VirtualFace: An Algorithm to Guarantee Packet Delivery of Virtual-Coordinate-Based Routing Protocols in Wireless Sensor Networks

Ming-Jer Tsai, Fang-Ru Wang, Hong-Yen Yang, and Yuan-Po Cheng
Department of Computer Science
National Tsing Hua University
Hsing Chu, Taiwan, ROC
E-mail: mjtsai@cs.nthu.edu.tw

Abstract—Because the global positioning system (GPS) consumes a large amount of power and does not work indoors, many virtual-coordinate-based routing protocols are proposed for wireless sensor networks in which geographic location information is unavailable. Each of them, however, cannot guarantee packet delivery or constructs a virtual coordinate system with a complex structure. In this paper, we propose a method capable of augmenting virtual-coordinate-based routing protocols to guarantee packet delivery. Firstly, we introduce the virtual face construction protocol and the virtual face naming protocol to construct and name virtual faces, respectively. Subsequently, the VirtualFace algorithm is presented to route a packet from a dead-end node to a progress node by traversing the boundaries of the virtual faces from face to face. Simulations show that virtual-coordinate-based routing protocols including GLIDER, Hop ID, GLDR, and VCap augmented with the VirtualFace algorithm guarantee packet delivery while ensuring moderate routing path length overhead costs.

I. INTRODUCTION

In a wireless sensor network, the sensor monitors and aggregates information about the environment, and returns the environmental information to the base station via inter-sensor communication. A wide range of applications exist in our lives, including environmental monitoring, battlefield surveillance, health care, intruder detection, etc. The design of routing protocols has received considerable attention because an efficient routing protocol significantly improves network performance. Because the global positioning system (GPS) consumes a large amount of power, a geographic location is obtained with difficulty in wireless sensor networks. In this paper, we undertake the development of virtual-coordinate-based routing protocols in wireless sensor networks.

MAP [1] constructs the media axes, and uses a medial axis graph that represents the connections between the media axes as the guide for routing. In MAP, several sample boundary nodes surrounding voids and several sample network boundary nodes must be handily picked, generating difficulty with implementing this approach in practice. ABVCap [2] constructs axes, including a parallel of latitude and a number of meridians, and assigns each node the longitude and latitude coordinates in a manner analogous to that for the degrees of longitude and latitude in the globe. MAP and ABVCap each guarantee packet delivery; however, each of them constructs a complex structure consisting of a number of axes, demanding a considerable amount of message communication overhead to reconstruct disconnected axes in networks with node failures.

In GLIDER [3], all nodes are divided into cells by the landmark Voronoi complex and the connections between neighboring cells are represented by combinatorial Delaunay triangulation (CDT) on landmarks. A node is addressed by the hop distances to the landmarks in neighboring cells, and a packet is routed via the shortest path from the source cell to the destination cell in CDT. In GLDR [4], a node is addressed by the hop distances to a constant number of nearest landmarks. A packet is routed along the shortest path toward the landmark near the destination which demonstrates the maximum ratio of the hop distances to the source and the destination. In VCap [5] and Hop ID [6], a node is addressed by the hop distances to all landmarks, and a packet is greedily routed to the destination using the power distance computed by virtual coordinates. In GLIDER, GLDR, VCap, and Hop ID, virtual coordinates are easy to update in networks with node failures, because virtual coordinates are assigned according to the hop distances to landmarks. GLIDER, GLDR, VCap, and Hop ID each, however, cannot guarantee packet delivery with the exception that the flooding mechanism is used, which requires a great deal of route overhead and suffers from the broadcasting storm problem [7].

In this paper, in a wireless sensor network without GPS assistance, we construct virtual faces and propose a VirtualFace algorithm to route a packet from a node with no neighbor closer to the destination (dead-end node) to a node closer to the destination (progress node), such that the virtual-coordinate-based routing protocols augmented with the VirtualFace algorithm can guarantee packet delivery. Face algorithms [8], [9], [10] that route a packet from a dead-end node to a progress node in geographic routing protocols by traversing the boundaries of faces in a planar graph does not work in virtual-coordinate-based routing protocols because virtual coordinates cannot be used to remove crossing links to build a planar graph. In FaceTrace [11], a face is delimited by a closed walk whose two consecutive edges are consecutive edges incident...
to a node in the counterclockwise direction. FaceTrace solves the dead-end node problem in geographic routing protocols without the construction of a planar graph; it, however, cannot be used in virtual-coordinate-based routing protocols because edges incident to a node cannot be labeled in the counterclockwise direction without GPS assistance. In this paper, we assume that each node has the information of the neighbors obtained by message exchange, where the information of a node includes the IDs of the node, the neighbors, and the neighbors in the connected dominating set, the hop count from a specific node, and the ID and the name of the virtual face whose boundary contains the node. Each node is also assumed to be static, have a unique identifier (ID), and have the same transmission range. The remainder of this paper is organized as follows. The virtual face construction protocol (VFCP) and the virtual face naming protocol (VFNP) are proposed to construct and name virtual faces in Sections II and III, respectively. In Section IV, the VirtualFace algorithm is presented. We analyze the VirtualFace algorithm in Section V, and evaluate, by simulations, the performance of the VirtualFace algorithm in Section VI. Finally, we conclude this paper in Section VII.

II. VIRTUAL FACE CONSTRUCTION PROTOCOL (VFCP)

VFCP constructs virtual faces with boundaries closely surrounding holes, such that a packet can efficiently turn around a hole by traversing the boundary of the surrounding virtual face. In VFCP, the boundaries of virtual faces are down-up cycles in the triangle-free subnetwork induced by the connected dominating set, which are constructed in a four-phase process. In the first phase, the triangle-free subnetwork induced by the connected dominating set, $G_D$, is generated. In the second and third phases, each node in $G_D$ evaluates the hop distance to a specific node, $S$, and each tail node computes the lengths of the shortest down-up cycles, respectively. Finally, virtual faces are constructed in $G_D$ in the fourth phase. In VFCP, virtual faces are constructed in $G_D$ to reduce the number of virtual faces. The following notations are necessary for the description of VFCP.

Definition 1. A network is triangle-free if any three nodes are not mutually neighbors.

Definition 2. A dominating set denotes a subset of nodes such that each node not in the set is a neighbor of at least one node in the set. A dominating set is connected if the induced subnetwork is connected.

Definition 3. In $G_D$, a cycle $(u_1, u_2, \cdots, u_n)$ is called a down-up cycle if a node, $u_i$, exists such that $u_i.hp(S) - 1 \leq u_{i+1}.hp(S) \leq u_i.hp(S)$ for $1 \leq i < h$ and $u_i.hp(S) \leq u_{i+1}.hp(S) \leq u_i.hp(S) + 1$ for $h \leq i < n$, where $u_1 = u_n$ and $u_i.hp(S)$ denotes the hop distance from $u_i$ to $S$ in $G_D$.

Definition 4. In $G_D$, a node, $u$, is called a tail node if at least two neighbors $v$ and $w$ exist such that $v.hp(S) = u.hp(S) - 1$ and $w.hp(S) \leq u.hp(S)$.

Example 1. In Fig. 1, each node depicted by a dashed circle is a neighbor of at least one node shown in a solid circle, and the subnetwork induced by nodes shown in solid circles is connected; therefore, the set of nodes shown in solid circles is a connected dominating set according to Definition 2. In cycle $(11, 7, 4, 3, 12, 25, 11)$, $u_1.hp(S) = 5$, $u_2.hp(S) = 4$, $u_3.hp(S) = 3$, $u_4.hp(S) = 2$, $u_5.hp(S) = 3$, $u_6.hp(S) = 4$, and $u_7.hp(S) = 5$ satisfying $u_i.hp(S) - 1 \leq u_{i+1}.hp(S) \leq u_i.hp(S)$ for $1 \leq i < 4$ and $u_i.hp(S) \leq u_{i+1}.hp(S) \leq u_i.hp(S) + 1$ for $4 \leq i < 7$; therefore, cycle $(11, 7, 4, 3, 12, 25, 11)$ is a down-up cycle according to Definition 3. According to Definition 4, node 11 is a tail node because nodes 7 and 25 are neighbors of node 11, $7.hp(S) = 11.hp(S) - 1$, and $25.hp(S) \leq 11.hp(S)$.

A. Generation of a Triangle-Free Subnetwork

There are two steps in this phase. The connected dominating set is first generated by the pruning algorithm [12]. Subsequently, the triangle-free subnetwork is constructed by removing each edge between two nodes with the largest two IDs in a triangle in the subnetwork induced by the connected dominating set.

B. Evaluation of Hop Distances

Let $S$ be the pre-programmed node or the node with the largest ID dominating the pre-programmed node. Each node, $u$, evaluates $u.hp(S)$, as implemented in the following. Firstly, $S$ broadcasts an $S\_SET$ message containing a hop counter initially set to 1. Once a node, $u$, receives an $S\_SET$ message, $u$ sets $u.hp(S)$ to the hop counter contained in the message and broadcasts the message containing the hop counter increased by 1. If $u$ receives more than one $S\_SET$ message, $u$ broadcasts the $S\_SET$ message containing the smallest hop counter and sets $u.hp(S)$ according to the message.

C. Computation of Shortest Down-Up Cycle Lengths

![Fig. 1. A wireless sensor network, $G$, and the triangle-free subnetwork induced by the connected dominating set, $G_D$. Nodes and links in $G_D$ are shown in solid circles and lines, respectively; nodes and links in $G - G_D$ are shown in dashed circles and lines, respectively. $S$ denotes the specific node. The number in the parenthesis denotes the hop distance to $S$ in $G_D$.](image)
A tail node, \( u \), evaluates the lengths of shortest down-up cycle, as implemented in the following. Each neighbor, \( v \), of \( u \) with \( u.hp(S) = u.hp(S) - 1 \) generates a DUCYCLE_SET message containing the ID of \( v \), a hop counter initially set to 1, and a state status initially set to down; subsequently, the message is forwarded to all neighbors \( y \) with \( y.hp(S) \leq u.hp(S) \). Once a node, \( x \), receives a DUCYCLE_SET message generated by \( v \), \( x \) sets \( x.hp(v) \) to the hop counter contained in the message, and forwards the message containing the ID of \( v \), the hop counter increased by 1, and a state status equal to up to all neighbors \( y \) with \( y.hp(S) \geq x.hp(S) \). In addition, if the state status contained in the received DUCYCLE_SET message equals down, \( x \) also forwards the message containing the ID of \( v \), the hop counter increased by 1, and a state status equal to down to all neighbors \( y \) with \( y.hp(S) \leq x.hp(S) \). If \( x \) receives more than one DUCYCLE_SET message generated by \( v \), \( x \) forwards the DUCYCLE_SET message containing the smallest hop counter, and sets \( x.hp(v) \) according to the message. After \( u \) receives the DUCYCLE_SET message generated by \( v \) from \( w \) with \( w.hp(S) \leq u.hp(S) \), the length of the shortest down-up cycle containing \( u \) and \( v \) is evaluated as \( 2 + w.hp(v) \).

**Example 2.** An example of construction of virtual faces using VFCP is illustrated in Fig. 2. In the first phase, the connected dominating set is first generated by the pruning algorithm. Subsequently, the edge between nodes 5 and 7 is removed because nodes 4, 5, and 7 are in a triangle in the subnetwork induced by the connected dominating set. In the second phase, node 14 is the pre-programmed node and broadcasts an S_SET message containing a hop counter equal to 1. After node 9 receives the S_SET message, node 9 sets \( 9.hp(S) \) to 1, and broadcasts the message containing a hop counter equal to 2. In the third phase, node 7 generates a DUCYCLE_SET message containing the ID of node 7, a hop counter equal to 1, and a state status equal to down because node 7 is a neighbor of a tail node 11, and \( 7.hp(S) = 11.hp(S) - 1 \). The message is forwarded to node 4 because \( 4.hp(S) \leq 7.hp(S) \). Node 4 sets \( 4.hp(7) \) to 1, and updates the hop counter contained in the message to 2. Subsequently, the message containing a state status equal to down is forwarded to node 3 because \( 3.hp(S) \leq 4.hp(S) \), and the message containing a state status equal to up is forwarded to node 5 because \( 5.hp(S) \geq 4.hp(S) \). Repeating the forwarding process, node 25 extends two DUCYCLE_SET messages generated by node 7. One travels nodes 7, 4, 3, 12, and 25, and the other travels nodes 7, 4, 3, 10, 14, 9, 22, 1, and 25. Node 25 forwards the former message and sets \( 25.hp(7) \) to 4. After node 11 receives the DUCYCLE_SET message from node 25, the length of the shortest down-up cycle containing nodes 11, 7, and 25 is evaluated as \( 2 + 25.hp(7) = 6 \). In the fourth phase, to construct a virtual face, \( vf_7 \), node 11 generates a FACE_SET message containing the ID of node 7, \( vf_7.id \), and \( vf_7.size = 2 + 25.hp(7) = 6 \). The FACE_SET message backtracks the cycle \((11, 7, 4, 3, 12, 25, 11)\), in which process nodes 11, 25, 12, 34, and 7 set \( 11.seq(vf_7), 25.seq(vf_7), 12.seq(vf_7), 3.seq(vf_7), 4.seq(vf_7), \) and \( 7.seq(vf_7) \) to \( vf_7.size = 6 \), \( 25.hp(7) + 1 = 5 \), \( 12.hp(7) + 1 = 4 \), \( 3.hp(7) + 1 = 3 \), \( 4.hp(7) + 1 = 2 \), and \( 7.hp(7) + 1 = 1 \), respectively. In addition, a virtual face, \( vf_0 \), with a boundary containing only node 15 is constructed because node 15 is not contained in the boundaries of virtual faces \( vf_1, vf_2, vf_3, vf_4, vf_5, \) and \( vf_7 \).

**III. VIRTUAL FACE NAMING PROTOCOL (VFNP)**

VFNP names virtual faces such that a packet can traverse the boundaries of virtual faces efficiently. A virtual face, \( vf \), is named by the polar coordinate system [13] as \( (vf.radial, vf.angular) \), where \( vf.radial \) and \( vf.angular \) denoting the radial and angular coordinates are an integer and an interval of real numbers \([a, b]\), respectively. The virtual face with the smallest virtual face ID is called the pole virtual face and denoted as \( vf_p \). The radial and angular coordinates of the pole virtual face equal 0 and \([0, 2\pi)\), respectively. The following notations are necessary for the description of VFNP.

**Definition 5.** The virtual face graph, \( G_{vf} = (V_{vf}, E_{vf}) \), is an undirected graph, in which node \( u_{vf} \) is in \( V_{vf} \) if a virtual face \( vf \) is constructed in \( G_{p} \), and edge \((u_{vf_1}, u_{vf_2})\) is in \( E_{vf} \) if \( vf_1 \) and \( vf_2 \) are neighboring virtual faces, in which \( vf_1 \) and \( vf_2 \) are...
neighboring virtual faces if a node in $G_D$ is contained in the boundaries of $vf_1$ and $vf_2$ or if two neighboring nodes in $G_D$ are contained in the boundaries of $vf_1$ and $vf_2$, respectively.

**Definition 6.** Let $u_{vf}$ be a node in $G_{vf}$, and let $u$ be a node contained in the boundary of virtual face $vf$. In $G_{vf}$, $u_{vf}$ is a $u$-neighbor of $u_{vf}$ if $vf$ is a $u$-neighbor of $u_{vf}$ if the boundary of virtual face $vf_1 (\neq vf)$ contains $u$ or a neighbor of $u$ in $G_D$.

**Definition 7.** Interval $[a, b)$ is smaller than interval $[c, d)$, denoted by $(a, b) < (c, d)$, if $b \leq c$, or is equal to interval $[c, d)$, denoted by $(a, b) = (c, d)$, if $a = c$ and $b = d$, and is between interval $[c, d)$, denoted by $(a, b) \subset [c, d)$, if $a \geq c$, $b \leq d$, and $(a, b)$ is not equal to $[c, d)$. In addition, $(a, b) \subset [c, d)$ if $(a, b) \subset [c, d)$ and $(a, b) \subset [c, d)$.

**Definition 8.** The pair of numbers $a$ and $b$ is smaller than the pair of numbers $c$ and $d$, denoted by $(a, b) < (c, d)$, if $a < c$, or $a = c$ and $b < d$.

**Example 3.** Fig. 3 illustrates the virtual face graph, $G_{vf} = (V_{vf}, E_{vf})$, for virtual faces in Fig. 2. According to Definition 5, $V_{vf} = \{u_{vf_1}, u_{vf_2}, u_{vf_3}, u_{vf_4}, u_{vf_5}, u_{vf_6}, u_{vf_7}\}$. $vf_1$ and $vf_7$ are neighboring virtual faces because node 25 is contained in the boundaries of $vf_1$ and $vf_7$; therefore, $(u_{vf_2}, u_{vf_3}) \in E_{vf}$. Additionally, $vf_1$ and $vf_7$ are neighboring virtual faces because nodes 3 and 15, contained in the boundaries of $vf_1$ and $vf_6$, respectively, are neighbors in $G_D$; therefore, $(u_{vf_1}, u_{vf_6}) \in E_{vf}$. According to Definition 6, $u_{vf_2}$ is a 3-neighbor of $u_{vf_1}$ in $G_D$ because the boundaries of $vf_1$ and $vf_2$ contain node 3 whereas $u_{vf_3}$ is a 4-neighbor of $u_{vf_2}$ in $G_D$ because node 7 contained in the boundary of $vf_3$ is a neighbor of node 4 contained in the boundary of $vf_2$.

Let the ID, the angular coordinate, and the radial coordinate of a virtual face, $vf$, in $G_D$ equal the ID, the angular coordinate, and the radial coordinate of $u_{vf}$ in $G_{vf}$, respectively. The radial coordinate of $u_{vf}$ equals the hop distance to the corresponding node of $vf_{p}$ in $G_{ef}$ ($u_{vf_{p}}$). Let $u$ be a node contained in the boundary of $vf$, and let $u_{vf_1}, u_{vf_2}, \ldots , u_{vf_n}$ be the $u$-neighbors of $u_{vf}$ in $G_{vf}$ in an increasing order of IDs. If the angular coordinate of $u_{vf}$ equals interval $[a, b)$, the size of $vf$ equals $c$, and the sequence number of $u$ in $vf$ equals $i$, the angular coordinates of $u_{vf_1}, u_{vf_2}, \ldots , u_{vf_n}$ each are between interval $[a + (i -1)(b-a)/c, a + i\cdot(b-a)/c)$, the $i$-th smallest equal sub-interval of $[a, b)$. More specifically, the angular coordinate of $u_{vf_2}$ equals interval $[a + (i-1)(b-a)/c + (j-1)(b-a)/(c \cdot n), a + (i-1)(b-a)/c + j\cdot(b-a)/(c \cdot n)]$, the $j$-th smallest equal sub-interval of $[a + (i -1)(b-a)/c, a + i\cdot(b-a)/c)$. In addition, if $u_{vf}$ has more than one neighbor with a radial coordinate smaller by 1 in $G_{vf}$, the angular coordinate is assigned according to the neighbor with the smallest angular coordinate ($u_{vf_{min}}$). If the boundary of $vf$ contains more than one node contained in or having a neighbor contained in the boundary of $vf_{min}$, the angular coordinate of $u_{vf}$ is assigned according to the node with the smallest sequence number in $vf_{min}$.

VFNP is implemented as follows. Let $u$ be a node in the boundary of virtual face $vf$. If $u_{vf}$ has a smaller ID than any $u$-neighbor in $G_{vf}$, $u$ assigns $vf_{p}.id, vf.radial$, and $vf.angular$ to $vf_{id}, 0$, and $[0, 2\pi)$, respectively, and then generates and broadcasts to neighbors a NAME_SET message containing $vf_{p}.id, vf.size, vf.id, vf.radial, vf.angular, u.seq(vf)$, and the IDs of $u$-neighbors of $u_{vf}$. If $u$ receives a NAME_SET message containing $vf.id$ and $vf.size$, $u$ broadcasts the message to the neighbors in the boundary of $vf$, and assigns $vf_{p}.id, vf.radial$ and $vf.angular$ to $vf_{p}.id, vf.radial$ and $vf.angular$ contained in the message, respectively. If $u$ receives from a node, $v$, a NAME_SET message containing $vf_{1}.id$ and $vf_{1}.size$ ($vf_{1} \neq vf$), $u$ broadcasts the message to the neighbors in the boundary of $vf$, assigns $vf_{2}.id$ to $vf_{p}.id$ contained in the message, assigns $vf.radial$ to $vf_{2}.radial + 1$, and assigns $vf.angular$ to $[a + (i -1)(b-a)/c + (j -1)(b-a)/(c \cdot n), a + (i -1)(b-a)/c + j\cdot(b-a)/(c \cdot n)]$, in which interval $[a, b)$ is the angular coordinate of $vf_{1}$, $i$ equals $v.seq(vf_{1})$, $c$ equals $vf_{1}.size$, $n$ is the number of $v$-neighbors of $u_{vf_{1}}$ in $G_{vf}$, and $j$ is the number of $u$-neighbors of $u_{vf}$ with a smaller or an equal ID than $u_{vf}$ in $G_{vf}$. In addition, if at least one $u$-neighbor of $u_{vf}$ exists, $u$ also generates and broadcasts to neighbors a NAME_SET message containing $vf_{p}.id, vf.size, vf.id, vf.radial, vf.angular, u.seq(vf_{p})$, and the IDs of $u$-neighbors of $u_{vf}$ after $v$ receives a NAME_SET message. If $u$ receives more than one NAME_SET message, $u$ assigns $(vf.radial, vf.angular)$ to the smallest pair of numbers $(vf.radial, vf.angular)$ according to the message containing the smallest $vf_{p}.id$.

**Example 4.** An example of naming virtual faces using VFNP is illustrated in Fig. 3. Assume that $vf_{1}.id < vf_{2}.id$ if $i < j$. We have $vf_{p} = vf_{1}, vf_{1}.radial = 0$, and $vf_{1}.angular = [0, 2\pi)$. Because node 3 is contained in the boundary of $vf_{1}$ and $u_{vf_{2}}$ is a 3-neighbor of $u_{vf_{1}}$, node 3 generates a NAME_SET message containing $vf_{p}.id = vf_{2}.id, vf_{2}.size = 8, vf_{1}.id, vf_{1}.radial = 0, vf_{1}.angular = [0, 2\pi), 3.seq(vf_{1}) = 2$, and the IDs of $u_{vf_{2}}, u_{vf_{6}}$, and $u_{vf_{7}}$. After node 4 receives the NAME_SET message generated by node 3, node 4 broadcasts the message, sets $vf_{2}.radial$ to 1, and sets $vf_{2}.angular$ to $[(i-1)\cdot 2\pi/8 + (j -1)\cdot 2\pi]$. The figure shows the virtual face graph, $G_{vf}$, and the names of nodes using VFNP. The first and second entries in the parenthesis denote the radial and angular coordinates of the node, respectively.
2π/24, (i − 1) · 2π/8 + j · 2π/24 = [π/4, π/3] because
vf1.angular = [0, 2π], vf1.size = 8, u0f2 has the smallest
ID among u0f2, u0fa, and u0fb, and 3.seq(vf1) = 2. Mean-
while, node 4 sets vf2.radial to 1, and sets vf2.angular to
[5π/12, π/2] because u0f2 has the largest ID among u0f2,
ufa, and u0fb. Similarly, node 12 generates a NAME_SET
message containing vfp.id = vf1.id, vfp.size = 8, vf1.id, vf1.angular
= 0, vf1.radial = 4, 12.seq(vf1) = 1, and the IDs of u0f2,
u0fa, and u0fb. After node 12 receives the NAME_SET
message generated by node 12, node 5 broadcasts the
message, sets vf2.radial to 1, and sets vf2.angular
to [(i − 1) · 2π/8 + (j − 1) · 2π/16, (i − 1) · 2π/8 + j · 2π/16] =
[0, π/8] because vf1.angular = [0, 2π], vf1.size = 8, u0f2
has a smaller ID than u0f2, and 12.seq(vf1) = 1. After node
12 receives the NAME_SET message generated by node 12,
node 12 broadcasts the message and updates vfp2.angular
to [0, π/8], because (1, [0, π/8]) < (1, [π/4, π/3]) according
to Definitions 7 and 8. Meanwhile, node 4 generates a NAME_SET
message containing vfp.id = vf1.id, vfp.size = 4, vf2.id, vf2.angular
= 1, vf2.radial = 8, 4.seq(vf2) = 1, and the IDs of u0f1,
u0fb, and u0f2. After node 7 receives the NAME_SET
message generated by node 4, node 7 sets vfp.radial
2, and sets vfp.angular to [(i − 1) · 2π/8/4 +
(j − 1) · π/8/12, (i − 1) · π/8/4 + j · π/8/12] = [π/96, π/48].

IV. THE VIRTUALFACE ALGORITHM

We assume that the routed packet carries the name of the
virtual face, vf, with a boundary containing the destination
or the dominating node of the destination. Once a message
counters a dead-end node, the VirtualFace algorithm routes
the message toward vf until a progress node is reached.
The following notations are necessary for the description of the
VirtualFace algorithm.

Definition 9. Let [a, b) and [c, d) be the angular coordinates
of virtual faces vf1 and vf2, respectively. The angular distance
from vf1 to vf2, denoted by distang(vf1, vf2), equals c−b+1
if [a, b) < [c, d), equals a − d + 1 if [c, d) < [a, b), and equals
0 otherwise. The radial distance from vf1 to vf2, denoted by
distrad(vf1, vf2), equals |vf1.radial − vf2.radial| if
distang(vf1, vf2) = 0, and equals vf1.radial otherwise.

Definition 10. The distance between virtual faces vf1
and vf2, denoted by dist(vf1, vf2), is a pair of numbers
(distang(vf1, vf2), distrad(vf1, vf2)). vf1 is said to be closer
to virtual face vf3 than vf2 if dist(vf1, vf3) < dist(vf2, vf3).

Example 5. Let the names of virtual faces vf2, vf3, vf4,
and vf7 be (1, [0, π/8)), (1, [3π/4, π]), (2, [π/96, π/48]),
and (1, [π/8, π/4]), respectively. According to Definition 9,
distang(vf2, vf3) = 5π/8 + 1, distang(vf1, vf3) = 35π/48 +
1, and distang(vf2, vf7) = π/2 + 1; distrad(vf2, vf3) = 1,
distrad(vf3, vf3) = 2, and distrad(vf7, vf3) = 1. Accord-
ing to Definition 10, dist(vf1, vf3) < dist(vf2, vf3) and
dist(vf1, vf7) < dist(vf2, vf3).

If a packet with the destination, d, encounters a dead-end
node, the VirtualFace algorithm routes the packet to a progress
node, as described in the following. If the forwarding node,
u, is not in G_D, u forwards the packet to the dominating
node of u. Otherwise, if one neighbor, v, of u in G is a
progress node, the packet is forwarded to v. Otherwise, if
u is contained in the boundary of vf, the packet is forwarded
to the successive node of u in the boundary of vf, where
vf denotes the virtual face closest to vf among all virtual faces
with boundaries containing u or at least one neighbor of u.
Otherwise, the packet is forwarded to one neighbor of u contained in the boundary of vf.

Example 6. An example of VCap augmented with the Vir-
tualFace algorithm is illustrated in Fig. 4, where a packet
is routed from node 8 to node 20. In VCap, a node forwards
a packet to a neighbor having a closer and minimal distance
to the destination, where the distance between two nodes with
virtual coordinates (x1, y1, z1) and (x2, y2, z2) is defined as
\(\sqrt{(x2 - x1)^2 + (y2 - y1)^2 + (z2 - z1)^2}\). Using VCap, node
8 forwards the packet to node 21 because node 21 is a
neighbor having a closer and minimal distance to node 20.
Subsequently, the packet is forwarded to node 2, and then
forwarded to node 6. Node 6 is a dead-end node because
none of its neighbors is closer to node 20. Therefore, the
VirtualFace algorithm is used to route the packet, where
vf = vf because the dominating node of node 20, node
34, is contained in the boundary of vf. Node 6 forwards
the packet to its dominating node, node 7. The boundaries
of vf, vf, and vf, each contain node 7 or a neighbor
of node 7. Because dist(vf1, vf3) < dist(vf2, vf3) and
dist(vf7, vf3) < dist(vf7, vf3), vf closest = vf7. The packet
is forwarded to node 11, the successive node of node 7 in the
boundary of vf7. As the packet is at node 11, vf closest = vf7.
The packet is forwarded to node 25, the neighbor of node 11
in the boundary of vf1. Node 26, the neighbor of node 25,
is a progress node because node 26 is closer to node 20 than node 6 (the dead-end node). The packet is forwarded to node 26. Using VCap, the packet is eventually routed to node 20.

V. ANALYSIS OF THE VIRTUALFACE ALGORITHM

We show the VirtualFace algorithm can always route a packet from a dead-end node to a progress node.

**Lemma 1.** For any virtual face \( v_f_1 \neq v_f_2 \), there exists a virtual face, \( v_f_2 \), such that \( v_f_2.radial = v_f_2.radial - 1 \) and \( v_f_1.angular \subseteq v_f_2.angular \).

**Proof:** Assume that \( v_f_1.radial \) and \( v_f_1.angular \) are assigned when a node, \( u \), contained in the boundary of \( v_f_1 \) receives a NAME_SET message from a node, \( v \), contained in the boundary of \( v_f_2 \). By VFNP, \( v_f_1.radial \) equals \( v_f_2.radial + 1 \), and \( v_f_1.angular \) equals the \( j \)-th smallest equal sub-interval of the \( i \)-th smallest equal sub-interval of \( v_f_2.angular \) if the sequence number of \( v \) in \( v_f_2 \) is \( i \) and the number of \( v \)-neighbors of \( u.v_f_2 \) with a smaller or an equal ID than \( u.v_f_1 \) in \( G_v \) is \( j \).

**Lemma 2.** Let \( v_f_1 \) and \( v_f_2 \) be two virtual faces. If \( v_f_1.radial = v_f_2.radial \), then \( v_f_1.angular < v_f_2.angular \).

**Proof:** It suffices to show the statement \( S \), there exists no virtual faces, \( v_f_1 \) and \( v_f_2 \), with radial coordinate \( k \) such that neither of \( v_f_1.angular < v_f_2.angular \) and \( v_f_2.angular < v_f_1.angular \) is satisfied, holds for all \( k \geq 0 \). \( S \) holds for \( k = 0 \) because there is only one virtual face with radial coordinate 0; therefore, a basis for the proof exists. We prove \( S \) holds for \( k \geq 1 \) by induction on \( k \). As an induction assumption, we take \( S \) holds for \( k \leq m - 1 \). If \( v_f_1.radial = v_f_2.radial = m \), according to Lemma 1, there exists virtual faces, \( v_f_3 \) and \( v_f_4 \), with \( v_f_3.radial = v_f_4.radial = m - 1 \) such that \( v_f_1.angular \subseteq v_f_3.angular \) and \( v_f_2.angular \subseteq v_f_4.angular \). If \( v_f_3 \neq v_f_4 \), then \( v_f_4.angular < v_f_3.angular \) by induction hypothesis, implying \( v_f_1.angular < v_f_2.angular \) or \( v_f_2.angular < v_f_1.angular \). If \( v_f_3 = v_f_4 \), \( v_f_3.angular \) (or \( v_f_2.angular \)) equals the \( j_1 \)-th (or \( j_2 \)-th) smallest equal sub-interval of the \( i_1 \)-th (or \( i_2 \)-th) smallest equal sub-interval of \( v_f_3.angular \), where \( i_1 \) (or \( i_2 \)) denotes the sequence number of the node, \( v_1 \) (or \( v_2 \)), contained in the boundary of \( v_f_3 \) from which the node contained in the boundary of \( v_f_1 \) (or \( v_f_2 \)) receives the NAME_SET message to assign the radial and angular coordinates of \( v_f_1 \) (or \( v_f_2 \)), and \( j_1 \) (or \( j_2 \)) denotes the number of \( v_1 \)-neighbors (or \( v_2 \)-neighbors) of \( u.v_f_3 \) with a smaller or an equal ID than \( u.v_f_1 \) (or \( u.v_f_2 \)) in \( G_v \). This implies that \( v_f_1.angular < v_f_2.angular \), \( v_f_2.angular < v_f_1.angular \), or \( v_f_1.angular = v_f_2.angular \). The last case holds only if \( v_1 = v_2 \) and the ID of \( u.v_f_1 \) equals to that of \( u.v_f_2 \), which is impossible because \( v_f_1.id \neq v_f_2.id \).

**Theorem 1.** For any two virtual faces \( v_f_1 \) and \( v_f_2 \), one of the following conditions is satisfied.

1. \( v_f_1.angular < v_f_2.angular \),
2. \( v_f_2.angular < v_f_1.angular \),
3. \( v_f_1.angular < v_f_2.angular \) or \( v_f_2.angular < v_f_1.angular \),
4. \( v_f_1.angular \subseteq v_f_2.angular \),
5. \( v_f_1.angular = v_f_2.angular \).

**Proof:** Three cases need to be discussed: (c1) \( v_f_1.angular = v_f_2.angular \); (c2) \( v_f_1.angular < v_f_2.angular \); (c3) \( v_f_1.angular > v_f_2.angular \). The proof of c3 is omitted due to its similarity with that of c2. For c1, \( v_f_1.angular < v_f_2.angular \) or \( v_f_2.angular < v_f_1.angular \) according to Lemma 1. If \( v_f_3 \neq v_f_1 \), \( v_f_1.angular < v_f_2.angular \) or \( v_f_2.angular < v_f_1.angular \) according to Lemma 2. If \( v_f_3 = v_f_1 \), \( v_f_2.angular \subseteq v_f_1.angular \).

**Lemma 3.** Let \( v_f_1 \) and \( v_f_2 \) be two virtual faces. If \( v_f_2.angular \subseteq v_f_1.angular \), then \( v_f_1.radial > v_f_2.radial \).

**Proof:** According to Lemma 2, \( v_f_1.radial \neq v_f_2.radial \). Assume that \( v_f_1.radial > v_f_2.radial \). According to Lemma 1, there exists a virtual face, \( v_f_3 \), such that \( v_f_3.radial = v_f_2.radial \) and \( v_f_1.angular \subseteq v_f_3.angular \). It implies \( v_f_2.angular \subseteq v_f_3.angular \). According to Lemma 2, \( v_f_2.radial \neq v_f_1.radial \), constituting a contradiction.

**Lemma 4.** Let \( v_f_1 \) and \( v_f_2 \) be two virtual faces with \( v_f_1.radial < v_f_2.radial \) and \( v_f_2.angular \subseteq v_f_1.angular \). Then, there exists a virtual face, \( v_f_3 \), such that \( v_f_3.radial = v_f_2.radial - 1 \) and \( v_f_3.angular \subseteq v_f_1.angular \), and there exists a virtual face, \( v_f_4 \), such that \( v_f_4.radial = v_f_2.radial - 1 \) and \( v_f_2.angular \subseteq v_f_4.angular \). It suffices to show \( v_f_3 = v_f_4 \) to complete the proof. Assume that \( v_f_1 \neq v_f_3 \). Because \( v_f_1.radial = v_f_3.radial \), \( v_f_1.angular < v_f_3.angular \), or \( v_f_3.angular < v_f_1.angular \) according to Lemma 2, resulting in the impossibility that \( v_f_2.angular \subseteq v_f_1.angular \) and \( v_f_2.angular \subseteq v_f_3.angular \). In addition, \( v_f_1 = v_f_3 \) can be proved in a manner analogous for that of \( v_f_1 = v_f_5 \).

**Theorem 2.** If \( G \) is connected, then for any virtual face \( v_f_1 \neq v_f_d \), there exists a neighboring virtual face \( v_f_2 \), such that \( dist(v_f_2, v_f_d) < dist(v_f_1, v_f_d) \).

**Proof:** According to Theorem 1, we have to discuss five cases: (c1) \( v_f_1.angular < v_f_d.angular \); (c2) \( v_f_d.angular < v_f_1.angular \); (c3) \( v_f_1.angular < v_f_d.angular \); (c4) \( v_f_d.angular < v_f_1.angular \); (c5)
\( v_{f1, \text{angular}} = v_{f2, \text{angular}} \). We adhere to the proofs of c1 and c3 and omit the proofs of the other cases due to the similarities. For c1, because \( v_{f1} = v_{f2} \) is impossible to hold, there exists a virtual face, \( v_{f2} \), such that \( v_{f2, \text{radial}} = v_{f1, \text{radial}} - 1 \), and c1.1) \( v_{f1, \text{angular}} \subset v_{f2, \text{angular}} \) or c1.2) \( v_{f1, \text{angular}} = v_{f2, \text{angular}} \) according to Lemma 1. For c1.1, neither of \( v_{f2, \text{angular}} < v_{f2, \text{angular}} \) and \( v_{f2, \text{angular}} \subset v_{f2, \text{angular}} \) holds because \( v_{f1, \text{angular}} < v_{f2, \text{angular}} \). Therefore, two cases need to be discussed: c1.1.1) \( v_{f2, \text{angular}} < v_{f2, \text{angular}} \); c1.1.2) \( v_{f2, \text{angular}} \subset v_{f2, \text{angular}} \). For c1.1.1, according to Definition 9, \( 0 < \text{dist}_{\text{ang}}(v_{f2}, v_{f3}) \leq \text{dist}_{\text{ang}}(v_{f1}, v_{f3}) \), and \( \text{dist}_{\text{rad}}(v_{f2}, v_{f3}) < \text{dist}_{\text{rad}}(v_{f1}, v_{f3}) \) because \( v_{f2, \text{radial}} < v_{f1, \text{radial}} \). This implies \( \text{dist}(v_{f2}, v_{f3}) < \text{dist}(v_{f1}, v_{f3}) \) according to Definition 10. For c1.1.2, \( \text{dist}_{\text{ang}}(v_{f2}, v_{f3}) = 0 < \text{dist}_{\text{ang}}(v_{f1}, v_{f3}) \), implying \( \text{dist}(v_{f2}, v_{f3}) < \text{dist}(v_{f1}, v_{f3}) \). c1.2 can be proved in a manner analogous for that of c1.1.1, completing the proof of c1. For c3, \( v_{f1, \text{radial}} > v_{f2, \text{radial}} \) according to Lemma 3. Therefore, there exists a virtual face, \( v_{f2} \), such that \( v_{f2, \text{radial}} = v_{f1, \text{radial}} - 1 \) and \( v_{f1, \text{angular}} \subset v_{f2, \text{angular}} \) according to Lemma 4. Clearly, \( \text{dist}_{\text{ang}}(v_{f2}, v_{f3}) = 0 = \text{dist}_{\text{ang}}(v_{f1}, v_{f3}) \). In addition, \( v_{f1, \text{radial}} > v_{f2, \text{radial}} \geq v_{f3, \text{radial}} \). Therefore, \( \text{dist}_{\text{rad}}(v_{f2}, v_{f3}) < \text{dist}_{\text{rad}}(v_{f1}, v_{f3}) \), and thus \( \text{dist}(v_{f2}, v_{f3}) < \text{dist}(v_{f1}, v_{f3}) \), completing the proof of c3.

**Theorem 3.** If \( G \) is connected, the VirtualFace algorithm can always route a packet to a progress node.

**Proof:** Let \( u \) be a forwarding node in the boundary of a virtual face, \( v_{f} \). Using the VirtualFace algorithm, \( u \) forwards a packet to the successor node in the boundary of \( v_{f} \) if no neighbor is in the boundary of a virtual face closer to \( v_{f} \); otherwise, \( u \) forwards a packet to a neighbor in the boundary of the virtual face closest to \( v_{f} \). It implies the packet traverses the boundaries of virtual faces in a decreasing order of the distance to \( v_{f} \). Because any virtual face except for \( v_{f} \) has a neighboring virtual face closer to \( v_{f} \) according to Theorem 2, the packet can be always routed to \( v_{f} \), and thus to the destination if no progress node is encountered during the routing process. Because the destination is a progress node, the proof is completed.

**VI. PERFORMANCE EVALUATION**

In our simulations, 100 connected networks with densities ranging from 10 to 30 were generated by randomly deploying nodes in square regions with a side length equal to 25, in which the network density denoted the average number of neighbors per node, and the transmission range of a node was a circle of radius 1. Network behavior such as packet loss, packet delay, and so on were not taken into consideration. GLIDER, Hop ID, GLDR, and VCap using the VirtualFace algorithm to route a packet from a dead-end node to a progress node were denoted by GLIDER+VF, Hop ID+VF, GLDR+VF, and VCap+VF, respectively, and compared with GLIDER, Hop ID, GLDR, VCap, and ABVCap, in terms of the packet delivery rate, the routing path length, the number of next hop neighbors, the load imbalance factor, the number of broadcasts per node, and the packet delivery rate in networks with node failures, in which the load imbalance factor denoted the ratio of the maximum number of packets routed by a node (the maximum load) to the average number of packets routed by a node (the average load). The existence of a node routing many packets was indicated by a large load imbalance factor.

In GLIDER and GLIDER+VF, 23 landmarks were randomly selected. In Hop ID and Hop ID+VF, the peripheral landmark selection of 30 landmarks was implemented. In GLDR and GLDR+VF, 10-sampling was used to select landmarks. The flooding mechanism in GLIDER, Hop ID, and GLDR was not implemented because it required a great deal of route overhead and suffered from the broadcasting storm problem [7]. Empirical data were obtained by averaging data of 1000 source-destination pairs from 100 networks.

**A. Packet Delivery Rate**

Fig. 5(a) illustrates the simulation results for the packet delivery rate. GLIDER+VF, Hop ID+VF, GLDR+VF, VCap+VF and ABVCap each successfully set a path for every source-destination pair. In GLIDER, Hop ID, GLDR, and VCap, the greater the network density, the higher the packet delivery rate because more dead-end nodes exist in a network with lower density due to the occurrence of more holes. GLIDER and VCap each have a lower packet delivery rate than Hop ID and GLDR, which results from the introduction of more dead-end nodes because large cells exist in GLIDER and multiple nodes share a virtual coordinate in VCap.

**B. Routing Path Length**

Fig. 5(b) illustrates the simulation results for the routing path length. GLIDER+VF, Hop ID+VF, GLDR+VF, and VCap+VF have longer routing paths than GLIDER, Hop ID, GLDR, and VCap, respectively, because many source-destination pairs separated by long distances are unreachable in GLIDER, Hop ID, GLDR, and VCap. The differences between Hop ID and Hop ID+VF and between GLDR and GLDR+VF are negligible because Hop ID or GLDR has a high packet delivery rate. Hop ID+VF, GLDR+VF, and VCap+VF each have a shorter routing path in most cases, as compared to ABVCap. By contrast, GLIDER+VF has a longer routing path than ABVCap. This observation results from the fact that GLIDER has a long routing path and a low packet delivery rate. Additionally, the greater is the network density the shorter is the routing path because the progress distance is larger.

**C. Number of Next Hop Neighbors**

Fig. 5(c) illustrates the simulation results for the number of next hop neighbors. In Hop ID and VCap, a node only forwards a packet to the neighbor closest to the destination; therefore, Hop ID and VCap have smaller numbers of next hop neighbors than GLIDER, GLDR, and ABVCap. In Hop ID, most forwarding nodes have only one next hop neighbor because a node is addressed by the hop distances to 30.
landmarks. Compared to Hop ID, a forwarding node in VCap has more next hop neighbors because more neighbors, which are addressed by the hop distances to 3 landmarks, share a virtual coordinate. In GLIDER, GLDR, VCap, and ABVCap, the greater the network density, the larger will be the number of next hop neighbors. In addition, because Hop ID and GLDR each have a high packet delivery rate, the differences between Hop ID and Hop ID+VF and between GLDR and GLDR+VF are minor. GLIDER+VF and VCap+VF have smaller numbers of next hop neighbors than GLIDER and VCap, respectively, because a forwarding node in the VirtualFace algorithm usually has only one next hop neighbor.

D. Load Imbalance Factor

Fig. 5(d) illustrates the simulation results for the load imbalance factor. GLIDER has the largest load imbalance factor because many packets are routed to the same node in the boundary of a cell. By contrast, unlike GLIDER and GLDR that route packets toward the landmarks and unlike ABVCap which routes packets toward the axes, VCap and Hop ID each route packets toward the destinations, and have a small load imbalance factor. In addition, the greater the network density, the larger will be the load imbalance factor because of the smaller average load. The differences between
Hop ID and Hop ID+VF and between GLDR and GLDR+VF are negligible, because Hop ID and GLDR each have a high packet delivery rate. Compared to GLIDER and VCap, GLIDER+VF and VCap+VF have smaller load imbalance factors, respectively, because of larger average loads.

E. Number of Broadcasts

Fig. 5(e) illustrates the simulation results for the number of messages broadcast by a node in the construction of virtual coordinate systems, in which messages are broadcast using the technique on trading time [5], and messages each travel at approximately the same speed. In Hop ID, GLDR and VCap, each node must obtain the hop distances to all landmarks; Hop ID and VCap have the largest and the smallest numbers of broadcasts due to the largest and the smallest numbers of landmarks, respectively. In GLIDER, each node must obtain the hop distances to all neighboring landmarks. As the network density increases, either the number of landmarks or the number of neighboring landmarks changes slightly, resulting in a negligible difference in the number of broadcasts in GLIDER, Hop ID, GLDR and VCap. By contrast, the greater the network density, the smaller the number of broadcasts in ABVCap because a node is assigned a smaller number of virtual coordinates in a greater density network. The number of broadcasts in GLIDER+VF, Hop ID+VF, GLDR+VF, and VCap+VF are larger than in GLIDER, Hop ID, GLDR, and VCap, respectively, because a node must broadcast 11.59 to 15.4 messages to construct and name virtual faces in networks with densities ranging from 10 to 30.

F. Packet Delivery Rate in Networks with Node Failures

Fig. 5(f) illustrates the simulation results for the packet delivery rate in networks with node failure equal to 10%. The packet delivery rate of each routing protocol degrades due to node failures. In Hop ID and VCap, a node closer to the destination can be a next hop neighbor candidate, and a next hop neighbor candidate can be a next hop neighbor only if it is closest to the destination; therefore, Hop ID and VCap each have few next hop neighbors, but have multiple next hop neighbor candidates. Consequently, node failure has a small impact on the packet delivery rates of Hop ID and VCap. By contrast, GLIDER, GLDR and ABVCap each have few next hop neighbor candidates, and thus have significantly degraded packet delivery rates in networks with node failures. In addition, GLIDER+VF, Hop ID+VF, GLDR+VF, and VCap+VF cannot guarantee packet delivery because the boundaries of some virtual face are disconnected due to node failures.

VII. CONCLUSION

Without GPS assistance, the virtual face construction protocol, VFCP, and the virtual face naming protocol, VFNP, are proposed in this paper to construct and name virtual faces, respectively, in a wireless sensor network. VFCP constructs virtual faces with boundaries closely surrounding holes. VFNP names virtual faces by the polar coordinate system, where the radial and angular coordinates of a virtual face denote the position of the virtual face in the network. A VirtualFace algorithm is proposed to route a packet from a dead-end node to a progress node by traversing the boundaries of virtual faces. As a result, virtual-coordinate-based routing protocols augmented with the VirtualFace algorithm guarantee packet delivery, in which the routed packet is only required to carry the radial and angular coordinates of the virtual face with a boundary containing the destination or the dominating node of the destination additionally.

Simulations show that virtual-coordinate-based routing protocols GLIDER, Hop ID, GLDR, and VCap augmented with the VirtualFace algorithm, denoted by GLIDER+VF, Hop ID+VF, GLDR+VF, and VCap+VF, respectively, have higher packet delivery rates, longer routing paths, larger coordinate assignment costs, and larger routing overheads, as compared to GLIDER, Hop ID, GLDR, and VCap, respectively. In terms of the number of next hop neighbors and the load imbalance factor, the differences between GLIDER and GLIDER+VF, between Hop ID and Hop ID+VF, between GLDR and GLDR+VF, and between VCap and VCap+VF are negligible. As compared to ABVCap, Hop ID+VF, GLDR+VF, and VCap+VF each have a shorter routing path and a lower load imbalance factor, GLDR+VF has a larger number of next hop neighbors, and GLIDER+VF, Hop ID+VF, GLDR+VF, and VCap+VF each have a higher packet delivery rate in networks with node failures.

REFERENCES