ProgressFace: An Algorithm to Improve Routing Efficiency of GPSR-like Routing Protocols in Wireless Ad Hoc Networks

Chia-Hung Lin, Shiao-An Yuan, Shih-Wei Chiu, and Ming-Jer Tsai
Department of Computer Science
National Tsing Hua University
Hsing Chu, Taiwan, ROC
E-mail: mjtsai@cs.nthu.edu.tw

Abstract—In GPSR-like routing, such as GPSR, GFG, GOAFR+, and GPVFR, perimeter forwarding is used to recover from a greedy forwarding failure by routing the packet to a progress node along the face boundary. The problem of perimeter forwarding is that many hops may be taken if the packet is forwarded in the wrong direction. We propose an algorithm, termed ProgressFace, that uses an additional traversal step to decide the direction of perimeter forwarding. A concave node sends a short packet to traverse the face boundary to identify the progress set, which consists of at most four nodes, such that, for any destination, at least one progress node is in the progress set or the neighbor set. Additionally, the hop distances of the nodes in the progress set along both directions are evaluated so that the shorter one is identified. The following packets encountering the concave node each are then sent along the corresponding direction toward the progress node in the progress set or the neighbor set. Simulations show that GPSR, GFG, GOAFR+, and GPVFR each conduct a shorter routing path, if augmented with the ProgressFace algorithm.

Index Terms—Wireless ad hoc network, routing protocol, guaranteed packet delivery.

1 INTRODUCTION

Wireless ad hoc networks consist of multiple nodes, each equipped with a wireless transceiver to communicate with each other via peer-to-peer transmission. There exist a wide range of applications for wireless ad hoc networks, including disaster relief, conference, environmental monitoring, and so on. In wireless ad hoc networks, routing is an important issue and has received considerable attention [1], [2], [3], [4], [5], [6]. Recently, the popularity of the global positioning system (GPS) facilitates the use of GPS in routing protocols. In this paper, we undertake the development of geographic routing protocols in wireless ad hoc networks.

MFR [7], DIR [8], and GEDIR [9] route packets greedily based on the physical addresses of nodes. In MFR, a packet is forwarded to the neighbor whose projection onto the line passing through the forwarding node and the destination is closest to the destination. In DIR, a packet is forwarded to the neighbor in the direction closest to that of the destination. In GEDIR, a packet is forwarded to the neighbor closest to the destination. MFR, DIR, and GEDIR cannot guarantee packet delivery because each suffers from the concave node problem [10], [11].

RANDOMWAY [12] and RandHT [13] are proposed to improve the throughput which significantly drops due to the packet traffic concentration near the hole. In RANDOMWAY, several copies of a packet are generated, each of which is forwarded toward a random point before it is forwarded toward the destination. In RandHT, a hole is enclosed by two concentric squares. If a packet is forwarded to a node in the annular region between two concentric squares, the packet is forwarded toward a random point in the closest corner of the annular region and then toward another corner to bypass the hole. RANDOMWAY and RandHT cannot guarantee packet delivery except that they are augmented with GPSR-like routing protocols.

In GPSR-like routing protocols, greedy forwarding and perimeter forwarding are used to forward packets alternately. A packet is forwarded greedily using the geographic distance to the destination until the packet encounters a node with no neighbor closer to the destination (concave node), and forwarded using perimeter forwarding until the packet reaches a node closer to the destination than the concave node (progress node). In GPSR [14] and GFG [15], a planar subgraph is constructed to partition the plane into regions, called faces, and a packet is routed along the face boundary using the right-hand rule during perimeter forwarding. Once the packet encounters an edge intersecting with the line segment from the concave node to the destination at a point that is closer to the destination than the last encountered intersection point, the packet is routed along the face boundary containing the intersected edge and the next edge of the intersected edge in the counterclockwise direction in GPSR, and along the
face boundary intersecting with the line segment from the encountered intersection point to the destination in GFG. In GOAFR+ [16], perimeter forwarding routes a packet along the face boundary using the right-hand rule and the left-hand rule alternately within a circle centered at the destination. Once the face boundary is traversed completely, the packet is routed along the face boundary intersecting with the line segment from the node that is closest to the destination on the completely-traversed face boundary to the destination. In GPVFR [17], each node obtains the physical addresses of the nodes within $h$ hops on the planar subgraph. As a packet encounters a concave node, hop forwarding routes the packet to the progress node closest to the destination if at least one progress node is within $h$ hops on the planar subgraph; otherwise, perimeter forwarding routes the packet along the face boundary in the direction toward the $h$-hop neighbor closest to the destination using the right-hand rule or the left-hand rule. Once the packet encounters an edge intersecting with the line segment from the concave node to the destination, the first encountered end node of the intersected edge is set as the new concave node, and the packet is routed along the face boundary intersecting with the line segment from the new concave node to the destination. It has been proved that if the planar subgraph is a Gabriel Graph (GG) [18] or a Relative Neighborhood Graph (RNG) [19], only one face boundary is traversed during each perimeter forwarding because at least one end node of the edge intersecting with the line segment from the concave node to the destination is a progress node [20], implying GPSR, GFG, GOAFR+, and GPVFR each guarantee packet delivery in GG or RNG.

In GCFR [21], a packet is forwarded greedily until the packet encounters a concave node, and then forwarded using perimeter forwarding hereafter. During perimeter forwarding, packets are forwarded along the face boundary using the right-hand rule and the left-hand rule in parallel. Once a packet encounters an edge intersecting with the line segment from the concave node to the destination, two copies of the packet are generated. The packet continues to be forwarded along the original face boundary, and two copies of the packet are forwarded along the face boundary intersecting with the line segment from the intersection point to the destination using the right-hand rule and the left-hand rule, respectively. In BoundHole [11] and FaceTrace [22], no planar subgraph is constructed. Using the right-hand rule, BoundHole identifies the boundaries of holes to route packets in the perimeter forwarding. BoundHole is efficient in routing because all edges are used in the perimeter routing; it, however, does not guarantee packet delivery. In FaceTrace, a face is delimited by a cluster graph consisting of cluster heads, in which two edges are consecutive in the face boundary only if they are consecutive edges incident to a node in the counterclockwise direction in the cluster graph. Each node in a face boundary obtains the physical addresses of the selected sample nodes in the face boundary, and the node forwards the packet to traverse the face boundary that contains the sample node closest to the destination. To guarantee packet delivery, a packet is routed from face to face using depth-first search (DFS), in which case a packet needs to record the faces whose boundaries have been traversed by the packet.

During perimeter forwarding of the existing GPSR-like routing protocols, a concave node has no information about progress nodes; therefore, the concave node often routes the packet along the face boundary in the wrong direction, resulting in inefficient routing. In this paper, we propose an algorithm, termed ProgressFace, that uses an additional traversal step to decide the direction of perimeter forwarding. By sending a short packet along the face boundary, a concave node constructs the progress set such that, for any destination, at least one progress node is in the progress set except that some progress node exists in the neighbor set, and identifies the directions of perimeter forwarding to reach the progress nodes in the progress set. With the help, the following packets encountering the concave node each are then sent along the corresponding direction toward the progress node in the progress set or the neighbor set, and thus the routing efficiency can be improved. In addition, as a large number of packets are transmitted in the network, the message overhead, including the data message overhead and the protocol message overhead, can be reduced; therefore, the ProgressFace algorithm can be used to improve the performance of the GPSR-like routing protocols in terms of message overhead in applications in which a large number of packets are transmitted. The remainder of this paper is organized as follows. Necessary notations are introduced in Section 2. In Section 3, the progress set construction protocol is proposed to construct the progress sets of nodes on GG. In Section 4, the ProgressFace algorithm is presented to route a packet based on progress sets during perimeter forwarding. We analyze the ProgressFace algorithm in Section 5, and evaluate, by simulations, the performance of the ProgressFace algorithm in Section 6. Finally, we conclude this paper in Section 7.

2 Progress Set and Stuck Region

Progress sets and stuck regions are first introduced. Progress sets are used to improve the routing efficiency of perimeter forwarding, as described in Section 4, and stuck regions are used to construct the progress sets of nodes, as described in Section 3. Subsequently, we show the property of stuck regions. In this paper, the radian of angle $\angle xyu$ is measured clockwise from the line segment between $u$ and $x$ to the line segment between $u$ and $y$.

Definition 1. Let $u$ be a node in graph $G$, and let $u_0, u_1, \ldots, u_n$ $(u_0 = u_n)$ be the neighbors of $u$ in the clockwise direction in $G$. The stuck region of angle $\angle u_1u_{i+1}$, denoted by $SR(\angle u_1u_{i+1})$, is the collection of points $p$, such that $D(u, p) > R$ and $\pi/3 < \angle u_1up < \angle u_1u_{i+1} - \pi/3$, where $R$ is the transmission range of nodes. The stuck region is the set of points that are not within the transmission range of a node, but are close enough to be within the transmission range of the next node in the clockwise direction.
where \( D(u, p) \) denotes the geographic distance between \( u \) and \( p \), and \( R \) denotes the transmission range of \( u \).

**Definition 2.** Let \( u \) be a node in graph \( G \). A progress set of \( u \), denoted by \( PS(u) \), is a set of the non-neighbors of \( u \), such that for each node \( w \neq u \) in \( G \) at least one node \( v \) with \( D(v, w) < D(u, w) \) is in \( PS(u) \cup N(u) \), where \( N(u) \) denotes the set of the neighbors of \( u \).

**Example 1.** Stuck regions of angles \( \angle qxs \) and \( \angle oxq \) in \( G \) are illustrated in Fig. 1(a), in which \( q, s, o, x \) are the neighbors of \( x \) in the clockwise direction. According to Definition 1, point \( p_1 \) lies in \( SR(\angle qxs) \) because \( D(x, p_1) > R \) and \( \pi/3 < \angle qxp_1 < \angle qxs - \pi/3 \); similarly, point \( p_2 \) lies in \( SR(\angle oxq) \) because \( D(x, p_2) > R \) and \( \pi/3 < \angle oxp_2 < \angle oxq - \pi/3 \). \( SR(\angle zxo) \) does not exist. This observation results from \( \angle zxo \leq 2\pi/3 \). Additionally, for each node \( w \neq x \) in \( G \), there exists at least one node \( v \in \{a, z, o, s, q\} \) such that \( D(v, w) < D(x, w) \); therefore, \( \{a, z\} \) is a progress set of \( x \) according to Definition 2. Fig. 1(b) illustrates the stuck region of angle \( \angle ryh \).

Because at least one node \( v \in \{f, k, l, n, h, r\} \) satisfies \( D(v, w) < D(y, w) \) for each node \( w \neq y \) in \( G \), \( \{f, k, l, n\} \) is a progress set of \( y \). Additionally, \( D(v, g) \geq D(y, g) \) for all nodes \( v \neq f, g \) in \( G \); therefore, each progress set of \( y \) contains either \( f \) or \( g \). Also, at least one of \( j \) and \( k \), at least one of \( l \) and \( m \), and at least one of \( n \) and \( p \) are in each progress set of \( y \). Consequently, \( \{f, k, l, n\} \) is a minimum progress set of \( y \). It is noted that the progress set of \( y \) can be used to improve the routing efficiency of perimeter forwarding. Take a packet routed from \( y \) to \( d \), for example, where \( y \) is a concave node. Clearly, the best way is to route the packet along the face boundary in the counterclockwise direction using perimeter forwarding until the first encountered progress node is reached, and then greedily route the packet to \( d \). Suppose that \( y \) has constructed the progress set and identified the better direction to reach each node in the progress set along the face boundary. Then, \( y \) finds the progress node, \( n \), in the progress set, and routes the packet toward \( n \) along the face boundary in the counterclockwise direction, as desired.

The following result shows that if \( u \) is a concave node for destination \( w, w \) lies in one stuck region of \( u \).

**Theorem 1.** Let \( u \) and \( w \) be nodes in \( G \), and let \( R \) be the transmission range of \( u \). If \( R < D(u, w) \leq D(v, w) \) for all nodes \( v \in N(u) \), \( w \) lies in one stuck region of \( u \).

**Proof:** Let \( u_0, u_1, \ldots, u_n \) (\( u_0 = u_n \)) be the neighbors of \( u \) in the clockwise direction in \( G \), and let \( \angle u_i w u \neq \angle u_i w u_{i+1} \) for some \( i \). Consider the triangle \( u_i w u \). \( \angle u_i w u \geq \angle u_i w u_i \geq \angle uu_i D(\angle u_i w, w) > D(u_i, w) \); therefore, \( \angle u_i w u > \pi/3 \). Also, \( \angle uu_{i+1} > \pi/3 \) can be proved in an analogous manner. Consequently, \( w \) lies in \( SR(\angle u_i w u_{i+1}) \) according to Definition 1.

\[ \square \]

3 **THE PROGRESS SET CONSTRUCTION PROTOCOL**

Each node is assumed to be static and have a unique ID. We also assume that the network is connected and any two nodes do not lie on the same point. Using the progress set construction protocol, \( u \) delimits one or more zones for each stuck region of \( u \) to construct the progress set. The following notations are used for the description of the progress set construction protocol.

**Definition 3.** The \( j \)-th \((j \geq 1)\) zone of angle \( \angle u_i w u_{i+1} \), denoted by \( Zone_j(\angle u_i w u_{i+1}) \), is the collection of points \( p \), such that \( D(u, p) > R \) and \( j \pi/3 < \angle u_i p u \leq (j + 1) \pi/3 < \angle u_i w u_{i+1} \).

**Definition 4.** The \( j \)-th \((j \geq 1)\) sector of angle \( \angle u_i w u_{i+1} \), denoted by \( Sector_j(\angle u_i w u_{i+1}) \), is the collection of points \( p \), such that \( D(u, p) \leq R \) and \( j \pi/3 < \angle u_i p u \leq (j + 1) \pi/3 < \angle u_i w u_{i+1} \).

**Definition 5.** The Gabriel Graph of graph \( G \), denoted by \( GG(G) \), is a subgraph of \( G \), such that an edge \((u, v)\) is in

\[ \text{Fig. 1. Stuck regions in } G: \text{ (a) the stuck region of angle } \angle qxs, SR(\angle qxs), \text{ is shown in grey, and the stuck region of angle } \angle oxq, SR(\angle oxq), \text{ is shown in silver; (b) the stuck region of angle } \angle ryh, SR(\angle ryh), \text{ is shown in grey.} \]
$$GG(G)$$ if no other node $$w$$ lies within the circle having diameter $$D(u, v)$$ and passing through nodes $$u$$ and $$v$$.

**Definition 6.** Let $$u_0, u_1, \ldots, u_n$$ be the neighbors of $$u$$ in the clockwise direction in graph $$G$$. The face with respect to angle $$\angle u_i u_{i+1} u$$, denoted by $$\text{Face}(\angle u_i u_{i+1} u)$$, is the face in $$GG(G)$$ with a boundary having interior angle $$\angle u_i u_{i+1}$$, in which $$u_i$$ and $$u_{i+1}$$ denote the neighbors of $$u$$ in $$GG(G)$$, such that angles $$\angle u_i u_{i+1}$$ and $$\angle u_{i+1} u_i$$ are at minima, respectively.

**Example 2.** Zones and sectors of node $$y$$ are illustrated in Fig. 2. According to Definition 3, point $$p_1$$ lies in $$\text{Zone}_4(\angle ryh)$$ because $$D(y, p_1) > R$$ and $$4\pi/3 < \angle ryp_1 \leq 5\pi/3 < \angle ryh$$. According to Definition 4, point $$p_2$$ lies in $$\text{Sector}_4(\angle ryh)$$ because $$D(y, p_2) \leq R$$ and $$4\pi/3 < \angle ryp_2 \leq 5\pi/3 < \angle ryh$$. Fig. 3 shows faces in $$GG(G)$$. According to Definition 5, edge $$(y, r)$$ of $$G$$ is not in $$GG(G)$$ because node $$h$$ lies within the circle having diameter $$D(y, r)$$ and passing through $$y$$ and $$r$$. The plane is partitioned into regions each bounded by a closed walk in $$GG(G)$$, called a face. The closed walk, which refers to the boundary of a face, can be traversed completely using the right-hand rule or the left-hand rule, where the forwarding direction of the right-hand rule used here is identical to the one used in GPDR [14], which is opposite to the one used in GFG [15]. As illustrated in Fig. 3, if the boundary of face $$F_1$$ is traversed using the right-hand rule, edge $$(p, s)$$ is traversed after edge $$(n, p)$$ is traversed because edge $$(p, s)$$ is the next edge of edge $$(n, p)$$ in the counterclockwise direction in $$GG(G)$$; if the boundary of face $$F_1$$ is traversed using the left-hand rule, edge $$(p, n)$$ is traversed after edge $$(s, p)$$ is traversed because edge $$(p, n)$$ is the next edge of edge $$(s, p)$$ in the clockwise direction in $$GG(G)$$. Additionally, because $$h$$ minimizes angle $$\angle u_{k+1} y h$$ among all neighbors $$u_k$$ of $$y$$ in $$GG(G)$$, we have $$u_k = h$$; similarly, because $$h$$ minimizes angle $$\angle u_{i} y h$$ among all neighbors $$u_i$$ of $$y$$ in $$GG(G)$$, we have $$u_i = h$$. Therefore, Face($$\angle ryh$$) denotes the face $$F_1$$ because $$\angle u_{k} y u_i = \angle h y h$$ is an interior angle of the boundary of face $$F_1$$ according to Definition 6. Similarly, Face($$\angle spn$$) denotes face $$F_1$$ because $$\angle u_{k} w u_i = \angle spn$$ is an interior angle of the boundary of face $$F_1$$.

The idea of the progress set construction protocol comes from the observation that for each $$\text{Zone}_j(\angle u_i u_{i+1} u)$$, the node in $$\text{Zone}_j(\angle u_i u_{i+1} u)$$ closest to $$u$$ is a progress node for any node in $$\text{Zone}_j(\angle u_i u_{i+1} u)$$, as proved in Lemma 1, and that the node in $$\text{Zone}_j(\angle u_i u_{i+1} u)$$ closest to $$u$$ is within the 1-hop neighborhood of some node on the boundary of $$\text{Face}(\angle u_i u_{i+1} u)$$, as proved in Lemma 4, where node $$x$$ is said to be within the 1-hop neighborhood of node $$y$$ if $$x \in N(y) \cup \{y\}$$. In the progress set construction protocol, $$u$$ generates a short packet that is forwarded along the face boundary to add the node closest to $$u$$ in each $$\text{Zone}_j(\angle u_i u_{i+1} u)$$ in PS($$u$$). During the traversal, the hop distance from $$u$$ to each node in PS($$u$$) is also evaluated so that the better direction to reach the node in PS($$u$$) is identified.

The implementation of the progress set construction protocol, in which a node in $$\text{Zone}_j(\angle u_i u_{i+1} u)$$ is added in PS($$u$$), if the node is closest to node $$u$$ among all nodes that lie in $$\text{Zone}_j(\angle u_i u_{i+1} u)$$ and within the 1-hop neighborhood of some node on the boundary of $$\text{Face}(\angle u_i u_{i+1} u)$$, is described as follows. For each $$\text{SR}(\angle u_i u_{i+1} u)$$, $$u$$ generates a PSSEARCH packet which contains the IDs and the physical addresses of $$u$$, $$u_i$$, and $$u_{i+1}$$, and a hop counter, $$hc$$, that is initially set to 0 and is advanced in increments by the forwarding node. The PSSEARCH packet is forwarded along the boundary of $$\text{Face}(\angle u_i u_{i+1} u)$$ using the right-hand rule. If a node $$w$$ receives the PSSEARCH packet, $$w$$ first sets the hop counter of each node $$w_j$$, $$w_j$$, $$hc$$, to $$hc$$ in the packet, where $$w_j$$ denotes the node closest to $$u$$, or, in case of a tie, the node that has the smallest ID among all nodes that lie in $$\text{Zone}_j(\angle u_i u_{i+1} u)$$ and within the 1-hop neighborhood of $$u$$. Subsequently, for each $$w_j$$, $$w$$ attaches the ID, the physical address, and the hop...
counter of \( w_j \) to the \( PS\_SEARCH \) packet if no node in \( Zone_j(\angle u_jw_{i+1}) \) has been attached, and substitutes the ID, the physical address, and the hop counter of \( w_j \) for the ID, the physical address, and the hop counter of \( v \) in the packet, respectively, if \( v \) lies in \( Zone_j(\angle u_jw_{i+1}) \) and \( D(u, v) > D(u, w_j) \). If \( u \) receives the \( PS\_SEARCH \) packet, all nodes attached to the packet are added in \( PS(u) \), and the physical addresses of the nodes in \( PS(u) \) are stored. Additionally, for each node \( v \) in \( PS(u) \), the existence of a shorter path from \( u \) to \( v \) on the boundary of \( Face(\angle u_jw_{i+1}) \) traversed using the right-hand rule rather than the left-hand rule is indicated by the non-zero value of the corresponding bit of \( v \), \( v.bit \). \( u \) sets \( v.bit \) to 1 if \( v.hc \leq u.hc - v.hc \); otherwise, \( u \) sets \( v.bit \) to 0.

**Example 3.** The progress set of node \( y \), \( PS(y) \), constructed by the progress set construction protocol is illustrated in Fig. 4. For \( SR(\angle ryh) \), \( y \) first generates a \( PS\_SEARCH \) packet which contains the IDs and the physical addresses of nodes \( y, r, \) and \( h \), and a hop counter, \( hc \), equal to 0. Subsequently, the \( PS\_SEARCH \) packet is forwarded to \( h \) using the right-hand rule. After \( h \) receives the \( PS\_SEARCH \) packet, it updates \( hc \) to 1 and forwards the packet to \( r \). The \( PS\_SEARCH \) packet is forwarded along the boundary of \( Face(\angle ryh) \) using the right-hand rule. After \( e \) receives the \( PS\_SEARCH \) packet, it updates \( hc \) to 6. Because \( f \) is closest to \( y \) among all nodes that lie in \( Zone_1(\angle ryh) \) and within the 1-hop neighborhood of \( e \), \( f.hc \) is set to 6 and is attached to the \( PS\_SEARCH \) packet in addition to the ID and the physical address of \( f \). The packet is then forwarded to \( f \). Because \( g \) is closest to \( y \) among all nodes that lie in \( Zone_1(\angle ryh) \) and within the 1-hop neighborhood of \( f \), the hop counter of \( g \) is set to 7. Then, \( f \) substitutes the ID, the physical address, and the hop counter of \( g \) for the ID, the physical address, and the hop counter of \( f \) in the packet, respectively, because \( f \) lies in \( Zone_1(\angle ryh) \) and \( D(y, f) > D(y, g) \). The \( PS\_SEARCH \) packet is eventually forwarded to \( y \) in which the IDs, the physical addresses, and the hop counters of \( g, k, m, \) and \( n \) are attached. Therefore, \( PS(y) \), constructed by \( y \), equals the set \{ \( y, k, m, n \) \}. Additionally, \( g.bit, k.bit, m.bit, \) and \( n.bit \) are set to 1, 1, 0, and 0, respectively, because \( g.hc = 7, k.hc = 12, m.hc = 17, n.hc = 22, \) and \( hc = 32 \).

As the GPSR-like routing protocol is augmented with the ProgressFace algorithm, the progress set construction protocol can proceed proactively or reactively. In the proactive version, each node begins the progress set construction protocol for each stuck region, if it exists, before the routing task is issued. In the reactive version, the progress set construction protocol proceeds as the packet reaches a concave node that does not have the progress set for the stuck region where the destination lies. In the reactive version, only necessary progress sets are constructed; therefore, the routing in the reactive version requires a smaller average protocol message overhead while incurring an additional latency, compared to the one in the proactive version.

### 4 The ProgressFace Algorithm

As the GPSR-like routing protocol is augmented with the ProgressFace algorithm, the ProgressFace algorithm routes a packet based on the progress sets constructed by the progress set construction protocol during perimeter forwarding. In the ProgressFace algorithm, the concave node \( u \) routes a packet toward a node \( v \) in \( PS(u) \) closest to the destination using the right-hand rule and the left-hand rule until it encounters a node with a progress node in the 1-hop neighborhood if \( v.bit = 1 \) and \( v.bit = 0 \), respectively. Take GPSR augmented with the ProgressFace algorithm, denoted by GPSR+PF, for illustration. Let node \( u \) have a packet with the destination \( d \). During greedy forwarding, \( u \) forwards the packet to a neighbor, \( w_i \) with \( D(w_i, d) < D(u, d) \) and \( D(w_i, d) \leq D(t, d) \) for all nodes \( t \in N(u) \). If \( D(u, d) \leq D(t, d) \) for all nodes \( t \in N(u) \), GPSR+PF proceeds to perimeter forwarding to route the packet along the boundary of \( Face(\angle u_jw_{i+1}) \), assuming that \( d \) lies in \( SR(\angle u_jw_{i+1}) \). In this case, \( u \), called the initiator of perimeter forwarding, attaches...
the physical address of \( u \) and the corresponding bit of \( v \) to the packet, where \( v \) is the node in \( PS(u) \) such that \( D(v, d) \leq D(t, d) \) for all nodes \( t \in PS(u) \). During perimeter forwarding, GPSR+PF proceeds to greedy forwarding, if the packet reaches a node having a neighbor closer to the destination than the initiator of perimeter forwarding; otherwise, the packet is forwarded to the neighbor using the right-hand rule and the left-hand rule if \( v.bit = 1 \) and \( v.bit = 0 \), respectively.

**Example 4.** A packet routed from node \( r \) to node \( d \) using GPSR augmented with the ProgressFace algorithm is illustrated in Fig. 5. Initially, greedy forwarding forwards the packet from \( r \) to \( y \) because \( D(y, d) < D(r, d) \) and \( D(y, d) \leq D(t, d) \) for all nodes \( t \in N(r) = \{h, y, z\} \). After \( y \) receives the packet, the routing protocol proceeds to the perimeter forwarding and routes the packet along the boundary of \( Face(\angle rh) \), the boundary of face \( F1 \) shown in Fig. 3, using the ProgressFace algorithm because \( D(y, d) \leq D(t, d) \) for all nodes \( t \in N(y) = \{h, r\} \) and \( d \) lies in \( SR(\angle rh) \). Because \( n \in PS(y) \) and \( D(n, d) \leq D(t, d) \) for all nodes \( t \in PS(y) \), \( y \) attaches the physical address of \( y \) and the corresponding bit of \( n \) to the packet. \( y \) then forwards the packet to \( h \) using the left-hand rule because \( n.bit = 0 \). Using the left-hand rule, the packet is forwarded to \( q \). Meanwhile, the routing protocol proceeds to greedy forwarding because \( x \in N(q) \) and \( D(x, d) > D(y, d) \). Using greedy forwarding, the packet is eventually forwarded to \( d \).

The GPSR-like routing protocol augmented with the ProgressFace algorithm introduces a protocol message overhead, and incurs an additional latency if the progress set construction protocol proceeds reactively. It, however, conducts a short routing path, and thus requires a small data message overhead. The progress set of a node for a stuck region can be used to route any packet that encounters the node and has the destination in the stuck region; therefore, the ProgressFace algorithm can improve the GPSR-like routing protocol in terms of the total of the data message overhead and the protocol message overhead in applications in which a large number of packets are transmitted.

## 5 Analysis of the ProgressFace Algorithm

We first show that the progress set construction protocol indeed constructs a progress set of a node. Subsequently, we compute the size of the progress set constructed by the progress set construction protocol. Finally, the complexity of the progress set construction protocol is analyzed.

### 5.1 Correctness of the Progress Set Construction Protocol

Let \( v \) be the node closest to a node, \( u \), in Zone\(_j(\angle u_iu_{i+1}) \). Lemma 1 shows that \( v \) is a progress node of \( u \) for any destination \( w \) in Zone\(_j(\angle u_iu_{i+1}) \). Subsequently, Lemma 4 shows that \( v \) must be within the 1-hop neighborhood of some node on the boundary of \( Face(\angle u_iu_{i+1}) \) with the exception that some neighbor of \( u \) is a progress node of \( w \) in Zone\(_j(\angle u_iu_{i+1}) \). Finally, Theorem 2 shows that \( PS(u) \), constructed by the progress set construction protocol, is a progress set of \( u \). Additionally, Lemmas 2 and 3 are used to prove Lemma 4.

**Lemma 1.** Let \( v \) be closest to node \( u \) among all nodes in Zone\(_j(\angle u_iu_{i+1}) \). Then, \( D(v, w) \leq D(u, w) \) for all nodes \( w \) in Zone\(_j(\angle u_iu_{i+1}) \).

**Proof:** Assume that \( D(v, w) \geq D(u, w) \). Due to symmetry, consider \( \angle u_iuw \geq \angle u_iuw \). In triangle \( uuv \), \( D(v, w) \geq D(u, w) \geq D(u, v) \); therefore, \( \angle uuv \geq \pi/3 \). It implies that \( u \) and \( w \) are not in the same zone. It is a contradiction. \( \square \)

**Lemma 2.** No node lies in a sector.

**Proof:** Assume that \( w \) lies in \( Sector_j(\angle u_iu_{i+1}) \). According to Definition 4, \( j \pi/3 < \angle u_iuw \geq (j + 1) \pi/3 < \angle u_iu_{i+1} \), and \( D(u, w) \leq R \). It implies that \( w \in N(u) \). Because \( \angle u_iuw < \angle u_iu_{i+1} \), \( u_i \) and \( u_{i+1} \) are not consecutive neighbors of \( u \) in the clockwise direction, a contradiction. \( \square \)

**Lemma 3.** Let \( v \) be closest to node \( u \) among all nodes in Zone\(_j(\angle u_iu_{i+1}) \). If \( v \) does not lie on the boundary of \( Face(\angle u_iu_{i+1}) \), the line segment from \( u \) to \( v \) intersects one edge, \((x, y)\), on the boundary of \( Face(\angle u_iu_{i+1}) \) at point \( p \) on which neither of \( x \) and \( y \) lies.

**Proof:** Let \((u, u_k)\) and \((u, u_l)\) be edges in \( GG(G) \) such that \( \angle u_ku_i \) and \( \angle u_{i+1}u_l \) are at minima in \( G \), respectively. It implies that \( \angle u_ku_i \) is an interior angle
of the boundary of $\text{Face}(\angle u_{i}u_{i+1}), \angle u_{i}u_{i+1} \leq \angle u_{w}w$, and $\angle u_{i}w \leq \angle u_{i}w$, as illustrated in Fig. 6. According to Lemma 3 of [20], $v$ does not lie in the interior of $\text{Face}(\angle u_{i}u_{i+1})$; therefore, the line segment from $u$ to $v$ does not intersect the boundary of $\text{Face}(\angle u_{i}u_{i+1})$ at one point or one node. We show the latter case does not hold. Assume that the line segment from $u$ to $v$ intersects the boundary of $\text{Face}(\angle u_{i}u_{i+1})$ at one node, $x$. Because $v$ is closest to $u$ among all nodes in $\text{Sector}_j(\angle u_{i}u_{i+1})$, $x$ must lie in $\text{Sector}_j(\angle u_{i}u_{i+1})$. It is a contradiction because no node lies in $\text{Sector}_j(\angle u_{i}u_{i+1})$ according to Lemma 2.

**Lemma 4.** Let $v$ be closest to node $u$ among all nodes in $\text{Zone}_j(\angle u_{i}u_{i+1})$. Then, $v \in N(u) \setminus \{t\}$ for some node $t$ on the boundary of $\text{Face}(\angle u_{i}u_{i+1})$ with the exception that some node $t \in N(u)$ exists such that $D(t, w) < D(u, w)$ for all nodes $w$ in $\text{Zone}_j(\angle u_{i}u_{i+1})$.

**Proof:** Assume that $v$ does not lie on the boundary of $\text{Face}(\angle u_{i}u_{i+1})$. According to Lemma 3, the line segment from $u$ to $v$ intersects one edge, $(x, y)$, on the boundary of $\text{Face}(\angle u_{i}u_{i+1})$, as illustrated in Fig. 6. It suffices to show at least one of the following conditions is satisfied for some node $t \in \{x, y\}$:

1. $v \in N(t)$;
2. $t \in N(u)$ and $D(t, w) < D(u, w)$ for all nodes $w$ in $\text{Zone}_j(\angle u_{i}u_{i+1})$.

Because edge $(x, y)$ is in $\text{GG}(G)$, no node lies within the circle having diameter $D(x, y)$ and passing through $x$ and $y$; therefore, $\angle xuy < \pi/2$ and $\angle yux < \pi/2$. It implies that, in quadrangle $vxyu$, $\angle vxu > \pi/2$ or $\angle uyv > \pi/2$. The proof of the latter case is omitted due to its similarity with that of the former one. Because $\angle vxu > \pi/2$, $D(u, v) > D(u, u)$ and $D(u, v) > D(v, u)$. It implies that $x$ does not lie in $\text{Zone}_j(\angle u_{i}u_{i+1})$ because $D(u, v) \leq D(u, w)$ for all nodes $w$ in $\text{Zone}_j(\angle u_{i}u_{i+1})$. Two cases are discussed: 1) $D(u, y) \geq D(u, v)$; 2) $D(u, y) < D(u, v)$.

We first prove if $D(u, y) \geq D(u, v)$, condition c1 is satisfied for $t = x$. Assume that $v \notin N(x)$, that is, $D(v, x) > R$. As illustrated in Fig. 7, $x$ must lie in region $A$ because $D(v, x) > R, D(u, x) < D(u, v)$, and $D(v, x) < D(u, v)$; $y$ must lie in region $B$ because $D(y, y) > D(u, v)$. This implies $D(x, y) > R$ because the minimum geographic distance between regions $A$ and $B$ is larger than $R$, which is a contradiction because edge $(x, y)$ is in $\text{GG}(G)$.

Next, we show if $D(u, y) < D(u, v)$, condition c2 is satisfied for $t = y$. In this case, $y$, the same as $x$, does not lie in $\text{Zone}_j(\angle u_{i}u_{i+1})$. Because neither of $x$ and $y$ lies in $\text{Sector}_j(\angle u_{i}u_{i+1})$ according to Lemma 2, $\angle xuy > \pi/3$. It implies that $\angle xuy < \angle xuy$ or $\angle xuy < \angle xuy$; therefore, $D(u, y) < D(x, y)$ or $D(u, y) < D(x, y)$. Because $D(x, y) \leq R, D(u, x) < R$ or $D(u, x) < R$. We first show $D(u, x) > R$. Assume that $D(u, x) < R$, that is, $x \in N(u)$. Because of the existence of $\text{Zone}_j(\angle u_{i}u_{i+1})$ and $\text{Sector}_j(\angle u_{i}u_{i+1})$, $\angle xuy > \pi/3$, which contradicts to $\angle xuy < \pi/2$ because edge $(x, y)$ is in $\text{GG}(G)$. Because $D(u, x) < R$ does not hold, $D(u, y) < R$, that is, $y \in N(u)$. We show $D(y, w) < D(u, w)$ for all nodes $w$ in $\text{Zone}_j(\angle u_{i}u_{i+1})$ to complete the proof. It suffices to show all nodes $w$ in $\text{Zone}_j(\angle u_{i}u_{i+1})$ lie in the same side of the perpendicular bisector of edge $(u, y)$, $L$, as $y$. Let $L_1$ be the collection of points $p$ with $\angle u_{i}up = j\pi/3$, and let $L_2$ be the collection of points $p$ with $\angle u_{i}up = (j+1)\pi/3$, as illustrated in Fig. 8. Because neither of $x$ and $y$ lies in $\text{Sector}_j(\angle u_{i}u_{i+1})$ and $\text{Zone}_j(\angle u_{i}u_{i+1})$, $\angle uxw \leq \pi/3$ and $\angle uyv > (j+1)\pi/3$. In addition, because edge $(x, y)$ is in $\text{GG}(G)$, $\angle xuy < \pi/2$, resulting in the impossibility that $L$ is parallel to $L_1$ or $L_2$. Let edge $(x, y)$ intersect $L_1$ and $L_2$ at points $p_1$ and $p_2$, respectively, and let $L$ intersect $L_1$ and $L_2$ at points $p_3$ and $p_4$, respectively. We only need to show no node in $\text{Zone}_j(\angle u_{i}u_{i+1})$ lies within the triangle $p_3p_4p_3$. $D(u, p_1) < D(u, v)$ and $D(u, p_2) < D(u, v)$ because $D(u, x) < D(u, v)$ and $D(u, y) < D(u, v)$. In...
addition, $x$, the same as $y$, does not lie in the same side of $L$ as $u$ because $D(u,x) > R \geq D(x,y)$. Therefore, $D(u,p_1) < D(u,p_2)$ and $D(u,p_2) < D(u,p_3)$. It implies that $D(u,p_3) < D(u,v)$ and $D(u,p_4) < D(u,v)$, completing the proof.

5.2 Sizes of Progress Sets

**Theorem 2.** Let $PS(u)$ be constructed by the progress set construction protocol. For all nodes $w \notin N(u)$, there exists at least one node $t \in N(u) \cup PS(u)$ such that $D(t,w) < D(u,w)$.

**Proof:** Assume that $D(t,w) \geq D(u,w)$ for all nodes $t \in N(u)$. We need to show $D(t,w) < D(u,w)$ for some node $t \in PS(u)$. According to Theorem 1 and Lemma 2, $w$ must lie in Zone$_j(\angle u_{i-1}u_i)$ for some $i,j$. Because the $PS$ SEARCH packet generated by $u$ for $SR(\angle u_{i-1}u_i)$ traverses all nodes on the boundary of Face$(\angle u_{i-1}u_i)$, the node $v$, closest to $u$ in Zone$_i(\angle u_iu_{i+1})$ is in $PS(u)$ according to Lemma 4. According to Lemma 1, $D(v,w) < D(u,w)$. □

**Theorem 3.** For each node $u$, the number of nodes in $PS(u)$ constructed by the progress set construction protocol is at most 4.

**Proof:** Any two nodes in $PS(u)$ are not in a zone. It suffices to show the number of zones of $u$ is no more than 4. Because $SR(\angle u_{i-1}u_i)$ exists only if $\angle u_{i-1}u_i > 2\pi/3$, at most two stuck regions of $u$ exist. Two cases are discussed: 1) there is only one stuck region of $u$, $SR(\angle u_{i-1}u_i)$; 2) there are two stuck regions of $u$, $SR(\angle u_{i-1}u_i)$ and $SR(\angle u_{k-1}u_k)$. Let $m$ and $n$ be the numbers of zones of angles $\angle u_{i-1}u_i$ and $\angle u_{k-1}u_k$, respectively. According to Definition 3, $(m+1)\pi/3 < \angle u_{i-1}u_i$ and $(n+1)\pi/3 < \angle u_{k-1}u_k$. It is easy to verify that $m \leq 4$ for case 1 and $m + n \leq 3$ for case 2. □
5.3 Complexity

Theorem 4. The progress set construction protocol has a worst-case computation complexity of $O(m\Delta(\Delta + 1))$ per node, where $\Delta$ is the maximum degree and $m$ is the maximum number of nodes in a face boundary.

Proof: If node $u$ receives a PS\_SEARCH packet generated by node $v$, $u$ takes $O(1)$ time to check whether a node in the 1-hop neighborhood is closer to $v$ compared to the node attached in the packet in the corresponding zone. Because there are at most $\Delta + 1$ nodes in the 1-hop neighborhood, $u$ takes $O(\Delta + 1)$ time once it receives a different PS\_SEARCH packet. Clearly, $u$ has at most $\Delta$ neighbors on the planar subgraph, implying $u$ lies on at most $\Delta$ face boundaries. Because $u$ receives at most $m$ different PS\_SEARCH packets generated by nodes on a face boundary, $u$ receives at most a total of $m\Delta$ different PS\_SEARCH packets. Therefore, the overall computation complexity for a node is $O(m\Delta(\Delta + 1))$.

Theorem 5. The progress set construction protocol has a worst-case message exchange complexity of $O(m\Delta)$ per node, where $\Delta$ is the maximum degree and $m$ is the maximum number of nodes in a face boundary.

Proof: Let $v$ be a neighbor of node $u$ on the planar subgraph. $u$ can send the PS\_SEARCH packet to $v$ along exactly one face boundary. Because at most $m$ different PS\_SEARCH packets are forwarded along a face boundary, $u$ sends at most $m$ packets to $v$. Because $u$ has at most $\Delta$ neighbors on the planar subgraph, the overall message exchange complexity for a node is $O(m\Delta)$.

6 Performance Evaluation

Simulations using the packet-level simulator were used to evaluate the performance of the proposed approach.
Network behavior such as packet loss, packet delay, and so on was not taken into consideration. In our simulations, 100 connected networks with densities ranging from 6 to 16 were generated by randomly deploying nodes in square regions with side lengths equal to 15 or 30, where the network density denotes the average number of neighbors per node and the transmission range of nodes is a circle of radius 1. Each square region has 0, 3 or 10 voids, in which the voids, denoting regions without nodes, are circles of radius 2 and could be overlapped. GPSR, GFG, GOAFR+, and GPVFR using the ProgressFace algorithm to route packets during perimeter forwarding were denoted by GPSR+PF, GFG+PF, GOAFR++PF, and GPVFR+PF, respectively, and compared with GPSR, GFG, GOAFR+, and GPVFR, in terms of the perimeter ratio, the path stretch, and the maximum load. The perimeter ratio denotes the ratio of hops using perimeter forwarding. The path stretch denotes the ratio of the routing path length to the shortest path length. The maximum load denotes the maximum number of data packets routed by a node. In the implementation of GPSR, GFG, GOAFR+, or GPVFR, the settings of parameters are the same as that in the original paper. In addition, we compared GPSR, GFG, GOAFR+, and GPVFR with GPSR+PF, GFG+PF, GOAFR++PF, and GPVFR+PF in terms of the message overhead, which denotes the number of packets, including the data packets and the protocol packets used to construct the progress sets, transmitted by a node. The additional latencies of GPSR+PF, GFG+PF, GOAFR++PF, and GPVFR+PF in the reactive version were also evaluated. In our simulations, 100 packets were transmitted between each of 100 or 1000 source-destination pairs in each network. Empirical data were obtained by averaging data of 100 or 1000

\[\text{Fig. 11. Maximum loads of GPSR, GFG, GOAFR+, GPVFR, GPSR+PF, GFG+PF, GOAFR++PF, and GPVFR+PF.} \]

The side length of the square region is 30 in (a) and (b), and 15 in (c) and (d); the number of voids in the square region is 0 in (a) and (c), 10 in (b), and 3 in (d).
source-destination pairs from 100 networks.

### 6.1 Perimeter Ratio

Fig. 9 illustrates the simulation results for perimeter ratios. GPSR and GFG have the same perimeter ratio because they route the packets in the same path in the Gabriel Graph during perimeter forwarding [20]. In GPVFR, hop forwarding routes a packet from a concave node to a progress node except that no progress node is within $h$ hops on the planar subgraph, in which case perimeter forwarding is used. Compared to GPSR and GFG, GPVFR uses less perimeter forwarding, resulting in a smaller perimeter ratio. In addition, in GPSR and GFG, perimeter forwarding routes a packet along the face boundary in a given direction using the right-hand rule, resulting in a long path to reach a progress node closer to reach along the face boundary in the other direction. By contrast, GOAFR+ routes a packet along the face boundary in either direction using either of the right-hand rule and the left-hand rule during perimeter forwarding; GOAFR+ has a smaller perimeter ratio than GPSR and GFG.

The difference in the perimeter ratio between GOAFR+ and GOAFR++PF is negligible because GOAFR+ and GOAFR++PF each route a packet along the face boundary using the right-hand rule and the left-hand rule alternately within a circle centered at the destination during perimeter forwarding. By contrast, GPSR+PF, GFG+PF, and GPVFR+PF have smaller perimeter ratios than GPSR, GFG, and GPVFR, respectively. In GPVFR, perimeter forwarding is used only when the packet encounters a concave node without any progress node within $h$ hops, in which case the hop distance to the closest progress node along the face
Fig. 13. Message overheads of GPSR, GFG, GOAFR+, GPVFR, GPSR+PF, GFG+PF, GOAFR++PF, and GPVFR+PF. The side length of the square region is 30; the number of voids in the square region is 0; the number of source-destination pairs is 100 in (a) and (b), and 1000 in (c) and (d); the network density is 6 in (a) and (c), and 16 in (b) and (d).

boundary is large. Therefore, the ProgressFace algorithm has a smaller impact on GPVFR than on GPSR and GFG, resulting in a smaller difference in the perimeter ratio between GPVFR and GPVFR+PF, as compared to the differences between GPSR and GPSR+PF and between GFG and GFG+PF.

In GOAFR+ and GOAFR++PF, some nodes are traversed more than once by the same packet, resulting in a greater perimeter ratio than GPSR+PF, GFG+PF, and GPVFR+PF. In addition, compared to GPSR+PF and GFG+PF, although GPVFR+PF has less perimeter forwarding, GPVFR+PF has a greater perimeter ratio. This is because GPVFR+PF has a larger hop distance from a concave node to the closest progress node along the face boundary. Moreover, as expected, the greater the network density, the smaller the perimeter ratio. As voids exist in the square region, the perimeter ratio becomes greater. As nodes are deployed in a square region with a side length equal to 15, the perimeter ratio becomes smaller because the packets encounter fewer concave nodes due to the existence of fewer faces with boundaries of large size.

6.2 Path Stretch

Fig. 10 illustrates the simulation results for path stretches. As expected, the smaller the perimeter ratio, the smaller will be the path stretch. GPSR+PF and GFG+PF have the smallest path stretch because of the smallest perimeter ratios. GPSR and GFG have the largest path stretches because they have the greatest perimeter ratios. Each protocol has a smaller path stretch in the network with a greater density, a smaller number of voids, and a smaller size of the square region because of a smaller perimeter ratio.
6.3 Maximum Load

Fig. 11 illustrates the simulation results for maximum loads. The node with the maximum load usually lies on the face boundary around a hole. In GPSR and GFG, the largest number of nodes on the face boundaries around holes are traversed during perimeter forwarding because GPSR and GFG have the greatest perimeter ratio; therefore, GPSR and GFG have the largest maximum loads. In GOAFR+ and GOAFR++PF, some nodes are traversed more than once by the same packet during perimeter forwarding, resulting in larger maximum loads than GPVFR and GPVFR+PF. It is noted that GPSR+PF and GFG+PF have the smallest maximum load because of the smallest perimeter ratio. In the network with a greater density, a smaller number of voids, and a smaller size of the square region, each protocol has a smaller maximum load because of a smaller perimeter ratio.

6.4 Additional Latency

Fig. 12 illustrates the simulation results for additional latencies. GPVFR+PF uses the least perimeter forwarding, resulting in the smallest additional latency. The difference in the additional latency among GPSR+PF, GFG+PF, and GOAFR++PF is negligible. As we might expect, the greater the number of source-destination pairs, the smaller the additional latency. This is because the progress set of a node for a stuck region is constructed only after the first packet whose destination lies in the stuck region encounters the node. In the network with a greater density, a smaller number of voids, and a smaller size of the square region, each protocol has a smaller additional latency because of less perimeter forwarding.

6.5 Message Overhead

Fig. 13 illustrates the simulation results for message overheads. Among GPSR, GFG, GOAFR+, and GPVFR, each of GPSR and GFG has the greatest message overhead due to the greatest path stretch, and GOAFR+ has the smallest message overhead due to the smallest path stretch. Among GPSR+PF, GFG+PF, GOAFR++PF, and GPVFR+PF, each of GPSR+PF and GFG+PF has the smallest message overhead due to the smallest path stretch, and GOAFR++PF has the greatest message overhead due to the greatest path stretch. For each protocol, the reactive version has a smaller message overhead, compared to the original one and the proactive one. In a high-density network, the difference in the message overhead among the original version, the proactive version, and the reactive version is negligible because of a small number perimeter forwarding. In addition, as we might expect, the greater the number of source-destination pairs, the smaller the ratio of the protocol message overhead to the number of source-destination pairs, resulting in, as a large number of packets are transmitted, the message overhead is dominated by the path stretch.

6.6 Comparison of GPSR-like Routing Protocols

The comparison of GPSR, GFG, GOAFR+, GPVFR, GPSR+PF, GFG+PF, GOAFR++PF, and GPVFR+PF is shown in Table 1. The perimeter ratio, the routing efficiency, the maximum load, the additional latency, and the message overhead are ranked by the simulation results for perimeter ratios, path stretches, maximum loads, additional latencies, and message overheads, respectively. In each protocol, a packet is required to carry one or two bits denoting the routing mode and the physical addresses of the concave node and the destination. GPSR, GFG, and GPVFR exhibit the best performance in terms of the routing overhead because each of them does not require the other routing overhead. In GOAFR+, a packet must carry the radius of the current circle, one bit to indicate whether the current circle is hit for the first or the second time, and two counters recording the numbers of the traversed nodes closer to or farther from the destination than the concave node; GOAFR+ has worse performance in terms of the routing overhead, compared to GPSR, GFG, and GPVFR. In GPSR+PF, GFG+PF, GOAFR++PF, and GPVFR+PF, a packet must carry the corresponding bit of the selected progress node; GPVFR+PF has worse performance in terms of the routing overhead than GPSR, GFG, GOAFR+, and GPVFR, respectively. In each protocol, each node must store the physical addresses of 1-hop neighbors. Because the node in GPSR, GFG, and GOAFR+ does not require the other memory overhead, each of them exhibits the best perfor-
formance in terms of the memory overhead. In GPVFR, the node must store the physical addresses of the h-hop neighbors on the planar subgraph; GPVFR have worse performance in terms of the memory overhead, compared to GPSR, GFG, and GOAFR+. In GPSR+PF, GFG+PF, GOAFR++PF, and GPVFR+PF, the node must store the corresponding bit and the physical address of each node in the progress set, where the number of nodes in the progress set is at most 4; GPSR+PF, GFG+PF, GOAFR++PF, and GPVFR+PF has worse performance in terms of the memory overhead than GPSR, GFG, GOAFR+, and GPVFR, respectively. In addition, the adaptability to voids is ranked by the simulation results for the ratio of the sum of the path lengths in square regions with 10 voids, to that in square regions with no void, in which the side length of the square region equals 30; the scalability to large networks is ranked by the simulation results for the ratio of the sum of the path lengths in square regions with side lengths equal to 30, to that in square regions with side lengths equal to 15, in which case no voids exist in the square region; the adaptability to sparse networks is ranked by the simulation results for the ratio of the sum of the path lengths in networks with densities equal to 6, to that in networks with densities equal to 16, in which the side length of the square region equals 30 and the number of voids equals 0.

7 Conclusion

The progress set of a node contains at least one progress node for any destination that is not closer to each neighbor of the node. In this paper, the progress set construction protocol is first proposed to construct the progress sets of nodes on a Gabriel Graph (GG). Subsequently, the ProgressFace algorithm is proposed to route a packet from a concave node to a progress node along the face boundary based on progress sets during perimeter forwarding. As a result, the GPSR-like routing protocol augmented with the ProgressFace algorithm has a high probability of routing a packet from a concave node to the closest progress node on the face boundary during perimeter forwarding. Simulations show that GPSR-like routing protocols GPSR, GFG, and GPVFR augmented with the ProgressFace algorithm, denoted by GPSR+PF, GFG+PF, and GPVFR+PF, respectively, have smaller perimeter ratios, routing paths, maximum loads, and message overheads, compared to GPSR, GFG, and GPVFR, respectively. In terms of the perimeter ratio, the routing path, the maximum load, and the message overhead, the difference between GOAFR+ and GOAFR++ augmented with the ProgressFace algorithm, denoted by GOAFR++PF, is negligible. In addition, GPSR+PF, GFG+PF, GOAFR++PF, and GPVFR+PF each incur a moderate additional latency. In our simulations, GPSR+PF and GFG+PF each have the best performance in terms of the perimeter ratio, the routing path, the maximum load, and the message overhead. Future research includes study of the manner in which to construct and maintain the progress set in a mobile network. Because GG contains short edges in a dense network with voids, leading to a large hop count between nodes, additional research will include study of the manner in which to construct the progress set on the other planar subgraphs.

References

**Chia-Hung Lin** received the B.S. degree in information engineering and computer science from Feng Chia University, in 2003. He is currently working toward the Ph.D. degree in computer science, National Tsing Hua University, Taiwan. His current research interests include mobile computing, wireless networks, and ad hoc wireless networks.

**Shiao-An Yuan** received the B.S. and M.S. degrees in computer sciences from Tung Hai University and from National Tsing Hua University in 2007 and 2008, respectively. His research interests is in distributed computing in wireless sensor networks.

**Shih-Wei Chiu** received the B.S. degree in computer science and information engineering from National Cheng Kung University in 2008. He is currently working toward the M.S. degree in computer science, National Tsing Hua University. His research interests include distributed and mobile computing.

**Ming-Jer Tsai** received the Ph.D. degree in electrical engineering from National Taiwan University in 1997. Since then, he joins Computer and Communication Laboratory, Industrial Technology Research Institute. In 2003, he joins Department of Computer Science, National Tsing Hua University, where he is currently an associate professor. His research interests include distributed systems and mobile computing. Dr. Tsai is a member of IEEE.