

Circular Sailing Routing for Wireless Networks

Fan Li Yu Wang

The University of North Carolina at Charlotte, USA

Email: {fli, yu.wang}@uncc.edu

Abstract—Routing in wireless networks has been heavily studied in the last decade and numerous routing protocols were proposed in literature. The packets usually follow the shortest paths between sources and destinations in routing protocols to achieve smallest traveled distance. However, this leads to the uneven distribution of traffic load in a network. For example, wireless nodes in the center of the network will have heavier traffic since most of the shortest routes go through them. In this paper, we first describe a novel routing method, called *Circular Sailing Routing* (CSR), which can distribute the traffic more evenly in the network. The proposed method first maps the network onto a sphere via a simple stereographic projection, and then the route decision is made by the distance on the sphere instead of the Euclidean distance in the plane. We theoretically prove that for a network the distance traveled by the packets using CSR is no more than a small constant factor of the minimum (the distance of the shortest path). We then extend CSR to a localized version, *Localized CSR*, by modifying the greedy routing without any additional communication overhead. Finally, we further propose CSR protocols for 3D networks where nodes are distributed in a 3D space instead of a 2D plane. For all proposed methods, we conduct simulations to study their performances and compare them with global shortest path routing or greedy routing.

I. INTRODUCTION

Routing is one of the key topics in wireless networks and has been well studied recently. Many routing protocols were proposed for different purposes. For example, there are power efficient routing for better energy efficiency, cluster-based routing for better scalability and geographical routing to reduce the overhead. In this paper, we are interested in designing a load balancing routing for large wireless networks. By spreading the traffic across the wireless network via the elaborate design of the routing algorithm, load balancing routing averages the energy consumption. This extends the lifespan of the whole network by extending the time until the first node is out of energy. Load balancing is also useful for reducing congestion hot spots thus reducing wireless collisions. Notice that there are already several load balancing routing protocols [1]–[3] in literature. However, most of them try to dynamically adjust the routes to balance the real time traffic load based on the knowledge of current load distribution (or current remaining energy distribution), which is not very scalable for large wireless networks. Here, we assume that individual node does not know the current load and each node may want to talk with all other nodes. We then address how to design load balancing routing for all-to-all communication scenario in a network.

Notice that most of routing protocols are based on shortest path algorithm where the packets are traveled via the shortest

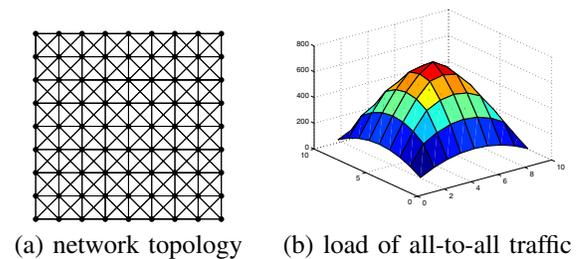


Fig. 1. In a grid network, nodes in the center area have much heavier traffic load than nodes in other areas. (SPR is applied for all possible pairs of nodes.)

path between a source and a destination. Even for the geographical localized routing protocols, such as greedy routing, the packets usually follow the shortest paths when the network is dense and uniformly distributed. In greedy routing, the packet is forwarded to the neighbor which is nearest to the destination. Taking the shortest path can achieve smaller delay or traveled distance, however it can also lead to the uneven distribution of traffic load in a network. For example, nodes in the center of a network will have heavier traffic since most of the shortest routes go through them. This is just like the transportation system around a big city where the downtown area is always the “hot spot”. Fig. 1 shows a simulation result on this scenario. The network is distributed on a 9×9 grid, and the network topology is shown in Fig. 1(a). Consider an all-to-all communication scenario, *i.e.*, each node sends one packet to all other nodes using Shortest Path Routing (SPR) algorithm. Fig. 1(b) illustrates the cumulative traffic load (*i.e.*, number of packets passing through) for each node. It is clear that nodes in the center area have much higher traffic load than nodes in other areas, therefore nodes in the center will run out of their batteries very quickly.

To avoid the uneven load distribution of shortest path routing, we focus on designing routing protocols for wireless networks which can achieve both small traveled distance and evenly distributed load in the network. Inspired from circular sailing (or called globular sailing), which sails on the arc of a great circle to make the shortest distance between two places on the earth, we propose a new routing algorithm called *Circular Sailing Routing* (CSR). In CSR, wireless nodes in a 2D network are mapped to a sphere using reversed stereographic projection¹ and the routing decision is made based on the circular distance on the sphere instead of the

¹Stereographic projection is a certain mapping that projects a sphere onto a plane. More detail is given in Section II.

Euclidean distance in 2D plane. By doing so, the traffic from one side to another side of the network area will avoid the center area. Thus, “hot spots” are eliminated and the load is balanced.

We also prove the competitiveness of CSR. Given a routing method \mathcal{A} , let $\mathbf{P}_{\mathcal{A}}(\mathbf{s}, \mathbf{t})$ be the path found by \mathcal{A} to connect the source node \mathbf{s} and the target node \mathbf{t} . A routing method \mathcal{A} is called l -competitive if for every pair of nodes \mathbf{s} and \mathbf{t} , the total length of path $\mathbf{P}_{\mathcal{A}}(\mathbf{s}, \mathbf{t})$ is within a constant factor l of the shortest path connecting \mathbf{s} and \mathbf{t} in the network. We can prove that CSR is $\frac{\pi}{2}(1 + \epsilon)$ -competitive. In other words, it can guarantee the total *distance* traveled by packets is constant competitive even in the worst case. Here, ϵ is an adjustable constant parameter defined in the stereographic projection.

In addition, CSR can be easily implemented using any position-based routing protocols (such as SPR or greedy routing). The only modification is a simple mapping calculation of the position information. There are no changes to the communication protocol and no any additional communication overhead.

The rest of the paper is organized as follows. In Section II, we first introduce stereographic projection, which comes from the field of projective geometry. Then we present our *Circular Sailing Routing* (CSR) protocol, prove its competitiveness, and compare its performance with Shortest Path Routing via simulations in Section III. In Section IV, we extend CSR to a localized version (LCSR) and compare its performance with greedy routing. In Section V, we further extend CSR and LCSR to 3D versions for 3D networks. We review related work in Section VI and conclude our paper in Section VII.

II. STEREOGRAPHIC PROJECTION

In projective geometry, the *stereographic projection* [4] is a certain mapping that projects a sphere onto a plane. Intuitively, it gives a planar picture of the sphere. The projection is defined on the entire sphere, except at one point – the projection point. Where it is defined, the mapping is smooth and bijective. It is also conformal, meaning that it accurately represents angular relationships (*i.e.*, local angles on a sphere are mapped to the same angles in the projection). On the other hand, it does not accurately represent area, especially near the projection point. Stereographic projection finds usage in many fields including cartography, geology, and crystallography. Sarkar *et al.* [5] first applied stereographic projection in wireless networks. They proposed a double rulings scheme for information brokerage in sensor networks where data replica are stored at a curve (a circle on the sphere), and the consumer travels along another curve which is guaranteed to intersect with the producer curve. In this paper, we use a reversed stereographic projection² to map wireless nodes in a 2D plane onto a 3D sphere and use the spherical distance as the routing metric.

Fig. 2(a) and (b) show two approaches to perform stereographic projection and they place the plane differently. In this paper, we use the first approach. As shown in Fig. 2(a), we

²When the context is clear, we ignore the word of “reversed”.

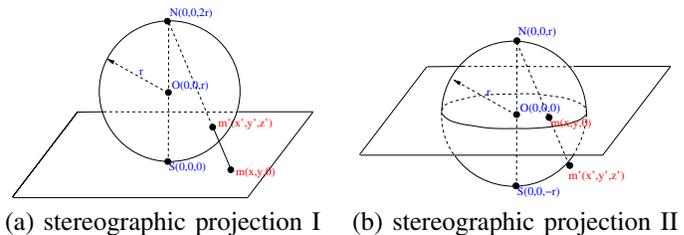


Fig. 2. Stereographic projections from a sphere to a plane: one-to-one mappings from a node m' on a sphere S to a node m in a plane.

put a sphere with radius r tangent to the plane at the origin $(0, 0, 0)$. Denote this tangent point as the south pole S and its antipodal point as the north pole N . A point m on the 2D plane is mapped to m' on the sphere, which is the intersection of the line through m and N and the sphere. This provides a one-to-one mapping from the 2D projective plane to a 3D sphere. Notice that stereographic projection preserves circles and angles. That is, a circle on the sphere is a circle in the plane and the angle between two lines on the sphere is the same as the angle between their projections in the plane.

We use $\|uv\|$ and $d(u'v')$ to represent the Euclidean distance between two points u and v and the geodesic shortest distance between two points u' and v' on the sphere, respectively. By simple geometric calculations, we can compute the 3D position on the sphere via the reversed projection by the following method.

Method 1: Given a node m with position (x, y) in the 2D plane, the 3D position of its reversed stereographic projection point m' is (x', y', z') , where $x' = \frac{4r^2x}{x^2+y^2+4r^2}$; $y' = \frac{4r^2y}{x^2+y^2+4r^2}$; and $z' = \frac{2r(x^2+y^2)}{x^2+y^2+4r^2}$.

Fig. 3 illustrates examples of reverse stereographic projection of a 81-nodes grid network (9×9 grid in a 20×20 square area). We use the position of the center node as the tangent point where the sphere is put. Nodes in the grid network are mapped to nodes on the sphere. Here, we try different sizes of the sphere (with radii 2, 5 or 10). It is clear that the size of the sphere affects the distribution of the mapped nodes on the sphere. With a larger sphere ($r = 10$), the mapped nodes are all nearer to the south pole in the lower half sphere and have similar distribution of the original grid network. With a smaller sphere ($r = 2$), more nodes are mapped to the upper half sphere. With $r = 5$ which is near the half of the radius of the grid network, all nodes are mapped to the lower half sphere more evenly.

The great circle distance (or called geodesic shortest distance) is the shortest distance between any two points on the surface of a sphere measured along a path on the surface of the sphere. Now we show how to compute the geodesic shortest distance between two points on the sphere. See Fig.4 for illustration. The shortest distance of m' and n' on the surface is the arc distance $d(m'n')$ along the greatest circle defined by the positions of m' , n' and O . Given the positions of m' and n' , we can easily get $\|Om'\|$, $\|On'\|$, and $\|m'n'\|$. Then, $\theta = \arccos \frac{\|Om'\|^2 + \|On'\|^2 - \|m'n'\|^2}{2\|Om'\|\|On'\|}$, and thus, $d(m'n') = \theta r$.

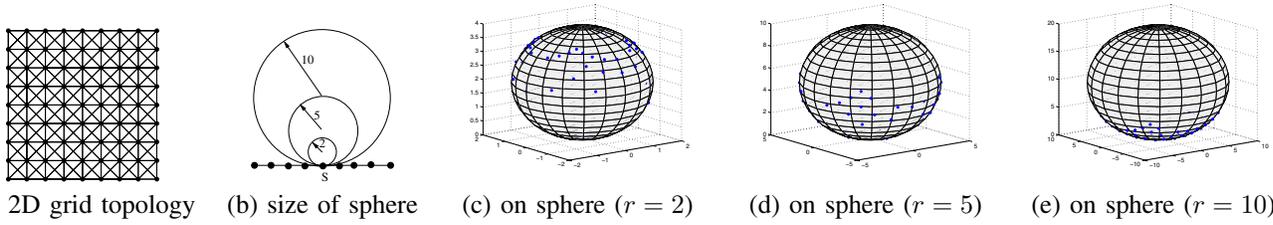


Fig. 3. The reversed stereographic projections of a grid network (9 × 9 grid in a 20 × 20 square area) to the sphere with various radii (2, 5 and 10).

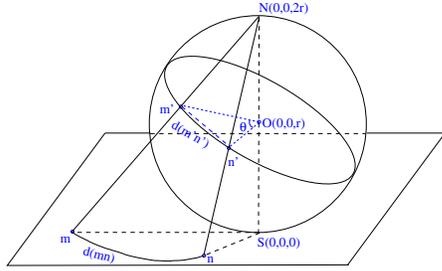


Fig. 4. The shortest distance between two points m' and n' on the sphere is the shorter segment of the greatest circle between m' and n' .

Actually, there are several one-to-one projections to map points on a sphere to points in the plane. Besides stereographic projections (Fig. 2(a) and (b)), there are area-preserving map projections, such as the Lambert azimuthal equal-area projection, which maintain size at the expense of shape. In this paper, we use stereographic projection in our scheme and remark that other spherical mapping can also be used.

III. CIRCULAR SAILING ROUTING

In this section, we first present our Circular Sailing Routing (CSR) based on stereographic projection, and then give both theoretical analysis on the competitiveness of CSR and simulation results of CSR compared with the shortest path routing.

A. Routing Algorithm

The stereographic projection maps an infinite plane onto a sphere. For a wireless network, the area in which the wireless nodes lie corresponds to a finite region of the plane. Let this region be called \mathbb{P} . With the information of the network region, we can place the south pole S of a sphere \mathbb{S} at the center of the network, whose coordinate is $(0, 0, 0)$. The radius r of \mathbb{S} is an adjustable parameter for our proposed routing method. Here, we assume each node knows the radius r of the projection sphere. This can be done via either a pre-set before the deployment or a broadcast operation after the deployment. Any point $m(x, y, 0)$ in \mathbb{P} maps to $m'(x', y', z')$ on the sphere \mathbb{S} . It is a one-to-one mapping, where $z' \leq k$ for some $0 < k < 2r$. Here k is the z' value of the highest projection on the sphere.

The basic idea of circular sailing routing is letting packet follow the circular shortest paths on the sphere instead of the Euclidean shortest paths in 2D plane. Because there is no hot

spot on the sphere where most of the circular shortest paths must go through, we expect circular sailing routing can achieve better load balancing than shortest path routing. The detailed routing algorithm is given as follows:

Algorithm 1 Circular Sailing Routing

- 1: **Mapping:** Map each node $m(x, y, 0)$ in the 2D plane to a node $m'(x', y', z')$ on the sphere \mathbb{S} (using Method 1).
- 2: **New Metrics:** For any existing link mn between two nodes m and n in the network, calculate the shortest circular distance on the sphere between their projected nodes m' and n' (i.e., $d(m'n')$). We use $d(m'n')$ as the cost of link mn , and call it *circular distance*.
- 3: **Routing:** Applying general shortest path routing with circular distance as the routing metric, choose the route with smallest total circular distance.

B. Analysis of Competitiveness

We will prove the competitiveness of CSR, i.e., the total length of the path taken by CSR between the source s and the destination t is at most constant times of the length of the shortest path between them in the plane. Before giving the proof, we need to present some preliminaries for stereographic projection. Assume that the furthest wireless node is of distance D from the center (i.e., south pole of the sphere), then the z' value of the highest projection on the sphere (i.e., the value of k) is

$$k = z'_{max} = 2r \left(\frac{D}{\sqrt{D^2 + (2r)^2}} \right)^2 = \frac{2rD^2}{D^2 + 4r^2}.$$

As in [5], we choose $r = \frac{D}{2\sqrt{\epsilon}}$, $\epsilon > 0$, thus $k = \frac{2r\epsilon}{(1+\epsilon)}$.

Recall that circles on the sphere map to circles in the plane, thus the projection of a great circle on the sphere \mathbb{S} is also a circle in the plane. Let $d(m'n')$ be the distance of an arc C' from a node m' to a node n' along a great circle on the surface of \mathbb{S} , and $d(mn)$ be the distance of an arc C between m and n along the projection of the great circle in the plane (Fig. 5). Remember that $\|mn\|$ denotes the Euclidean distance between m and n in the plane. The next two lemmas show that $d(m'n')$ (or the Euclidean distance $\|mn\|$ between m and n) is not too much different from $d(mn)$ in the plane.

Lemma 2: Consider any two nodes m' and n' on the sphere \mathbb{S} with their projections in the plane m and n , we have

$$\|mn\| \leq d(mn) \leq \frac{2r}{2r - k} d(m'n') = (1 + \epsilon) d(m'n').$$

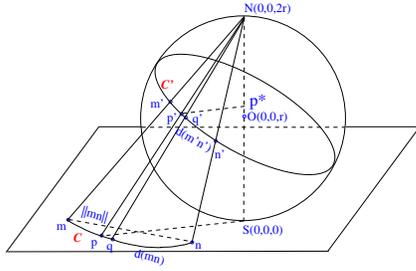


Fig. 5. The length of the projection $d(mn)$ is bounded by the great circle distance $d(m'n')$ on the sphere. $d(mn) \leq d(m'n')(1 + \epsilon)$.

Proof: First, since the Euclidean distance of two points is always smaller than the distance along any arc passing them, i.e., $\|mn\| \leq d(mn)$. Thus, we only need to prove $d(mn) \leq \frac{2r}{2r-k}d(m'n') = (1 + \epsilon)d(m'n')$.

Notice that it is one-to-one mapping between points on C' and points on C . $\int_{C'} dx' = d(m'n')$, where dx' is a miniature segment on C' . Similarly, $\int_C dx = d(mn)$, where dx is the projection of dx' in the plane. See Fig. 5 for illustration. $p'q'$ is a tiny segment on C' with length $dx' \rightarrow 0$, and $dx' = \|p'q'\|$. The projection of $p'q'$ is pq with the length $dx = \|pq\|$. Let p^* be the projection of p' on the line segment NS . The z' value of p^* (or p') is denoted by z_{p^*} . Then

$$\frac{\|Np^*\|}{\|NS\|} = \frac{2r - z_{p^*}}{2r}.$$

When $dx', dx \rightarrow 0$, i.e., $pq, p'q' \rightarrow 0$, we can look pq and $p'q'$ as in the same plane (the plane defined by nodes N, p and q), more specifically, the two arcs pass through pq and $p'q'$ are concentric at north pole N . Then,

$$\frac{dx'}{dx} = \frac{\|p'q'\|}{\|pq\|} = \frac{\|Np'\|}{\|Np\|} = \frac{\|Np^*\|}{\|NS\|} = \frac{2r - z_{p^*}}{2r}.$$

Because the highest value of z_{p^*} is k , we have

$$\frac{dx'}{dx} \geq \frac{2r - k}{2r} = \frac{2r - \frac{2r\epsilon}{(1+\epsilon)}}{2r} = \frac{1}{1 + \epsilon}.$$

Thus,

$$d(m'n') = \int_{C'} dx' \geq \int_C dx / (1 + \epsilon) = \frac{d(m, n)}{(1 + \epsilon)}.$$

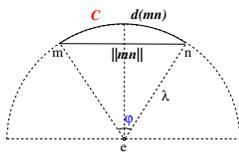


Fig. 6. Arc distance $d(mn)$ along a circle and Euclidean distance $\|mn\|$.

Lemma 3: Consider any two points m' and n' on the sphere with their projections on the plane m and n , we have

$$d(m'n') \leq d(mn) \leq \frac{\pi}{2} \|mn\|.$$

Proof: Similar to the proof of Lemma 2, assume that dx' is a miniature segment on C' and dx is the projection of dx' in the plane. From the proof of Lemma 2, we know $\frac{dx'}{dx} = \frac{2r - z'_p}{2r} \leq 1$. Thus, $dx' \leq dx$, and

$$d(m'n') = \int_{C'} dx' \leq \int_C dx = d(mn).$$

Fig. 6 shows a top view of the arc C in the plane \mathbb{P} . Arc C is a segment between m and n of a circle centered at e with the radius λ . Notice that e is not necessarily the center O of the sphere. Then we have $d(mn) = \varphi\lambda$ and $\|mn\| = 2\lambda \sin \frac{\varphi}{2}$, since the arc mn is the projection of the shortest circular distance between m' and n' on the sphere, so $0 < \varphi \leq \pi$. Then,

$$\frac{d(mn)}{\|mn\|} = \frac{\varphi\lambda}{2\lambda \sin \frac{\varphi}{2}} = \frac{\varphi}{2 \sin \frac{\varphi}{2}}.$$

When $\varphi = \pi$, $\frac{\varphi}{2 \sin \frac{\varphi}{2}}$ reaches its maximum value, $\frac{\pi}{2}$. Thus, $\frac{d(mn)}{\|mn\|} \leq \frac{\pi}{2}$. This concludes the proof: $d(m'n') \leq d(mn) \leq \frac{\pi}{2} \|mn\|$. ■

Now we are ready to prove the main theorem of this paper about the competitiveness of CSR. Recall that a routing method \mathcal{A} is called l -competitive if for every pair of nodes s and t , the total length of path $\mathbf{P}_{\mathcal{A}}(s, t)$ found by \mathcal{A} is within l times of the shortest path connecting s and t in the network. In other words, we want to prove CSR can find a path whose length is within a small constant factor of the minimum even in the worst case scenario. Hereafter, we call l the *Competitiveness Factor* (CF).

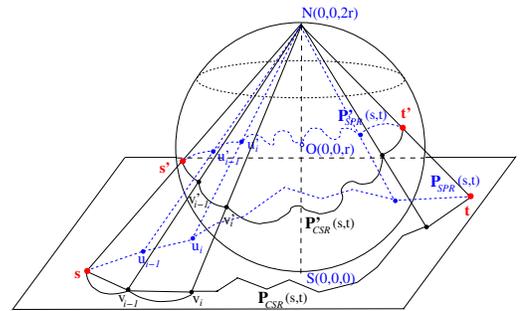


Fig. 7. The Euclidean path length of proposed CSR protocol is bounded by the Euclidean path length of shortest path routing.

There are four paths we will use in the proof. Fig. 7 illustrates their definitions and the relationship among them. The dotted line in the plane represents the shortest path generated by a shortest path routing connecting the source s and the destination t , denoted by $\mathbf{P}_{SPR}(s, t)$. The dotted line on the sphere is the surface path connecting all the projections on the sphere of each node along $\mathbf{P}_{SPR}(s, t)$ using the circular distance, denoted by $\mathbf{P}'_{SPR}(s, t)$. The solid line in the plane represents the path found by CSR protocol, denoted by $\mathbf{P}_{CSR}(s, t)$ and the solid line on the sphere is the surface path connecting all the projections of each node along $\mathbf{P}_{CSR}(s, t)$, denoted by $\mathbf{P}'_{CSR}(s, t)$. Notice that, in any two points along a path in the plane, the shortest distance is the

straight line connecting them, meanwhile the shortest circular distance of its projection on the sphere is a segment (an arc) of a great circle. For a path \mathbf{P}_A in the plane, we define $\|\mathbf{P}_A\|$ as the summation of the Euclidian distance of each link in \mathbf{P}_A . For a path \mathbf{P}'_A on the sphere, we define $\mathbf{d}(\mathbf{P}'_A)$ as the summation of the length of each arc in \mathbf{P}'_A .

Theorem 4: The distance traveled by CSR routing strategy is $\frac{\pi}{2}(1 + \epsilon)$ -competitive, *i.e.*,

$$\|\mathbf{P}_{CSR}(\mathbf{s}, \mathbf{t})\| \leq \frac{\pi}{2}(1 + \epsilon)\|\mathbf{P}_{SPR}(\mathbf{s}, \mathbf{t})\|.$$

Proof: Let $\mathbf{P}_{CSR}(\mathbf{s}, \mathbf{t}) = v_0, v_1, v_2, \dots, v_n$, where $v_0 = \mathbf{s}$ and $v_n = \mathbf{t}$. Let the projection of $\mathbf{P}_{CSR}(\mathbf{s}, \mathbf{t})$ on the sphere $\mathbf{P}'_{CSR}(\mathbf{s}, \mathbf{t}) = v'_0, v'_1, v'_2, \dots, v'_n$. Similarly, let $\mathbf{P}_{SPR}(\mathbf{s}, \mathbf{t}) = u_0, u_1, u_2, \dots, u_m$, where $u_0 = \mathbf{s} = v_0$ and $u_m = \mathbf{t} = v_n$. Let the projection of $\mathbf{P}_{SPR}(\mathbf{s}, \mathbf{t})$ on the sphere $\mathbf{P}'_{SPR}(\mathbf{s}, \mathbf{t}) = u'_0, u'_1, u'_2, \dots, u'_m$, where $u'_0 = \mathbf{s}' = v'_0$ and $u'_m = \mathbf{t}' = v'_n$.

From Lemma 2, we know $\|v_{i-1}v_i\| \leq (1 + \epsilon)d(v'_{i-1}v'_i)$, therefore, $\|\mathbf{P}_{CSR}(\mathbf{s}, \mathbf{t})\| = \sum_{i=1}^n \|v_{i-1}v_i\| \leq \sum_{i=1}^n (1 + \epsilon)d(v'_{i-1}v'_i) = (1 + \epsilon)\mathbf{d}(\mathbf{P}'_{CSR}(\mathbf{s}, \mathbf{t}))$. According to the CSR protocol, $\mathbf{d}(\mathbf{P}'_{CSR}(\mathbf{s}, \mathbf{t})) \leq \mathbf{d}(\mathbf{P}'_{SPR}(\mathbf{s}, \mathbf{t}))$ since $\mathbf{P}'_{CSR}(\mathbf{s}, \mathbf{t})$ is the shortest path using circular distance metric on the sphere. From Lemma 3, we have $d(u'_{i-1}u'_i) \leq \frac{\pi}{2}\|u_{i-1}u_i\|$. Thus, $\mathbf{d}(\mathbf{P}'_{SPR}(\mathbf{s}, \mathbf{t})) = \sum_{i=1}^m d(u'_{i-1}u'_i) \leq \sum_{i=1}^m \frac{\pi}{2}\|u_{i-1}u_i\| = \frac{\pi}{2}\|\mathbf{P}_{SPR}(\mathbf{s}, \mathbf{t})\|$. Consequently, we have

$$\begin{aligned} \|\mathbf{P}_{CSR}(\mathbf{s}, \mathbf{t})\| &\leq (1 + \epsilon)\mathbf{d}(\mathbf{P}'_{CSR}(\mathbf{s}, \mathbf{t})) \\ &\leq (1 + \epsilon)\mathbf{d}(\mathbf{P}'_{SPR}(\mathbf{s}, \mathbf{t})) \\ &\leq \frac{\pi}{2}(1 + \epsilon)\|\mathbf{P}_{SPR}(\mathbf{s}, \mathbf{t})\|. \end{aligned}$$

■

Theorem 4 gives a theoretical bound of the competitiveness of CSR protocol. It shows that the path length in CSR protocol is not too much different from the shortest path routing. Since $\epsilon = \frac{D^2}{4r^2}$, with the adjustable parameter r (*i.e.*, the radius of the sphere), we can control the competitiveness factor.

C. Simulation

We now evaluate the performance CSR via simulations for both grid networks and random networks. In both cases, wireless nodes are distributed in a 20×20 square area. In CSR, the south pole of the sphere is tangent at the center of this area. We try different sizes of the sphere (with radii 2, 5 or 10, as shown in Fig. 3). Clearly, the size of the sphere affects the distribution of the mapped nodes on the sphere.

Grid Networks: We first deploy the 81 nodes on a 9×9 grid and set the transmission range R of all nodes to 3. The resulted topology is shown in Fig. 3(a). We compare the performance of SPR and CSR under the all-to-all communication scenario where every pair of nodes in the network has unit message to communicate. Fig. 8(a) shows the distributions of each node's traffic load for both SPR and CSR when the radius of the sphere $r = 5$. It is clear that the load of CSR (the lower figure in Fig. 8(a)) is more evenly distributed than the load of SPR (the upper figure in Fig. 8(a)). The hot spot problem (center nodes with highest load) is avoided in CSR. Fig. 8(b) shows the average (Avg), maximum (Max) traffic load and standard

deviation (STD) of traffic load for all nodes in the network for SPR and CSR with different radii. The average traffic load of CSR are larger than SPR, especially when $r = 2$ (*i.e.*, most nodes are mapped to the upper half sphere). This is reasonable because the SPR has the least total traffic load than any other routing algorithms. Remember that SPR uses the shortest path for each pair of nodes. When $r = 5$ and 10, CSR has smaller maximum load and the STD of load is much less than SPR. Thus, CSR can balance the load traffic for each node (*s.t.*, the power consumptions of all nodes are more even). These results meet our design objective very well with only a little bit more average traffic load. We also find that when the nodes are mapped to the bottom half sphere (*i.e.*, $r = 5$), CSR has the best performance compared with other sizes of the sphere. When the radius is very large, the nodes are mapped to the area around the south pole, which has similar distribution with the original network. In such case, simulation results show that CSR's performance is similar to SPR on the original network.

We also study the competitiveness factor (CF) of CSR. From Theorem 4, the distance traveled by CSR satisfies $\|\mathbf{P}_{CSR}(\mathbf{s}, \mathbf{t})\| \leq \frac{\pi}{2}(1 + \epsilon)\|\mathbf{P}_{SPR}(\mathbf{s}, \mathbf{t})\|$, where $\epsilon = \frac{D^2}{4r^2}$. In our simulation settings, $D = 10\sqrt{2}$. Thus, when $r = 2, 5$ and 10, $CF = 21.2, 4.7$ and 2.4 , respectively. We measure the CF for each route generated by CSR in our simulation. Table I gives the average and maximum competitiveness factor (Avg CF and Max CF) of CSR with different radii. The simulation results of CFs confirm our theoretical bounds. Actually the practical CFs are much smaller than the bounds, and very close to 1. In other words, not only CSR has balanced traffic load but also the distance traveled by the packets is almost the same as the minimum (the distance of the shortest path).

TABLE I
COMPETITIVENESS FACTOR (CF) OF CSR (VARIOUS SPHERE SIZE)

Network Topology	Radius r	Avg CF	Max CF
Grid	2	1.2202	2.6485
	5	1.0085	1.2589
	10	1.0000	1.0000
Random	2	1.2050	3.0615
	5	1.0252	1.4121
	10	1.0016	1.0974

Random Networks: We also test the performance of CSR with random networks. 81 nodes are randomly deployed in the field with transmission range R set to 4. We run the simulation for 100 random networks and take the average. Fig. 8(c) and the lower half of Table I summarize the performance comparison of CSR for random networks. CSR ($r = 5$) has the best performance, *i.e.*, much smaller maximum load and load STD with little greater average load and the average CF very close to 1.0. CSR ($r = 10$) has similar performance with SPR because the mapped positions on the sphere are similar to those in the original network.

IV. LOCALIZED CIRCULAR SAILING ROUTING

The geometric nature of wireless networks allows the promising idea: localized routing protocols. In localized routing protocols, by assuming each node has position information,

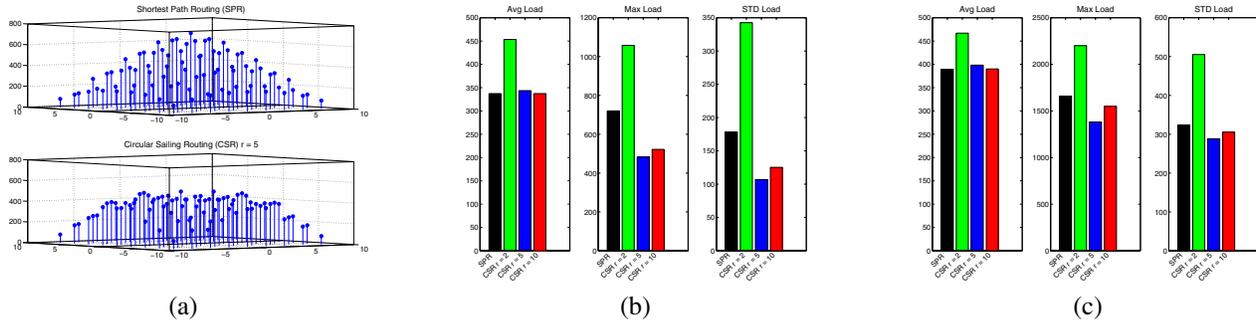


Fig. 8. Load of SPR and CSR: (a) Traffic load of SPR and CSR ($r = 5$) on a 9×9 grid; (b) Comparison of traffic load of SPR (black) and CSR on a 9×9 grid with $r = 2$ (green), 5 (blue) and 10 (red). (c) Comparison of traffic load of SPR and CSR on 81-nodes random networks with $r = 2, 5$ and 10.

the routing decision is made at each node by using only local neighborhood information. It does not need the dissemination of route discovery information, and no routing tables are maintained at each node. The most popular localized routing is *greedy routing* [6] where the current node u always finds the next relay node v such that the distance from v to the destination \mathbf{t} is the smallest among all neighbors of u . Our CSR is easy to be extended to a localized version.

A. Routing Algorithm

Similar to the classical greedy routing, the *Localized Circular Sailing Routing* (LCSR) just forwards the packet to the neighbor whose projection is closest to the projection of the destination on the sphere. Notice that each node only needs to know its neighbors' positions to make the routing decision. The detailed routing algorithm is given in Algorithm 2:

Algorithm 2 Localized Circular Sailing Routing

- 1: For each neighbor v , node u maintains both a 2D position of v in the plane and a 3D position of its projection v' on the sphere \mathbb{S} . Node u also maintains its own 2D position and its projection's 3D position.
 - 2: **while** node u receives a packet with destination \mathbf{t} **do**
 - 3: **if** $\|u\mathbf{t}\| \leq R$, where R is the transmission range **then**
 - 4: Forward the packet to \mathbf{t} directly and return.
 - 5: Map \mathbf{t} to its projection \mathbf{t}' (*i.e.*, get its 3D position).
 - 6: **if** $\exists v$, *s.t.*, its projection v' satisfies $d(v'\mathbf{t}') < d(u'\mathbf{t}')$ **then**
 - 7: Forward packet to node v with the minimum $d(v'\mathbf{t}')$.
 - 8: **else**
 - 9: Simply drop the packet.
-

If LCSR can find a neighbor to forward the packet at each step, it will guarantee to reach the destination in finite steps. The proof will be similar to the one for greedy routing. However, LCSR (Algorithm 2) cannot always find the forwarding neighbor, since it could fail into a local minimum where no such neighbor v exists. To solve this problem, we can switch to greedy routing to find a forwarding neighbor who is nearest to destination in 2D plane. If the greedy routing cannot find a forwarding neighbor either, face routing in the plane can be applied to get out of the local minimum as in [6], [7]. If

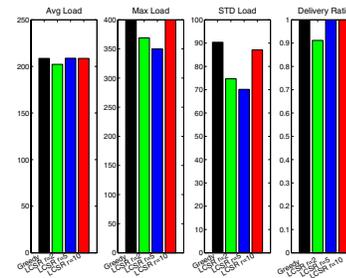


Fig. 9. Traffic load of Greedy Routing and LCSR on a 9×9 grid network.

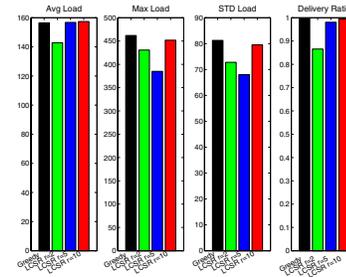


Fig. 10. Traffic load of Greedy Routing and LCSR in random networks.

the packet reaches a location whose projection is closer to the projection of the destination than the projection of the position where the previous LCSR has failed, then LCSR is resumed.

B. Simulation

We test the performance of LCSR algorithm by using the same grid and random networks which are used in Section III-C. We also assume all-to-all communication in the networks. Classical greedy routing is used for comparison. For simplicity, in the simulation, we implement LCSR without any recovery mechanisms, *i.e.*, LCSR (Algorithm 2) simply drops the packet at the local minimum.

Fig. 9 and Fig. 10 show the performance comparison of Greedy Routing and LCSR for the grid networks and random networks, respectively. Here, the data for random networks is the average value of 50 random generated networks. It is clear that LCSR with $r = 5$ has the best performance, *i.e.*, smallest maximum traffic load and STD load for both grid and random networks. The delivery ratio is 100% and almost 100% for grid

and random networks, respectively. For example for the grid network, the max_load of Greedy Routing is 400 while the max_load of LCSR ($r = 5$) is 350, which is reduced about 12.5%. The STD_load is also decreased by about 22.2%(from 90 to 70). The avg_load of LCSR ($r = 5$) and Greedy Routing are at the same value of 208. Again, LCSR ($r = 10$) has very similar performance with greedy routing, since the larger the sphere, the more alike the distribution on the sphere to the original 2D distribution.

We also measure the CF of LCSR. Here, CF is the factor between the distance traveled by the packet in CSR and the distance traveled in greedy routing, if both routing methods can find a path between the source and the destination. In the simulation, we randomly select 100 routes (10 source nodes and 10 destination nodes are randomly chosen) and calculate the CF for each route. Table II gives the results for both grid and random networks. Though we do not have any proof of theoretical bounds, the CFs are very small in practice.

TABLE II
COMPETITIVENESS FACTOR (CF) OF LCSR (VARIOUS SPHERE SIZE)

Network Topology	Radius r	Avg CF	Max CF
Grid	2	1.0100	1.2761
	5	1.0073	1.2071
	10	1.0036	1.1380
Random	2	1.1059	2.7769
	5	1.0197	1.6072
	10	1.0013	1.3469

V. 3D CIRCULAR SAILING ROUTING

So far we consider routing in a 2D network and how to map the nodes onto a sphere so that routing along the sphere can balance the load. The assumption of 2D network may no longer be valid if a wireless network is deployed in space, atmosphere, or ocean, where nodes of a network are distributed over a 3D space and the difference in the third dimension is too large to be ignored. In fact, recent interest in underwater acoustic sensor networks hints at the strong need to understand how to design networks in 3D. In a 3D network, the problem of uneven load distribution also exists. Fig. 11(a) shows a 3D grid network with 216 nodes. If we apply classic greedy routing for every pair of nodes, the total node load is plotted in the left sub-figure of Fig. 11(b). Clearly, the center nodes of each level have larger load and the two middle levels have larger load than the top/bottom levels. Therefore, we are also interested in developing load balance routing for 3D networks. The right sub-figure of Fig. 11(b) shows the load from our new 3D Localized CSR (3D-LCSR) method which can achieve better traffic distribution under all-to-all communication.

A. Routing Algorithm

Fortunately, the idea of circular sailing routing can also be extended to 3D. Instead of mapping a plane to the surface of a sphere, 3D-CSR (or 3D-LCSR) maps nodes in a 3D region to the surface of a sphere. All the routing algorithms (3D-CSR or 3D-LCSR) in 3D networks are the same as 2D-CSR and 2D-LCSR (Algorithm 1 and 2), except for the projection method and the definition of surface distance $d(m'n')$.

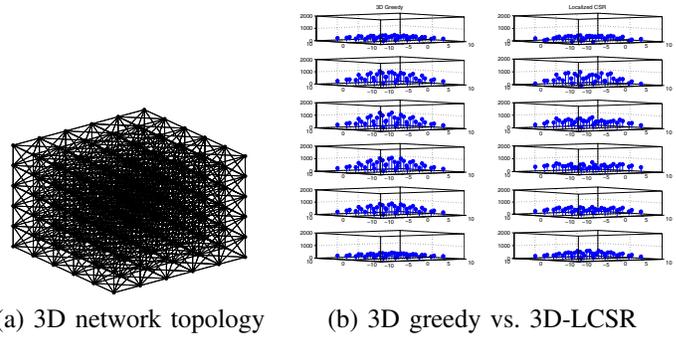


Fig. 11. Balance the load in 3D network: (a) A 3D grid network with 6 levels of 6×6 grids. (b) Traffic load of each node in the 6 levels for 3D Greedy Routing and 3D-LCSR. For 3D Greedy (Left Sub-figure), $avg_load = 479$, $max_load = 1219$, $STD_load = 250$. For 3D-LCSR (Right Sub-figure), $avg_load = 431$, $max_load = 948$, $STD_load = 171$. Clearly, 3D-LCSR achieves better load balancing than 3D Greedy Routing.

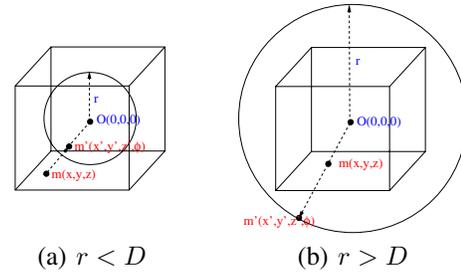


Fig. 12. 3D surface mapping with various sizes of spheres: from a node $m(x, y, z)$ in 3D space to a node (x', y', z', ϕ) on the sphere.

For a 3D wireless network, wireless nodes are distributed in a finite 3D region \mathbb{R} . With the information of the network region, we can place the center O of a sphere \mathbb{S} at the center of the network, whose coordinate is $(0, 0, 0)$. The radius r of \mathbb{S} is again an adjustable parameter. Assume that the furthest wireless node is of distance D from the center. Fig. 12 shows two cases: $r < D$ and $r \geq D$. Any point $m(x, y, z)$ in \mathbb{R} maps to $m'(x', y', z', \phi)$ on the sphere \mathbb{S} . Here (x', y', z') is the 3D position of the projection node m' , and ϕ is the distance from m to the center O . As shown in Fig. 12, m' is the intersection point of sphere \mathbb{S} and line mO . It is easy to show that the position of m' can be computed by the following equations: $\phi = \sqrt{x^2 + y^2 + z^2}$, $x' = \frac{r}{\sqrt{x^2 + y^2 + z^2}}x$, $y' = \frac{r}{\sqrt{x^2 + y^2 + z^2}}y$, and $z' = \frac{r}{\sqrt{x^2 + y^2 + z^2}}z$. Notice that we need to specially deal with the node $(0, 0, 0)$ to guarantee that the projection is a one-to-one mapping. Here, we force to map it to $(0, 0, r, 0)$.

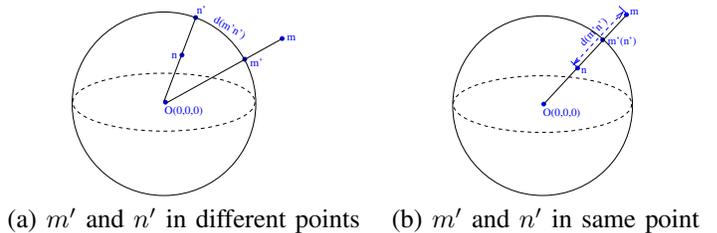


Fig. 13. Two cases of calculation of the distance $d(m'n')$.

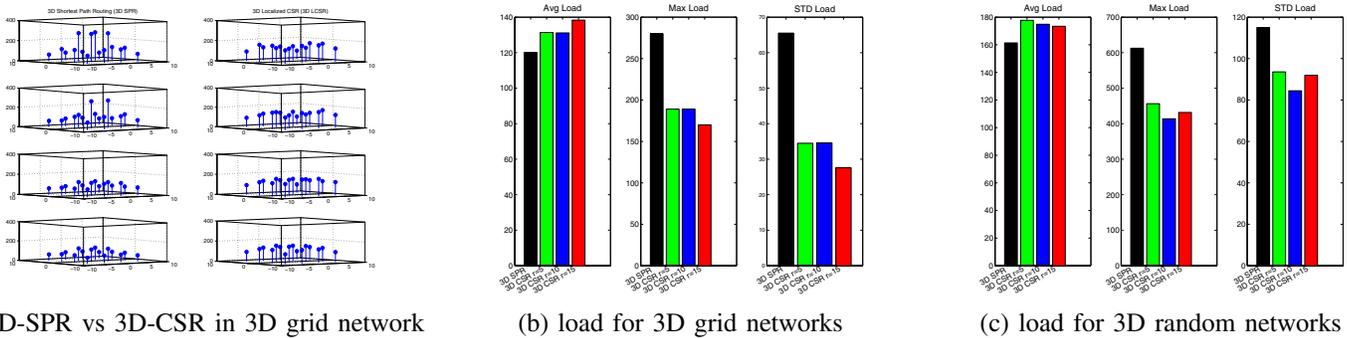


Fig. 14. Traffic load of 3D-SPR and 3D-CSR for both 3D grid and 3D random networks. (a) Traffic load of each node for a 3D grid network which has 4 levels of 4×4 grids. For 3D-SPR, $avg_load = 120$, $max_load = 280$, $STD_load = 65$. For 3D-CSR ($r = 15$), $avg_load = 138$, $max_load = 170$, $STD_load = 27$. (b) Comparison of traffic load of 3D-SPR (black) and 3D-CSR on the 3D grid with $r = 5$ (green), 10 (blue) and 15 (red). (c) Comparison of traffic load of 3D-SPR and 3D-CSR on 64-nodes 3D random networks with $r = 5, 10$ and 15.

To calculate the distance $d(m'n')$ between two projections $m'(x'_m, y'_m, z'_m, \phi_m)$ and $n'(x'_n, y'_n, z'_n, \phi_n)$, there are two cases. If m' and n' are in different positions on the sphere, $d(m'n')$ is the distance of the shorter arc between m' and n' of the greatest circle defined by m' , n' and o on the sphere, which can be computed by $d(m'n') = r \arccos \frac{||Om'||^2 + ||On'||^2 - ||m'n'||^2}{2||Om'||||On'||}$. See Fig. 13(a) for reference. In the second case, m' and n' are at the same point on the sphere (i.e., $x'_m = x'_n$, $y'_m = y'_n$ and $z'_m = z'_n$), $d(m'n')$ is the Euclidean distance between nodes m and n , i.e., $d(m'n') = ||mn|| = |\phi_m - \phi_n|$ (Fig. 13(b)).

B. Simulation

We also conduct extensive simulation for 3D-CSR and 3D-LCSR. Again both 3D grid networks and 3D random networks are tested. For 3D grid networks, we use $4 \times 4 \times 4$ grids in a $20 \times 20 \times 20$ cube. The number of nodes is 64 and transmission range of each node is set to 6. For 3D random network, 64 nodes are randomly distributed in a $20 \times 20 \times 20$ cube. The transmission range is set to 8. We also try different sizes of the sphere (with radii 5, 10 and 15). The furthest node is of distance $D = 10\sqrt{2} = 14.1$ from the center of the network cube (or projection sphere). When $r = 15$, the projection sphere is a little bit larger than the network cube.

3D-CSR vs 3D-SPR: Fig. 14(a) is the distribution of traffic load of each level when $r = 15$ for the grid networks. In 3D-SPR (left sub-figure), the upper two levels have heavier traffic load (e.g., maximum load is around 280) and the center nodes in each levels have heavier traffic load. However, in 3D-CSR, the maximum load of upper two levels is only around 170 and load is more evenly distributed in each levels. Therefore, 3D-CSR has more evenly distributed traffic and the maximum value is much smaller than 3D-SPR. Fig. 14(b) demonstrates the average, maximum traffic load, and STD of load for 3D-SPR and 3D-CSR with different sphere sizes ($r = 5, 10$ and 15) for the grid networks. 3D-CSR with all three different sphere sizes have much smaller maximum load than 3D-SPR. The traffic is more evenly distributed, and the average CF is only around 1.05 to 1.10 (Table III). Notice that unlike the 2D-CSR, we can not easily find a specific size of the

TABLE III
COMPETITIVENESS FACTOR (CF) OF 3D-CSR (VARIOUS SPHERE SIZE)

Network Topology	Radius r	Avg CF	Max CF
Grid	5	1.0943	3.1462
	10	1.0927	3.1462
	15	1.0598	2.0731
Random	5	1.0880	2.9047
	10	1.0933	3.0388
	15	1.0867	2.5823

sphere which leads to the best performance, because 3D-CSR uses a different mapping technique. The mapped position is the intersection point of the sphere and a line starts from the center of the sphere instead of the north pole of the sphere in stereographic projection, thus the size of the sphere does not have significant effect on the performance.

Fig. 14(c) shows the results for random networks generated by the average of 100 random 3D networks. Similarly, 3D-CSR with all three sizes have smaller maximum load and STD of traffic load compared with 3D-SPR, and their CFs (lower half of Table III) are still competitive.

3D-LCSR vs 3D-Greedy: The performance comparison between localized version of 3D-CSR (denoted by 3D-LCSR) and 3D greedy for grid network have been demonstrated in Fig.11(b). We then study the performance of 3D-LCSR in random networks. Fig. 15 and Table IV show that 3D-LCSR has smaller average, maximum load and STD of traffic load than 3D greedy routing for all three sizes, and their average and maximum CFs are still very small. For example, the max_load of 3D-Greedy routing is 293 while it is only 235 for 3D-LCSR ($r = 15$), it has been decreased by about 19.8%. The STD_load has been decreased by about 25.5% (from 55 to 41), meanwhile 3D-LCSR has almost the same avg_load as 3D greedy routing and its delivery ration is around 96%.

TABLE IV
COMPETITIVENESS FACTOR (CF) OF 3D-LCSR (VARIOUS SPHERE SIZE)

Network Topology	Radius r	Avg CF	Max CF
Random	5	1.0514	2.4368
	10	1.0558	2.5463
	15	1.0495	2.7630

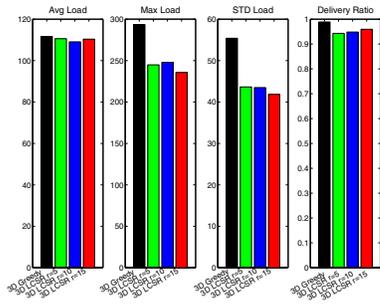


Fig. 15. Load of 3D-LCSR and 3D Greedy Routing for 3D random networks.

In summary, even for 3D networks, CSR has small CF and more balanced traffic load than previous routing method.

VI. RELATED WORK

Load balancing routing for wireless networks has been studied in [1]–[3]. Both [1] and [2] used the traffic load of the intermediate nodes as the main route selection criteria. In the routing construction phase, each intermediate node records its current traffic load in the control packet. The destination uses this information to select the least loaded route. Another simple load balancing approach was proposed in [3] which allows each node to drop RREQ or give up packet forwarding depending on its own traffic load. If the traffic load is high, node may deliberately give up packet forwarding to save its own energy. Some other load-aware routing methods use different load metrics, such as contention information, expected transmission count. Multi-path routing [10]–[12] was also used for load balancing. However, [13] showed unless using a very large number of paths the load distribution is almost the same as single path routing. All of the above methods are different from our proposed method. Unlike them, we assume that each node does not know the current load information, and our approach focus on balance the load for the whole network under all-to-all traffic scenario using a novel geometric technique.

Global load balancing in fixed networks has also been studied [8], [9]. The proposed methods usually define the load balancing routing as flow problems and use integer linear programming to solve them. Even they can optimally handle arbitrary traffic distribution (other than all-to-all unit traffic here), these methods are too complex for wireless systems with small devices (e.g. sensor networks) or large dense networks.

Recently, Hyttiä and Virtamo [14] also studied how to avoid the crowded center problem by analyzing the load probability in a dense network. They proposed a randomized choice between shortest path and routing on inner/outer radii to level the load. During the submission period of this paper, Popa *et al.* [15] published a similar routing technique, called *curveball routing*, which also maps the 2D network onto a sphere and route the packets based on nodes' virtual coordinates on the sphere. However, they used the different stereographic projection method as shown in Fig. 2(b) and did not give any theoretical analysis of the competitiveness of their routing method. Gao and Zhang [16] discussed the tradeoffs between

the competitiveness factor and load balancing ratio in routing on certain type of graphs (namely, growth restricted graphs).

VII. CONCLUSION

In this paper, we proposed a set of novel routing protocols, called *Circular Sailing Routing* for wireless networks to avoid the uneven load distribution caused by shortest path routing or greedy routing. By spreading the traffic across a virtual 3D sphere which is mapped from the network, CSR (LCSR) can reduce hot spots in the networks and increase the energy lifetime of the network. CSR can be easily implemented using any existing position-based routing protocols without any major changes or additional overhead. The only modification is a simple mapping calculation of the position information. In this paper, we not only provided a theoretical proof of the competitiveness of CSR, but also conducted extensive simulations to evaluate the proposed protocols. In addition, we presented the extended versions of our proposed methods (3D-CSR and 3D-LCSR) for 3D wireless networks. We leave theoretical analysis of the load distribution of CSR as our future work.

ACKNOWLEDGMENT

The authors would like to thank Xiang-Yang Li for the helpful discussions which inspired this work as well as Jie Gao for providing the detailed proof of one theorem in [5]. This work was supported in part by the US National Science Foundation (NSF) under Grant No. CNS-0721666 and funds provided by the University of North Carolina at Charlotte.

REFERENCES

- [1] Sung-Ju Lee and Mario Gerla, "Dynamic load-aware routing in ad hoc networks," in *Proc. of IEEE ICC*, 2001.
- [2] H. Hassanein and A. Zhou, "Routing with load balancing in wireless ad hoc networks," in *Proc. of 4th ACM MSWIM*, 2001.
- [3] Y. Yoo and S. Ahn, "A simple load-balancing approach in cheat-proof ad hoc networks," in *Proc. of IEEE Globecom*, 2004.
- [4] H.S.M. Coxeter, *Introduction to Geometry*, John Wiley & Sons, 1969.
- [5] R. Sarkar, X. Zhu, and J. Gao, "Double rulings for information brokerage in sensor networks," in *Proc. of ACM MobiCom*, 2006.
- [6] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," *ACM/Kluwer Wireless Networks*, vol. 7, no. 6, pp. 609-616, 2001.
- [7] B. Karp and H.T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proc. of ACM MobiCom*, 2000.
- [8] Y. Wang and Z. Wang, "Explicit routing algorithms for internet traffic engineering," in *Proc. of IEEE ICCCN*, 1999.
- [9] L. Fratta, M. Gerla, and L. Kleinrock, "The flow deviation method: an approach to store-and-forward communication network design," *Networks*, vol. 3, no. 2, pp. 97-133, 2007.
- [10] M.R. Pearlman, *et al.*, "On the impact of alternate path routing for load balancing in mobile ad hoc networks," in *Proc. of ACM MobiHoc*, 2000.
- [11] P.P. Pham and S. Perreau, "Performance analysis of reactive shortest path and multi-path routing mechanism with load balance," in *Proc. of IEEE Infocom*, 2003.
- [12] S. Yin and X. Lin, "Adaptive load balancing in mobile ad hoc networks," in *Proc. of IEEE WCNC*, 2005.
- [13] Y. Ganjali and A. Keshavarzian, "Load balancing in ad hoc networks: single-path routing vs. multi-path routing," in *Proc. of Infocom*, 2004.
- [14] E. Hyttiä and J. Virtamo, "On traffic load distribution and load balancing in dense wireless multihop networks," *EURASIP Journal on Wireless Communications and Networking*, 2007.
- [15] L. Popa, A. Rostamizadeh, *et al.*, "Balancing traffic load in wireless networks with curveball routing," in *Proc. of ACM MobiHoc*, 2007.
- [16] J. Gao and L. Zhang, "Tradeoffs between stretch factor and load balancing ratio in routing on growth restricted graphs," in *Proc. of ACM PODC*, 2004.