteed. □