Geographic and Energy-Aware Routing in Sensor Networks

HANNES FREY
University of Trier, Trier, Germany

IVAN STOJMENOVIĆ
University of Ottawa, Ontario, Canada

Typical communication patterns within a sensor network are data delivery from sensor nodes to one of selected information sinks, and information sinks requesting a certain physical phenomenon or requesting sensor nodes lying within a sensed area. In general, addressing is achieved by utilizing sensor locations. Geographic routing algorithms allow routers to be nearly stateless since packet forwarding is achieved by utilizing location information about candidate nodes in vicinity and the location of the final destination only. By their localized nature, geographic routing algorithms are highly scalable solutions which do not require any additional control overhead when network topology changes due to mobility or energy conserving sleep cycles. Recent work investigated that location information may be utilized to define new link metrics aiming on energy and physical layer optimized routing paths instead of only minimizing the number of hops needed to reach the desired destination. This chapter reviews geographic and energy aware routing algorithms for sensor networks. It includes simple heuristic greedy forwarding strategies, strategies which obtain guaranteed delivery by memorizing information about all ongoing routing task, memoryless recovery strategies, energy aware ~~link metrics and~~ routing strategies aiming on increased network lifetime, and routing without information about their neighbor nodes. The majority of geographic routing protocols assume a simplified network model which does not take into account random variations in correct message receipt. This chapter also discusses physical

Handbook of Sensor Networks: Algorithms and Architectures, Edited by I. Stojmenović
ISBN 0-471-68472-4

layer impact on both greedy geographical routing strategies and their recovery strategies.

12.1 INTRODUCTION

Sensor networks are typically ~~comprise~~ thousands of small collaborating wireless sensor nodes that have limited computation and communication capabilities. ~~Typical~~ communication patterns within a sensor network are data delivery from sensor nodes to one or a subset of selected information sinks, and information sinks requesting a certain physical phenomenon or requesting sensor nodes lying within a sensed area. In general, addressing is achieved by utilizing sensor locations or querying all sensor nodes matching a certain criterion instead of utilizing individual node addresses. Ease of deployment and the fact that sensor nodes are small and closely located at the measured phenomenon makes an external power supply, recharging batteries, or replacing depleted batteries impractical or even impossible. Consequently, the lifetime of a sensor node is directly related to its on-board power supply, and thus energy is the most sensitive resource with respect to the whole network lifetime.

Communication between sensor nodes and information sinks can be achieved by setting an appropriate transmission power (if possible) and sending data or control messages directly to the desired recipient. However, this simple communication form may degrade the bandwidth of the limited shared wireless communication media, and moreover will drastically increase energy consumption at sender nodes, since signal attenuation increases significantly with the distance to the message recipient. If no fixed networked infrastructure is additionally available, a resource-saving communication may only be achieved by multihop ad hoc routing techniques, where communication between any two network nodes requires collaborating with intermediate next-hop forwarding nodes.

Since location information is often available due to the very nature of sensor networks, the special class of geographic routing algorithms may be a good choice in order to build a scalable resource-saving communication infrastructure. Geographic routing algorithms allow routers to be nearly stateless, since packet forwarding is achieved by utilizing location information about candidate nodes in the vicinity and the location of the final destination only. By their localized nature, geographic routing algorithms are highly scalable solutions that do not require any additional control overhead when network topology changes due to mobility or energy-conserving sleep cycles. In particular, due to the addressing scheme of sensor networks there is no need for an additional location service (producing an additional network load), which is used in other ad hoc network scenarios in order to acquire location information about individual network nodes before communication can take place. Finally, location information about all neighbor nodes can be used in order to estimate the signal strength needed to reach a certain neighbor node. Recent work investigated that location information can be utilized to define new link metrics aiming at energy and physical-layer optimized routing paths instead of only minimizing the number of hops needed to reach the desired destination.

12.2 GREEDY ROUTING ALGORITHMS

Greedy routing algorithms limit forwarding decisions on information about the position of all nodes in the vicinity and forward a message to the "best" neighbor regarding the position of the final destination and the metric being optimized. Each forwarding node applies this greedy principle until the final destination (if possible) is finally reached. The current required location information about neighbor nodes is maintained by proactively exchanging short beacon messages (containing node ID and location) transmitted with maximum signal strength.

12.2.1 Progress, Distance, and Direction

The first geographic routing algorithm was described by Takagi and Kleinrock in the mid-1980s [1]. They introduced the notion of *progress* in order to define the *most forward within radius* (*MFR)* greedy routing scheme. The distance between the current node S and the projection A' of a neighbor node A onto the line SD connecting S and final-destination node D is termed progress (see node A in Fig. 12.1). MFR selects the neighbor node that maximizes progress, while nodes with negative progress are ignored (e.g., MFR selects node A in Fig. 12.1).[1] Alternatively, *distance-based* greedy forwarding considers *Euclidean distance* instead of progress. Finn [3] proposed the first distance-based greedy routing scheme, which selects a node closer and minimizes the distance d to the final destination (e.g., node B in Fig. 12.1). This scheme is the most widely applied greedy strategy in the literature, and it will subsequently be referred as *GREEDY*. In recent years *direction-based* (*DIR*) greedy routing, which considers the angle between the next hop, current, and destination nodes, was investigated as a third alternative of greedy forwarding. The DIR method, described by Kranakis et al. [4], selects the next hop forwarding node, minimizing the deviation from the line connecting the current and the destination node (e.g., node C in Fig. 12.1).

There are several variants of nodes that will be considered, along with the stopping criterion in progress or distance-based routing schemes. In originally proposed articles, greedy forwarding based on progress or distance considers nodes in the forward direction (respectively closer to destination) only (e.g., nodes A, B, C, and E in Fig. 12.1), since choosing a node in the backward direction (e.g., nodes F and G in Fig. 12.1) might lead to a packet loop. ~~In~~ Ref. [5], all nodes ~~are considered~~, but ~~the~~ routing stops at a node whose best choice is to return the packet to a neighbor that sent the packet to it (the loop-free property has been proved for this variant). A node where packet forwarding is stopped due to lack of a neighbor

[1]More precisely, in their original work MFR also considered nodes with negative progress. However, in later studies (e.g., ref. [2]) MFR was often described to consider nodes with nonnegative progress only. This chapter will use this variant when speaking of MFR.

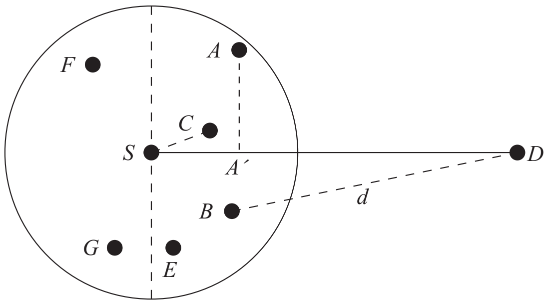

Figure 12.1 Node *A* is maximizing progress, node *B* has the least distance to *D*, and node *C* lies closest in direction to *D*.

in the forward direction, or by applying the described stoppage criterion, is termed a *concave node*.

12.2.2 Sensing Coverage and Greedy Routing

Many applications of sensor networks (e.g., tracking of moving targets) require a specific class of sensor networks that provide *sensing coverage*, that is, every point of a geographic area must be within the sensing range of at least one sensor node. A simplified formal model assumes that every node *N* has the same sensing range, which is a circle with radius R_s centered at *N*. Thus, for a sensor network covering an area *A*, the union of the circular sensing ranges of all network nodes must at least contain the area *A*. In a similar way, a simplified formal communication model can be defined by using a unique communication range R_c. The communication network, which is often referred to as the *unit-disk graph*, has a bidirectional communication link between any two sensor nodes *X* and *Y*, if and only if the Euclidean distance between *X* and *Y* is less than R_c.

Xing et al. [6] investigated properties of greedy routing algorithms in sensing covered networks having the *double-range property*, that is, $R_c/R_s \geq 2$. Focus on this special class is motivated by the geometric analysis from Wang et al. [7], which showed that a sensing covered network is always connected if it has the double-range property. The geometric analysis and simulation results from ref. [6] demonstrate that greedy geographic routing is a viable and effective routing scheme in sensing covered networks, and it turns out that the *range ratio* R_c/R_s has a significant impact on the quality of greedy routing in sensing covered networks.

The qualitative properties of greedy routing in sensing covered networks may be expressed in terms of *network* and *Euclidean dilation*. A subgraph *H* of *G* is termed a *network t-spanner* of *G* if the length (measured in hops) of the shortest path between any two nodes *U* and *V* in *H* is at most *t* times longer than the shortest path produced in *G*. The value *t* is termed the *network stretch factor* of the spanner *H*. Network dilation represents the stretch factor of a graph *G* relative to an ideal network producing a minimum number of hops of about $|SD|/R_c$ between source S and destination D. *Euclidean stretch factor* and *dilation* can be defined in a similar way by utilizing Euclidean distance instead of hop count.

Xing et al. [6] studied the dilation properties of sensing covered networks by utilizing a known upper bound of the Euclidean stretch factor of *Delaunay triangulations* (*DTs*). This well-studied graph structure can be defined as the twin of ~~the~~ *Voronoi diagrams*. For a set of n nodes in two-dimensional (2D) space, the Voronoi diagram partitions the plane into n *Voronoi regions* $Vor(U)$, while each Voronoi region contains all points in the plane that are closest to U (see the face surrounding node F in Fig. 12.2, for instance). The DT can be obtained by connecting each node pair (U, V) that shares a common boundary in the Voronoi diagram (see the dotted lines in Fig. 12.2).

It is proved in ref. [6] that the DT is a subgraph of a sensing covered network when the double-range property holds. This result and the known upper bound of the Euclidean stretch factor of DTs is finally used to derive a constant upper bound for the network dilation in a sensing covered network that has the double-range property. Additionally, it is observed that any DT edge is shorter than $2R_s$. Thus, when range ratio R_c/R_s increases, the shortest path found in the DT gets significantly longer than the shortest possible path in the complete network, since all edges longer than $2R_s$ are ignored by the DT.

Besides the results based on DT, properties of greedy routing are investigated in ref. [6] as well. It is proved that greedy routing will always find a routing path between any two nodes if the sensing covered network has a convex network boundary and the double-range property holds. Additionally, the progress made in each routing step is at least $R_c - 2R_s$ closer to the destination than the current forwarding node. Furthermore, the latter result is used in order to estimate the quality of the routing path produced by greedy routing. It is observed that in a sensing covered network with the double-range property the path (from source S to destination D) found by greedy routing is always no longer than about $|SD|/(R_c - 2R_s)$.

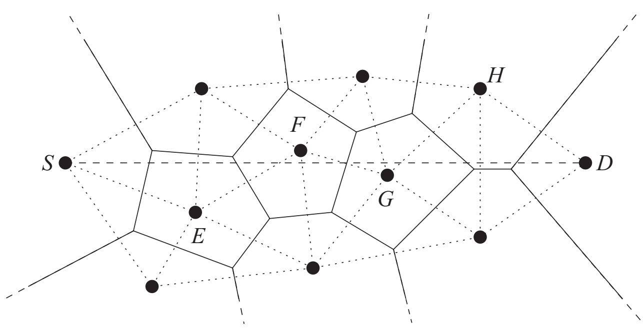

Figure 12.2 The bounded Voronoi greedy forwarding (BVGF) routing algorithm will select the node closest to D, but considers only nodes having a Voronoi region intersecting the straight line SD.

In summary, the result of DT and greedy routing motivate the *bounded Voronoi greedy forwarding* (*BVGF*) algorithm [6] as follows. Greedy routing applied in sensing covered networks will produce satisfactory path lengths if the range ratio R_c/R_s is significantly greater than 2, while, on the other hand, the upper bound of the path lengths produced $|SD|/(R_c - 2R_s)$ tends to infinity when the range ratio is close to 2. The shortest path found in a DT is always upper bound by a constant, while the upper bound becomes very conservative when the range ratio is increased. Thus, the BVGF algorithm is a combination of both methods, greedy routing, and routing along the edges of a DT. A node holding a packet addressed from the source node S to the final destination D will consider only those neighbor nodes U, where the line segment connecting S and D intersects $Vor(U)$ or coincides with one of the boundaries of $Vor(U)$. From these subsets of one-hop neighbors, the node closest to D will be selected as the next hop node. For instance, in Figure 12.2 only nodes E, F, G, and H can be visited by BVGF. Note that BVGF is not constrained to the edges of the DT. For instance, if source S is able to reach node F, it will send the message to node F directly, since node F is closer to D than E.

A theoretical analysis shows that BVGF will always find a path in sensing covered networks with the double-range property. An additional result shows that each node visited by the path produced by BVGF has a distance of at most R_s from the line connecting source node S and final destination D. Finally, this result is used to prove for the sensing covered network that the network dilation of the paths produced by BVGF is always upper bound by a constant value, provided that the range ratio is at least 2. In addition to this theoretical analysis, the average dilation of BVGF has been investigated in a simulation environment. The simulation results show that network dilations produced by BVGF are comparable to greedy forwarding, while Euclidean dilations are always better for all range ratios within 2 and 10.

12.2.3 Real-Time Communication in Sensor Networks

Sensor network applications such as surveillance systems may require sensor nodes to meet certain "soft" real-time communication constraints. Only a few results that adequately address such real-time requirements exist for sensor networks. The *SPEED* protocol by Lu et al. [8] is the first greedy-based protocol addressing real-time guarantees for sensor networks. SPEED utilizes the notion of *relay speed* in order to select one "best" next hop node in a greedy manner. Relay speed toward a next hop node A is calculated by dividing the advance in distance by the estimated send delay toward A. The single-hop delay toward a neighbor A can be estimated by continuously measuring the round-trip delay between current unicast data transmissions and the receipt of related acknowledgments. The estimated single-hop delay is calculated by means of an exponential weighted moving average over the previous average with the current single-hop delay. The latter is obtained by subtracting the receiver-side processing time from the round-trip delay experienced by the sender.

Before selecting the next hop node, a set of candidate nodes is calculated by selecting all nodes closer to the final destination than the current node and removing all nodes having a relay time smaller than a certain threshold s, which is a system-dependent parameter. The next hop node is selected according to a discrete exponential distribution, while the node with the fastest relay speed is selected with the highest probability. Selecting only nodes with a relay speed greater than a certain threshold assures that this routing scheme, if successful, will guarantee delivery of a packet within time d/s, with d being the distance between source and final destination. Furthermore, the randomized selection scheme provides traffic balance, and thus reduced congestion, since packets are dispersed into a large relay area. The authors of ref. [8] also propose neighborhood feedback-loop and back-pressure rerouting mechanisms.

Huang, Dai, and Wu [9] considered a quality of service (QoS) routing scheme, using progress instead of distance metric to advance toward the destination. The selected neighbor is one that maximizes the ratio of progress and delay in sending to a neighbor, where progress from node S when forwarding to neighbor A and with destination D can be measured as $SD \cdot SA$ (the dot product of vector SD and SA), and delay can be replaced by any other additive QoS metric. The authors also proposed several ticket-based multipath schemes to search for QoS paths. They also proposed a backward checking method that corresponds to the iterative improvement method described here for power and cost aware routing protocols.

12.3 GUARANTEED DELIVERY BASED ON MEMORIZATION

Stojmenović and Lin proposed neighbor flooding as a recovery mechanism at concave nodes, while every intermediate node handles received messages using the basic routing algorithm (named *f-GEDIR*, *f-MFR*, and *f-DIR*, for instance) [5]. Each concave node memorizes message IDs and rejects further copies of the same message (more precisely, neighbors learn about their concave status from the packet and do not select them as forwarding nodes). In original *f-GEDIR* or *f-MFR*, each neighbor of a concave node initiates a separate routing task toward destination D. Lin et al. proposed *component routing* [10], a more elaborate recovery strategy where concave nodes determine connected components in the subgraph of its neighbors and forward the message to only one "best" node in each component. The number of routing tasks initiated due to concave nodes is thus reduced significantly, since there are at most four connected components of neighbors of any concave node in the unit graph model [10].

The *geographical routing algorithm* (*GRA*) by Jain et al. [11] maintains a routing table that maps locations on next-hop forwarding nodes. A node receiving a message addressed to destination D, will look up its routing table and find the position p that is closest to the final destination D. The message will then be forwarded to the neighbor node that is assigned with position p. Initially, the routing table contains position information about neighbor nodes only, thus, operation of GRA is the same as greedy forwarding. Message forwarding is deferred and a route discovery is invoked if the routing table contains no position closer to the destination than the position of the current forwarding node *S* itself. The route discovery will find an acyclic path from the current node to the final destination and update the routing tables of all nodes lying on that path. A new entry is added to the routing table that maps the location of destination *D* on the next hop along the discovered path. Jain et al. propose *breadth-first search* (*BFS*) and *depth-first search* (*DFS*) as two possible route discovery mechanisms. BFS is the same as flooding, that is, a node receiving a discovery message appends its address on the path discovered so far and rebroadcasts the packet. Additional broadcast packets received subsequently are ignored. DFS yields only a single acyclic path from node *S* to destination *D*. Similar to BFS, a node receiving a route-discovery packet puts its address into the packet, but forwards it to a single neighbor *Y* that has not been visited so far and that minimizes the sum of the distance between *S* and *Y* and *Y* and *D*. If all neighbors have been visited, the current forwarding node removes its address from the path discovered so far and returns the packet to the node from which it was first received. Once the final destination receives a route-discovery packet, it is able to send an acknowledgment to the originator of the route discovery by utilizing the reverse of the path stored in the discovery packet. All nodes along that path will receive the destination acknowledgment and will update their routing table accordingly. On receipt of the destination acknowledgment the originator *S* will continue to forward the original message toward destination *D*.

Independently a localized DFS-based routing algorithm was proposed by Stojmenović et al. [12]. In contrast to the GRA algorithm, nodes do not store any routing-table entries and the list of visited nodes~~, which~~ is not stored in the DFS packet. In order to enable DFS in a distributed manner, each node remembers if it has already been visited by the DFS traversal. Additionally, each node memorizes the node from where the packet was received for the first time. Packet forwarding is performed by sorting all neighbor nodes with respect to their distance from the final destination *D* and selecting the node that is closest to *D*. Already-visited neighbor nodes have already transmitted a forward packet, therefore neighboring nodes can overhear it and can learn their status and do not select them for another forwarding. A returned message will be sent to the next choice in the sorted list of all next-hop nodes. If all neighbors already have been visited or have returned the packet, then the message will be returned to the node that sent the message ~~to *A*~~ for the first time. In addition to the basic algorithm, Stojmenovic et al. discussed a possible improvement with respect to QoS support. By utilizing information about its own physical location and periodically updated position information about all neighbor nodes, a node *A* can estimate the current speed and send direction of itself and its neighbor *B* and can thus estimate how long the link between *A* and *B* will remain stable. This link measure can be used in order to construct a path that provides a specific connection-time requirement. Each node visited by the search message will simply ignore all adjacent links that do not match this QoS requirement. In addition, a minimum bandwidth requirement and maximum delay may be considered as well during DFS traversal. In a simplified model, total delay is decomposed into the number of hops $\times$ propagation delay per hop (which is directly related to the bandwidth requirement per hop). In order to find a path with the required maximum propagation delay, DFS traversal is limited to a maximum path length and will consider only edges that have at least the minimum required bandwidth. A node will return the search message immediately if the maximum number of hops is exceeded or no outgoing edge matches the minimum bandwidth requirement. Nodes located along the path found will memorize the uplink and downlink edges of the path, which finally enables communication between source *S* and destination *D* within the established QoS requirements. The DFS-based QoS routing protocol can also be designed by using an advance (distance- or progress-based) per delay metric over links with sufficient bandwidth and connection times. The search for such a path proceeds until the destination is found and overall delay is acceptable.

12.4 MEMORYLESS GUARANTEED DELIVERY

Bose et al. described *FACE*, the first memoryless single-path recovery mechanism with guaranteed delivery [13] in a unit-disk graph model of communication (assuming ideal medium-access control (MAC) layer and connectivity). The FACE algorithm is an improvement of the routing algorithm due to Kranakis et al. [4], which guaranteed delivery in connected geometric planar graphs. A geometric planar graph partitions the plane into faces bounded by the polygons made up of the edges of the graph, and the nodes are described by geographic positions. A geometric graph is said to be planar if there is no intersection between any two edges of the graph (see the graph depicted in Fig. 12.3, for example).

The main idea of the FACE algorithm is to route a packet along the interiors of the faces intersected by the straight line connecting the source node *S* and destination *D* (see Fig. 12.3). Each face interior is traversed by applying the *right-hand rule*

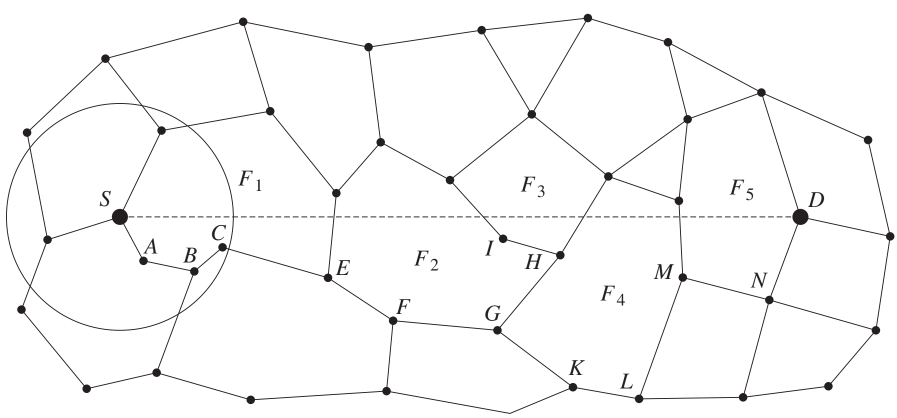

Figure 12.3 Face routing of a packet sent from source *S* to destination *D* leads to the path *SABCEFGHIHGKLMND* if the right-hand rule is applied.

or the *left-hand rule*, that is, a packet is forwarded along the next edge clockwise or counterclockwise from the edge where it arrived. When the packet arrives at an edge intersecting the line connecting S and D, the next face intersected by this line is handled in the same way. For example, in Figure 12.3 a packet routed from source S to destination D visits the faces $F_1, \ldots, F_5$. The algorithm proceeds until the destination node is eventually reached or if the first edge of current face traversal is traversed twice in the same direction. In the latter case, the destination node is not reachable. Face routing is shown to be loop-free and to guarantee delivery in static connected planar geometric graphs [13]. There are two main variants of FACE routing: the *before crossing* and *after crossing* protocols. They defer in the selection of the next edge after the current node detects that the face for traversing needs to be changed. The example in Figure 12.3 shows the before-crossing variant. Note that nodes cannot be certain locally whether they are following the right-hand or left-hand rule, because an open face has the opposite orientation to the closed faces, and nodes are not aware locally whether or not they are on the open face.

Ad hoc and sensor wireless networks can be modeled as unit-disk graphs, where two nodes communicate with each other if and only if the distance between them is at most R, where R is the transmission radius that is equal for all nodes. However, the unit-disk graph is not planar in general. Thus, before the FACE recovery procedure can be performed, a planar subgraph has to be extracted from the complete network graph. In the description of FACE, Bose et al. [13] proposed a distributed algorithm for extracting a planar subgraph from a unit-disk graph, which is based on *Gabriel graphs (GG)* [14], a well-known geometric planar graph construction. A GG for a finite-point set S is constructed by connecting any two nodes X and Y of S if and only if the circle with diameter (X, Y) contains no other node of S. This test can be performed by each node without any message exchange with neighbors, other than "hello" messages to learn their position. It is proved in ref. [13] that the minimal spanning tree belongs to the intersection of the GG and the unit-disk graph, therefore the network connectivity is preserved.

When the average density (average number of neighbors) increases, edges in GG become smaller, therefore the routes in FACE routing become longer. The other problem is that the routes may be long if an external face is encountered on the route. On the other hand, the path produced by successful greedy routing is comparable to the one produced by Dijkstra's single-source shortest path algorithm. Thus, Bose et al. [13] proposed a combination of the FACE algorithm with distance-based greedy routing, called *GFG* (*greedy-face-greedy*). A packet arriving at a concave node is switched into recovery mode and routed along the faces until reaching a node closer to the destination than the position of the concave node where recovery mode was entered. At this node, routing is again performed in greedy mode. The integration of GFG algorithm [13] with IEEE 802.11 was later implemented in the *greedy perimeter stateless routing* (*GPSR*) protocol by Karp and Kung [15]. Their GPSR protocol is the same as GFG (more precisely, they use the before-crossing instead of after-crossing variant, and also discuss the relative neighborhood graph (RNG) as an alternative to the GG, but these modifications do not improve the performance of the routing protocol).

12.4.1 Connected Dominating Sets and Shortcuts

Face routing has an increased hop count compared to Dijkstra's single-source shortest path algorithm, since planar graph construction based on GGs favors short edges over long ones. Datta et al. [16] improved the performance of GFG by the concept of *connected dominating sets* (*CDS*), *shortcut-based routing*, and a combination of both. Localized dominating-set construction in unit graphs is only possible with one-hop neighbor information, while shortcut-based routing also requires information about 2-hop neighbors.

A subset S of all network nodes G is called a *dominating set* if each node of G is either an element of S or has at least one neighbor in S. If the dominating set is connected, FACE routing constrained on CDS will produce shorter paths, since the corresponding GG edges will be longer on average. If a concave node is not in CDS, then it forwards the message to one of its adjacent nodes from CDS. Face routing (in recovery mode of GFG) then proceeds using only nodes from CDS. In greedy mode, the GFG algorithm works somewhat better on the whole set than on CDS, since there are more neighbor choices and longer edges can be used. The construction of CDS for unit-disk graphs is discussed in ~~a separate~~ chapter ~~on backbones~~ in this book.

In addition to the next forwarding node, there might be more neighbor nodes on the same path produced by FACE routing. For example, in Figure 12.3 the nodes A, B, and C on the path produced by traversal of face F_1 are all within transmission range of node S (the circle around S). When information about 2-hop neighbors is available, the concept of shortcut-based routing can be applied at each node. A forwarding node locally constructs the part of the planar graph seen by all its neighbors. Based on this information a node can make a shortcut by sending the message to the last known hop directly instead of forwarding it to the next hop along the path. For example, in Figure 12.3 node S could send the packet to node C directly.

12.4.2 Asymptotic Optimality of Face Routing

In order to analyze asymptotic behavior of combined greedy and face routing algorithms, Kuhn et al. [17] constructed a family of networks where each localized memoryless algorithm will produce a routing path that has quadratic cost compared to the cost of the shortest weighted path. The cost of a path is calculated by summing the cost produced by each path edge, while the theoretic results from ref. [17] are valid for all cost metrics that are polynomial in the Euclidean distance. Due to the lower-bound argument given in ref. [17], a localized memoryless algorithm producing at most quadratic path costs (compared to the shortest weighted path) for any network configuration in the worst case can be denoted as asymptotic optimal.

It can be observed [18] that asymptotic optimality is sacrificed if face traversal is switching back to greedy mode when the line connecting concave node S and final destination D is intersected for the first time (i.e., GFG and its previously described variants are not asymptotic optimal). On the other hand, the combination of greedy and face routing becomes asymptotic optimal when packets in face mode traverse the complete face and change back to greedy mode at the face edge that is closest to the destination *D*. However, successful greedy routing is more efficient than face routing in the average case, that is, even when not optimal in the worst case, switching back to greedy mode as soon as possible may be the better strategy in practice.

Kuhn et al. [19] described *greedy other adaptive face routing plus* (*GOAFR*+), a ~~greedy~~ routing algorithm that overcomes the trade-off between asymptotic optimality and average case efficiency of combined greedy and face routing. The efficient operation of face routing depends on the decision in the starting node of whether a face is being traversed in the clockwise or counterclockwise direction. For example, in Figure 12.4 applying the left-hand rule to traverse the outer face F_1 leads to the path *MLK* ... *UTSR* until arriving at the edge (R, Q) intersecting the line connecting source *S* and destination *D*. In contrast, if the face traversal was started in the opposite direction, the packet is forwarded along the significantly shorter path *MNOPQ* before switching to face F_2.

In order to cope with that suboptimality, Kuhn et al. proposed an extension of the GFG algorithm limiting the searchable area during face traversal. The GOAFR+ algorithm uses a circle *C* centered at the destination node *D* in order to restrict face traversal to the searchable area *C*. The radius of *C* is initially set to $\rho_0|SD|$ with $\rho_0 \geq 1$ so that source node *S* is also included within *C* (see the dashed circular arc centered at *D* in Fig. 12.4). The greedy mode is applied as long as there is a next-hop node closer to the destination *D*, and whenever possible the radius r_C of *C* is exponentially decreased ($r_C = r_C/\rho$ with $\rho > \rho_0$) as long as the currently visited node stays within *C*. Whenever the greedy mode encounters a local minimum at a node *U*, the algorithm continues with a modified version of face routing. When the face is traversed completely without hitting the current circle *C*, the packet will be sent to the node visited so far that is closer to *D* than *U* (and handled in greedy mode again). However, if no visited node is closer to *D* than *U*, the algorithm will terminate and report that no path from *S* to *D* exists. When *C* is hit for the first time, face traversal is reversed and face exploration is applied in the opposite

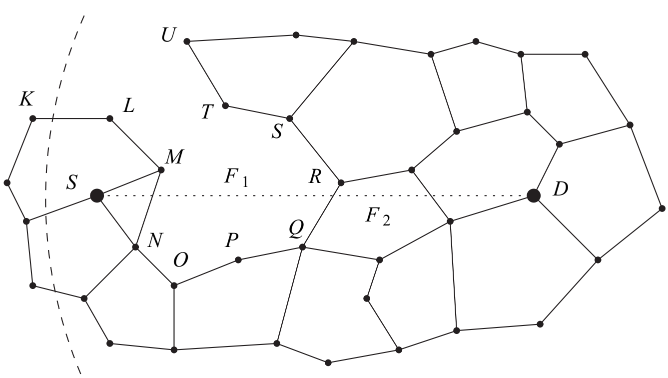

Figure 12.4 The GOAFR+ algorithm limits exploration to a circle centered at *D* and containing at least the node where the recovery procedure was invoked.

direction. If C is hit for the second time and none of the visited nodes is closer to D than U, face exploration is continued as if started at node U, but the radius of circle C is exponentially increased ($r_C = \rho r_C$). In order to avoid a complete face exploration, the algorithm applies an elaborate "early fallback" technique to return to greedy routing as soon as possible. However, it is proved in ref. [18] that algorithms will lose their asymptotic optimality when resuming greedy routing as soon as they arrive at the first node closer to the destination D than the concave node U. GOAFR+ maintains two counters to keep track of the number of nodes closer to and the number of nodes not closer to the destination than the starting node U of the current face traversal. If face exploration has visited up to a constant factor σ more nodes closer to D, GOAFR+ will interrupt face traversal, advance to the node seen so far that is closest to the destination D, and the packet will be handled in greedy mode again. Thus, GOAFR+ does not explore the complete face in general, but on the other hand, greedy routing is not resumed at the first node closer to destination D than concave node U. The latter property of GOAFR+ is finally used in ref. [19] in order to prove its asymptotic optimality. (From simulation results it turned out that $\rho_0 = 1.4$, $\rho = \sqrt{2}$, and $\sigma = 1/100$ are good choices for practical purposes.)

A simplified example of GOAFR+ is depicted in Figure 12.4, where source node S will forward in greedy mode to node M, which has no neighbor closer to destination D. Thus, the recovery strategy of GOAFR+ will begin exploration of face F_1 in the clockwise direction. At node L, face traversal hits the circle centered at node D, and the algorithm switches to exploration in the opposite direction. Each routing step updates the number p of nodes closer and the number q of nodes not closer to the destination D. When arriving at node P, a certain threshold condition $p > 1/3q$ holds ($p = 2$, $q = 3$, and $\sigma = 1/3$) and the message will be handled in greedy mode again.

For arbitrary unit-disk graphs (i.e., no restrictions regarding minimum node distance and maximum node degree), cost metrics divide into two classes, *linearly bounded* and *superlinear* cost functions. The first ones are lowerbound by a linear function, while for the latter there exists no such function. A theoretical result from ref. [19] reveals that for any localized memoryless routing algorithm A, there exists a node configuration where the cost of the path produced by A is unbounded with respect to the path produced by the shortest weighted-path algorithm if a superlinear cost function is considered. Thus, discussion of asymptotic optimality is reasonable only if restricted minimum node distance, maximum node degree, or linearly bounded cost metrics are considered. Standard cost metrics like hop count or Euclidean distance are linearly bound from below, while energy metrics defined as a polynomial d^α, with $\alpha > 1$ and d being the distance between sender and receiver, fall into the class of superlinear functions. However, as will be discussed in Section 12.5, "Routing with Energy-Aware Cost Metrics," from a practical point of view even energy-aware metrics are often considered to be of the form $d^\alpha + c$, with $c > 0$, and are thus linearly bound in practice.

In order to prove asymptotic optimality for linearly bound cost metrics and arbitrary unit-disk graphs, Kuhn et al. described an improved version of GOAFR+, which utilizes a *routing backbone* instead of using all possible edges from the unit-disk graph. Before routing takes place GOAFR+ precomputes a subgraph of the unit-disc graph that forms a connected dominating set of bounded degree. The distributed construction of such a graph structure is not described in ref. [19], but it is referred to in existing methods described in refs. [20–22]. Similar to GFG-I, the execution of GOAFR+ is restricted on the routing backbone, that is, a message is first sent to a dominating set member (if necessary), and from there on routing takes place along the routing backbone only until reaching a dominating set member that has the final destination D in its neighborhood.

12.4.3 Routing Along Geographical Clusters

Typical sensor network scenarios assume that sensor nodes are densely deployed in the monitored area. Greedy routing algorithms applied on uniformly distributed and densely deployed network nodes perform close to the shortest-path algorithm and are thus the first choice for such a scenario. However, even for densely deployed network nodes a recovery strategy may still be necessary, since greedy forwarding might get stuck at convex network boundaries or at network voids resulting from an inhomogeneous node distribution. Such an inhomogenity may be due to the physical properties of the monitored surface (e.g., a lake inside the monitored area where sensor nodes cannot be deployed).

Frey and Görgen [23] observed that such an inhomogeneous node distribution can have a significant impact on the recovery strategy being applied. Simulation experiments show that performance of face routing and the internal nodes concept may even degrade when node density is increased in a network scenario with an inhomogeneous node distribution. On the other hand, face recovery in combination with the shortcut procedure is almost unaffected by such an inhomogenity. However, the shortcut procedure has an increased message complexity compared to face routing or the internal nodes concept, since information about all 2-hop neighbors is required.

Frey and Görgen [23] described the *geographic cluster routing* (*GCR*) algorithm, which is based on the concept of GFG and assumes that the network is modeled as a unit-disk graph in the 2D Euclidean space as well. Routing in GCR is not performed on a per-node basis, but packets are forwarded along the edges of adjacent *geographical clusters*. In order to define geographical clusters, the plane is partitioned by an infinite mesh of regular hexagons (see Fig. 12.5), while each hexagon defines one cluster. Two geographical clusters C_1 and C_2 are denoted as adjacent, if there are at least two connected nodes with one located in C_1 and the other located in C_2. The graph resulting from adjacent clusters is not necessarily planar, thus, before face routing can be applied, a planar subgraph has to be extracted in advance. This is obtained by a variant of the localized planar graph construction applied by GFG. However, the method used by GCR may produce a disconnected subgraph even when the original graph is connected. Thus, in contrast with GFG and its variants, GCR cannot guarantee delivery, even if there is a path from source to destination. However, simulation results reveal that the delivery rate quickly tends to 100% when network degree is increased. In particular, in densely deployed networks

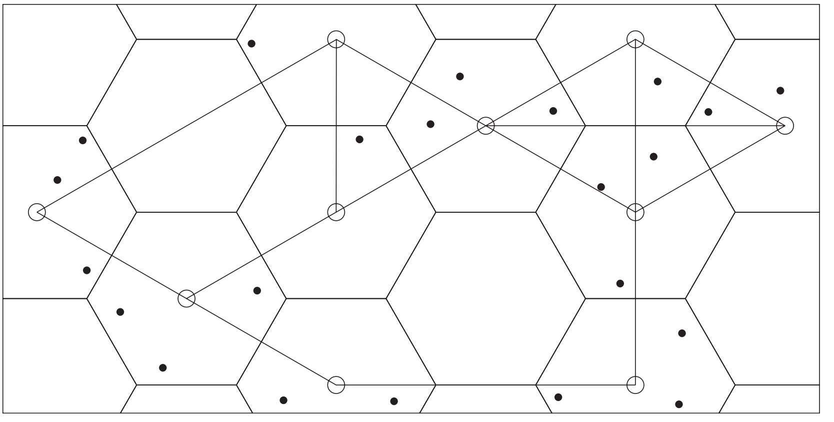

Figure 12.5 Geographic cluster routing explores the faces resulting from a planar graph extracted from the graph defined by all connected clusters.

GCR achieves a comparable performance to that of GFG combined with the shortcut procedure. However, message complexity is significantly decreased to an exchange of one-hop neighbor information only. Thus, GCR is a good choice to apply as a recovery strategy in densely deployed sensor networks.

12.4.4 Multicast Routing

A sensor network request may simultaneously address several different network nodes or network regions. This can be achieved by sending a unicast message to each individual entity. However, a resource-saving multicast strategy may be the better choice in order to reduce the bandwidth requirement when the same packet has to be delivered to multiple destinations. The majority of multicast protocols addressed to wireless networks require a distribution structure for the delivery of multicast messages. Mauve et al. [24] described a quasi-stateless protocol that achieves multicast addressing based on destination positions and neither requires construction and maintenance of a distribution structure nor resorts to some sort of flooding. The proposed *position-based multicast* (*PBM*) algorithm is a generalization of the GFG algorithm, with rules for splitting multicast greedy packets and a repair strategy for concave nodes that includes one or more addressed destinations.

Minimizing the path length for individual nodes and reducing the total number of message transmissions are two desirable and potentially conflicting properties of multicast forwarding strategies. The greedy routing part of PBM utilizes a localized criterion aimed at optimizing both objectives. In order to achieve short path lengths, greedy routing may select for each destination D the neighbor node that is closest to D. Applied as the sole optimization criterion, this strategy would lead to splitting message forwarding as soon as there is no single node that is optimal with respect to progress toward all destination nodes. Thus, while this criterion is a good choice to minimize the path length for individual nodes, the total number of message transmissions remains suboptimal. On the other hand, reducing bandwidth usage can be obtained by sending along a single path as long as possible, that is, the message will be duplicated only if no neighbor node exists that is closer to all the destination nodes considered. However, it can be observed that splitting a packet too late may again increase the total number of hops.

Mauve et al. derived an optimization criterion for greedy multicast forwarding that combines both objectives into one expression. The function described in ref. [24] depends on a parameter λ within [0, 1] that can be used to bias the expression between both extremes. A value close to 0 will result in splitting a message as soon as possible, while the total number of single-hop transmissions is likely to decrease when λ is increased up to a value $s < 1$. Simulation results show that there exists an optimal value for λ within $[0, 1]$ regarding the total network load produced.

Greedy multicast forwarding may arrive at nodes with no node closer to some of the addressed destination nodes. Thus, similar to unicast greedy routing, a recovery strategy must be employed in order to guarantee delivery to all destination nodes. Mauve et al. generalized the face routing algorithm to support message forwarding with multiple destinations. If a node has no neighbor with forward progress with regard to one or more destination nodes, face recovery will be invoked for all these destinations, while all other destinations are handled in a greedy mode further on. Face recovery is started by sending the recovery packet to the next edge in a counterclockwise direction to the line connecting the current node and a virtual position averaged over all affected destination nodes. A node receiving a recovery packet checks to see if it is closer to some of the destinations addressed by the packet. For all destinations where the receiving node is closer than the node where face routing was invoked, the packet will revert to the greedy mode again. For all remaining destinations face recovery is continued by transmitting to the next edge in a counterclockwise direction from the last edge at which the packet arrived.

Splitting a packet into a greedy and a recovery copy may lead to redundant message transmissions, since the greedy packet and the recovery packet may travel the same edges for some hops. In order to reduce the load due to such redundant transmissions, PBM combines the greedy and recovery packet in one transmission as long as possible, that is, the greedy packet will follow the path of the recovery packet as long the next hop node selected by face exploration also provides progress toward all destinations addressed by the greedy packet.

12.4.5 Routing Toward a Single Information Sink

All known memoryless routing strategies for arbitrary sender/receiver pairs resort to some variant of face traversal in order to provide guaranteed delivery in a connected network. However, for a typical sensor network scenario where each sensor node is aware of the location of a single information sink D, a quasi-stateless alternative (some nodes do need to memorize some information to facilitate routing) to face routing has recently been proposed that enables a reliable traffic flow from each

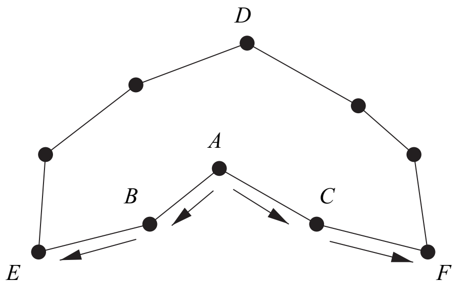

Figure 12.6 The algorithm PAGER establishes an acyclic graph leading toward one information sink.

sensor node toward *D*. The *partial-partition avoiding geographic routing* (*PAGER*) algorithm by Zou et al. [25] is a two-phase distributed and stateless construction of an acyclic graph leading toward the information sink *D*.

In the first phase the algorithm subsequently finds all *shadow nodes* where greedy routing toward the information sink would fail. Concave nodes are declared to be shadow nodes, and recursively other nodes are declared shadow nodes as well. A node becomes a shadow node when all its neighbors, closer to destination *D*, already became shadow nodes. For instance, in Figure 12.6, nodes *A*, *B*, and *C* are shadow nodes as far information sink *D* is concerned.

Greedy routing started at nonshadow nodes is always successful when shadow nodes are ignored. In order to enable successful traffic flow from all sensor nodes, shadow nodes establish exit pointers as follows. Each shadow node that has a non-shadow neighbor, or a neighbor with an already established pointer, will point to that neighbor. Packets originated in shadow nodes will follow the exit pointers until the first nonshadow node is reached. Routing then follows the greedy strategy until the final destination *D* is finally reached. An example is given in Figure 12.6. Nodes *B* and *C* will establish an exit pointer to nodes *E* and *F*, respectively. Afterwards, node *A* will establish an exit pointer to both nodes *B* and *C*. Traffic originated in *A* will follow exit pointers until reaching node *E* for instance. From there on, the packet will be delivered successfully in greedy mode.

12.5 ROUTING WITH ENERGY-AWARE COST METRICS

Sensor nodes are typically equipped with small low-power batteries, and it is impossible to recharge them in most sensor network scenarios. Thus, the lifetime of a sensor network is directly related to the energy consumption produced by the routing mechanism applied. If sensor nodes are able to adjust their signal strength, routing algorithms could attempt to reduce power consumption by selecting next-hop nodes within optimal transmission range. Geographic information can be incorporated in order to enable a localized computation of the best next-hop node by means of a power metric, which is a function that depends on the distance to the receiving node. However, a power-metric considers the optimal transmission range only, thus, single nodes might be selected by many routing tasks, which will result in their premature failure. Using a cost-metric or a combination of both power- and cost-metric might cause the nodes remaining battery power to increase the total network lifetime by spreading the energy consumption evenly among all network nodes. Such energy-aware metrics have been used to define novel energy-aware routing algorithms, replace traditional link-metrics in existing routing algorithms, and finally have also implicitly been applied to existing routing protocols by restricting the selection of next-hop forwarding nodes on an energy-optimized subgraph that has been constructed in advance.

12.5.1 Making Existing Protocols Energy Aware

The total energy needed for communication between two devices S and D might be reduced if the communication were relayed over an intermediate node R, while R and S transmit with the minimal power needed to reach nodes S and D, respectively. However, possible energy reduction depends on the position of the relay node R and the additional energy dissipation at the receiving device. This observation was used by Rodoplu and Meng [26] in order to define the *minimum energy communication network* (*MECN*) algorithm[2] that constructs power-optimized paths between a set of source nodes to one master node (i.e., the information sink in a sensor network scenario). It is implicitly assumed that each node is able to reach each other node in the network by transmitting with appropriate signal strength. In order to find all power-efficient routes to the master node, the algorithm first extracts a subnetwork (termed *enclosure graph*) containing at least all shortest-path edges (with respect to the power metric being optimized) leading from source nodes to the master node. This is achieved by a localized algorithm utilizing position information about all neighbor nodes and eliminating all nodes A for which it takes less power to send messages over a relay node instead of sending it directly to A. As a result each node obtains a reduced set of immediate neighbors, and thus in a second phase optimal routes can be constructed in a more power-efficient way, since communicating with neighbors in the enclosure graph requires less power than communicating with all neighbors from the original network. Optimal routes are found in ref. [26] by applying the distributed Bellman–Ford shortest-path algorithm. Each node calculates the minimum cost it can attain given the cost values of all its neighbors from the enclosure graph and the power needed to transmit a message to that neighbor. When the cost value of any neighbor is reduced, the current minimum cost value is recalculated, and if it was reduced, the new value is announced again to all immediate neighbors from the enclosure graph. The initial route setup from all sources to the sink can be obtained by broadcasting from the sink using only the edge of the enclosure graph, until all sources are reached.

[2]The algorithm was not termed MECN in the original work. However, this chapter will follow subsequent publications (e.g., ref. [27]), which referred this algorithm as MECN.

12.5.2 Localized Power- and Cost-Aware Routing

The majority of energy-aware geographic routing schemes described in the literature utilizes the distance to neighbors in the vicinity and apply some sort of distributed shortest weighted path algorithm to that information in order to construct a path from the source to the final destination. Stojmenovic and Lin [28] were the first investigating localized energy-aware greedy routing algorithms, that is, according to the greedy routing principle a received message will be forwarded to the best node regarding the energy metric being optimized.

According to refs. [26] and [29], a general power metric can, depending on node distances, be derived based to the most common channel model used for radio frequency systems. The received signal power for radio frequency communication decreases by a factor $1/d^{\alpha}$ (referred to as *path loss model*), with $\alpha \geq 2$ and d denoting the distance between the sending and receiving device. The correct choice of α depends on the system being used and can be determined from field measurements. A value of $\alpha = 2$ is often used to model radio propagation at short distances (referred as the free-space propagation model), while $\alpha = 4$ is used for radio transmission at longer distances (referred to as the two-ray ground reflection model). Additionally, the expression may be normalized by t, which denotes the predetection threshold at the receiver. Altogether this leads to an expression td^{α}, which denotes the minimum power the sender has to radiate in order to enable a signal detection at distance d. Besides power consumption at the sender there is additional power consumption at the receiver that is independent of the distance d and can thus be described as a constant $c > 0$. Summing the power expenditure for one signal transmission altogether amounts to $td^{\alpha} + c$. (*Note*: The constant c may also incorporate additional power expenditure due to computer processing and encoding/decoding on the sending and receiving devices.)

Assuming that additional nodes can be placed arbitrarily between source S and destination D, the polynomial power consumption $u(d) = td^{\alpha} + c$ in case of direct transmission can be converted to a linear function in d, producing minimal power consumption. More precisely, there is an optimal number $n = dc_1$ of equally spaced intermediate nodes producing a minimal total power consumption of $v(d) = dc_2$, where the constant values c_1 and c_2 are calculated from the constant power metric parameters t, c, and α [28]. In reality, it is not possible to insert equally spaced intermediate nodes. However, assuming that the power consumption for the rest of the path is equal to the optimal one, this result can be used to define the power-aware greedy routing algorithm *POWER*, where each intermediate node S selects the best next-hop neighbor E closer to the final destination D, which minimizes the sum $u(s) + v(t)$, with $s = |SA|$ and $t = |AD|$. For example, in Figure 12.7, node S will select node C as the next forwarding node, since the power $u(s)$ needed to transmit a message directly to C and the minimal power $v(t)$ needed to forward the packet over the remaining distance between C and D is minimal compared to all other neighbors.

The theoretical result of the power optimal number of equally spaced intermediate nodes is directly related to the polynomial power-consumption formula $td^{\alpha} + c$.

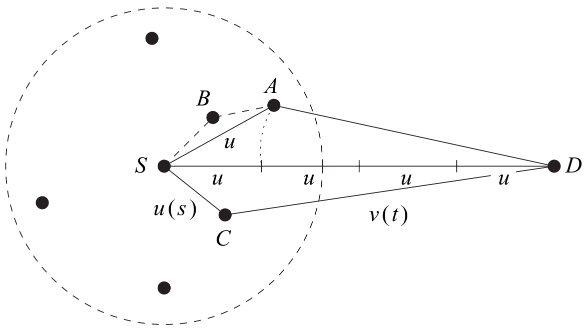

Figure 12.7 *S* selects nodes *C*, *A*, and *B* in the POWER, PowerProgress, and IPowerProgress methods, respectively.

If other power metrics were applied (e.g., a metric changing at a reference distance between free-space propagation and the two-ray ground model) a new theoretical analysis will be necessary in order to calculate these optimal values again before the POWER routing algorithm can be applied. In particular, if the power metric is given by empirical values only, an approximation by an appropriate choice of t and α may be necessary. Kuruvila et al. [30] proposed a novel power-aware greedy routing scheme, *PowerProgress*, which does not suffer from this fact. Let $u(s)$ be the power needed to transmit a message from node S to neighbor node A at distance s. If all subsequent forwarding nodes make the same progress toward the destination, the minimum number of forwarding steps amounts to $d/(d-t)$, with d being the distance between S and D and t being the remaining distance between A and D. When each forwarding step consumes the same amount $u(s)$ of power, the total power consumption will be at least $u(s)d/(d-t)$. Thus, a forwarding node applying the PowerProgress routing strategy will select the neighbor node A that minimizes $u(s)/(d-t)$ (with $t < d$), that is, the power spent per unit of progress made in terms of getting closer to the destination D (see Fig. 12.7).

Additionally, the *IPowerProgress* algorithm, which is an iterative refinement of the optimal node found by the PowerProgress method, was described by Kuruvila et al. In the first iteration step a node S applies the PowerProgress selection criterion in order to find the optimal next-hop node E regarding the distance between S and final destination D. However, sending the packet to node E may still be optimized locally, that is, it might still be more power efficient to send the packet over a relay node instead of sending it directly to E. Thus, the next iteration step selects (if possible) a neighbor F of both S and E, which has a distance to D less than the distance between S and D and which minimizes the sum R of power needed to send from S to F and finally from F to E. However, the relay node F is selected only if the power r needed to relay the packet is less than sending it directly to E. If such a node F is found, the original next hop node E is replaced by F and the iterative refinement method is applied again. The procedure repeats until no better node can be found and the packet is sent to the last optimal relay node found. Note that the node E, which was found initially, is not necessarily visited by the selected routing path. An example of the algorithm is depicted in Figure 12.7. First, node S will select A according to the PowerProgress method. However, there is an optimal relay node B producing less power consumption than would be spent by sending directly to node A. The algorithm will terminate at node B, since there is no additional node U that would further improve the power consumption when sending to B over relay node U.

Kuruvila et al. [30] also defined the *ProjProgress* and *IProjProgress* algorithms, which differ from PowerProgress and IPowerProgress in terms of measuring the progress made in each routing step. Instead of calculating distances, the progress made by the projection of neighbor node A onto the line SD connecting source node S with destination node D is considered. A node S, applying the ProjProgress, will forward a message to the neighbor node A, minimizing the expression $u(s)/(SD \cdot SA)$, where $SD \cdot SA$ denotes the dot product of vectors SD and SA (cf. difference between MFR and GREEDY). The IProjProgress method is very similar to IPowerProgress, but differs in the first iteration step, which selects the best node by applying the ProjProgress method instead.

Singh et al. [31] have observed that minimizing hop count, delay, or the power consumption of the paths produced by routing algorithms may be misguided in the long term. A longer path passing through nodes that have plenty of energy may be a better solution in terms of total network lifetime. In order to avoid energy-critical nodes and to maximize the number of successful routing tasks, a cost metric $f(A)$ expressing a node's reluctance to forward a packet is defined in ref. [31]. It is an expression proportional to the inverse of the node's remaining battery power, thus, a node's reluctance increases significantly when its battery power approaches 0. Stojmenovic and Lin [28] proposed a localized algorithm, *COST*, which is based on that cost metric. The cost to route a packet addressed to D via a neighboring node A is the sum of the cost $f(A)$ and the estimated cost produced to send along the remaining distance between node A and final destination D. The cost of the remaining path is assumed to be proportional to the number of hops between A and D, which in turn can be estimated as td/R, with d being the distance between A and D, R expressing a node's sending radius, and time t set to an appropriate value (empirical results showed $t = f(A)$ to be a good performing definition). A node holding a packet addressed to D will select a next-hop node A closer to the destination, which minimizes the expression $c(A) = f(A) + td/R$ (with $d = |AD|$). In their recent paper, Kuruvila et al. [30] also investigated the principle of proportional progress in combination with the cost metric defined in ref. [31] and defined the CostProgress routing scheme, which selects the forwarding neighbor closer to the destination, which minimizes $f(A)/(d - x)$. An iterative improvement like IPowerProgress cannot be defined for CostProgress, since the overall cost increases by adding intermediate nodes on a path.

Stojmenovic and Lin [28] also investigated combinations of power and cost metrics in one expression in order to achieve both objectives, reducing energy dissipation of the current message forwarding and increasing total network lifetime for many routing tasks. A multiplicative expression termed power/cost metric can be defined as $powercost(S,A) = f(A)u(r)$ (with $r = |AS|$ and $u(r) = r^\alpha + c$). Based on that metric, a forwarding node running the PowerCost routing algorithm will select the neighbor node A, minimizing the expression $powercost(S, A) + v(d)f'(A)$ with $r = |SA|$, $d = |AD|$, and $f'(A)$ being the average reluctance of A and its neighbors. A simpler algorithm is proposed in ref. [30], by selecting a neighbor that minimizes $powercost(S, A)/(|SD| - |AD|)$ (the algorithm is named *Power-Cost Progress*). The *ProjPowerCostProgress* proposed in ref. [30] applies the same metric, but a node S will forward a message to the neighboring node A closer to the destination, which is minimizing the expression $powercost(S, A)/(SD \cdot SA)$. Finally, similar iterative versions of to Power-Cost Progress and ProjPowerCostProgress protocols are also described in ref. [30].

12.5.3 Energy-Aware Guaranteed Delivery

The localized energy-aware greedy routing algorithms described so far do not guarantee delivery even if there is a path from source to destination. Stojmenovic and Datta [32] investigated a combination of face routing with power, cost, and power/cost greedy routing algorithms (called *PFP*, *CFC*, and *PcFPc*, respectively), which guarantee delivery in connected unit-disk graphs. More precisely, routing will start with a power-, cost-, or power/cost routing scheme, respectively. As with GFG, a message that encounters a concave node E also will be handled by the face routing mechanism until the final destination D is reached or a node having a neighbor that is closer to D than the distance between E and D is found. In the latter case, the message is sent to the best of these neighbors and is again handled by the corresponding PFP, CFC, or PcFPc routing method. The choice of such nodes enables it to be proved that the combined routing mechanism remains loop-free and guarantees delivery.

Energy savings of PFP, CFC, and PcFPc result from the energy efficiency of the greedy methods being applied when not in recovery mode. An additional performance gain can be achieved by providing energy awareness in recovery mode, too. Stojmenovic and Lin [32] investigated the impact of CDS construction on the energy-efficiency of PFP, CFC, and PcFPc. A static selection of CDS results in a shorter lifetime of nodes from CDS, which ultimately leads to a shorter lifetime of the whole network. Thus, with the same argument applied to cost-routing, a cost metric might be applied to the construction of the dominating set, taking the node's remaining battery power into consideration. This kind of energy-aware dominating-set construction has been proposed by Wu et al. [33]. Roughly, the algorithm is an extension of the basic distributed dominating-set construction from ref. [34], where the energy level of each node serves as the primary key when comparing two identifiers for a decision about including a CDS (the details are in Chapter 11 in this book).

An additional improvement has been achieved in ref. [32] by applying the shortcut procedure during the recovery mode of PFP, CFC, and PcFPc. In contrast to the original shortcut procedure, the forwarding node considers an energy metric instead of a hop-count metric. To apply this shortcut procedure, 2-hop neighbor information is required.

12.6 BEACONLESS ROUTING

The greedy forwarding mechanisms described need periodic hello messages (beaconing) transmitted with maximum signal strength by each node in order to provide current position information to all one-hop neighbors. This proactive component of greedy routing leads to additional energy consumption, which occurs independently of current data traffic.

Heissenbüttel and Braun [35] proposed the *beaconless routing* (*BLR*) algorithm. The *contention-based forwarding* (*CBF*) by Füssler et al. [36] and *implicit geographic forwarding* (*IGF*) by Blum et al. [37] implement the same idea, focusing on the integration of BLR with the IEEE 802.11 MAC layer. Since no beacons are transmitted, a node, currently holding the packet with the known destination, is generally not aware of any of its neighboring nodes and simply broadcasts a data packet. The main idea of BLR is that each neighboring node receiving the packet calculates a small transmission time-out before forwarding the packet, depending on its position relative to the last node and the destination. The node located at the "best" position introduces the fewest delays and retransmits the packet first. The remaining nodes then cancel the scheduled packet.

For example, in Figure 12.8, node *B* is closest to the destination *D*, but its transmission is not heard by node *F*, also closer to destination than *S*. To ensure that all potential forwarding nodes detect this transmission, only nodes within a certain forwarding area may be allowed as candidate nodes for the next forwarding step. The forwarding area has the property that each node is able to overhear the transmission of every other node within that area. Heissenbüttel and Braun show that the circle with a diameter equal to the transmission radius, centered at the line *SD* with *S* as one endpoint (the dotted circle in Fig. 12.8) is a good forwarding area with regard to progress and successful hops before greedy routing fails. Several delay functions are investigated, resulting in different forwarding behavior.

The authors of ref. [36] also propose a technique called the *active selection method*. A forwarding node sends a control packet instead of the full message to all its neighbors. Neighbors that provide forward progress respond after a time-out

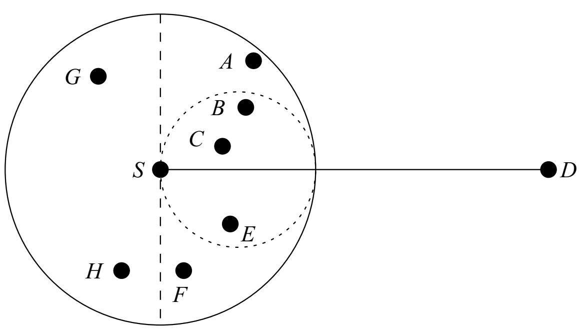

Figure 12.8 A possible forwarding region for BLR.

that depends on their distance to the destination. If a neighbor hears another neighbor's response, it does not respond itself (it is suppressed). The forwarding node then sends the full message, indicating which of its neighbors will forward the message. In a similar way, Zorzi [38] proposed to avoid duplicate forwarding in a BLR scheme by applying the request-to-send/clear-to-send (RTS/CTS) MAC scheme known from IEEE 802.11. The current node sends an RTS signal instead of the message. Afterwards, the node waits for a node to respond with a CTS signal. If several responses are received, the node selects the one that appears to be the best for forwarding and then sends the packet to that neighbor directly.

The principle of sending a control message before selecting the appropriate next-hop node can also be applied in order to describe a beaconless GFG (or alternative protocol, for example, beaconless GOAFR+) scheme [39]. If no CTS signal is received, the node assumes that no neighbor closer to the destination exists and sends another RTS packet to enter the recovery mode. The following procedure is repeated at each intermediate node *S* during the recovery phase of the beaconless GFG protocol. Each receiving neighbor of *S* sets a time-out based on the distance to *S*, so that closer neighbors have a smaller time-out, following the preference of the localized planar graph extraction method. All neighbors participate (including those closer to the destination) in competing for the forwarding neighbor for recovery mode. When the time-out at neighbor *A* expires, *A* makes a decision whether or not to report to *S*. If *A* heard a transmission from any node *B* such that *B* is located inside the circle with diameter *AS*, then *A* cancels reporting to *S*. Otherwise, *A* reports. Note that *A* reports even if it learns in the process that it will not be selected as forwarding neighbor, because its report may prevent other nodes, not in GG, from falsely reporting to *S*, which may contribute to the wrong choice of forwarding neighbor at *S*. After receiving all replies from GG neighbors, node *S* selects, among all neighbors from the GG, the one that creates the smallest angle in relation to the incoming packet direction, in the direction decided (clockwise or counterclockwise), following the GFG (or alternative, e.g., GOAFR+) scheme.

12.7 PHYSICAL-LAYER IMPACT ON ROUTING

Almost all existing literature on geographic routing uses the unit-graph communication model, which does not take into account random variations in received signal strengths. It was demonstrated by Schmitz et al. [40] that signal-strength fluctuations have a significant impact on ad hoc network performance metrics, sometimes "outperforming" the impact of node mobility. Thus, nondeterministic radio fluctuations cannot be ignored when designing robust ad hoc network protocols based on ad hoc network simulation and analysis. For instance, in order to find the positions of neighbors, nodes need to resort to a hello message exchange. This is a simple procedure in the unit-graph model, accomplished by each node sending one hello packet, which is then received by all neighbors located within transmission radius *R*. However, with a realistic physical layer, hello message operation requires a closer look [41].

Independent of the physical-layer model being used (e.g., the combined Friis and two-ray ground model used in ref. [45] or the log normal shadowing model used in ref. [41], protocols dealing with physical-layer impact require nodes to estimate the probability of receiving a bit or a packet based on either signal strength, distance between nodes, or merely by deriving statistics from a number of bits or packets recently sent between two nodes. The basic property of each of realistic physical modeling is a rapidly decreasing packet reception probability. For example, in the shadowing model used in ref. [41], the packet reception probability $p(x)$ depends on the probability of receiving a bit successfully, the length of the packet, and the distance x between two nodes. Suppose R can be determined in that way, so that the packet error rate at distance R is 0.5. Then the function $p(x)$ may have approximately the following values: $p(0) = 1$, $p(0.1R) \approx 1$, $p(0.5R) \approx 0.9$, $p(R) = 0.5$, $p(1.5R) \approx 0.25$, and $p(2R) = 0$. The given values are only an illustration, but give a sufficient intuition on how to design physical layer–aware routing schemes.

Kuruvila et al. [42,43] described geographic routing schemes that are amenable to any realistic physical-layer model (which follows the basic properties of the wireless medium) and consider two basic medium access-layer approaches, with fixed and variable packet lengths, while cases both with and without acknowledgments are being considered. The described methods assume that all nodes use the same transmission power for sending messages, and, in most cases, optimize the expected (packet or bit) hop count on a route. In the case of routing without acknowledgments, the goal is to find the route with the maximal probability of delivering a packet at the destination.

In order to apply position-based routing, the first step is to find a reasonably accurate approximation for the bit and packet reception probabilities for the given physical-layer model. In refs. [42] and [43], Kuruvila et al. considered the lognormal shadowing model and used the following function $P(q, x)$ as an approximation within 5% accuracy of the actual one. The functions $P(q, x) = 1 - (x/R)^{q\beta}/2$ for $x < R$ and $P(q, x) = (2 - x/R)^{q\beta}/2$ otherwise, where β is the power attenuation factor (between 2 and 6). The constant R is determined so that the value of the considered probability at distance R is $P(q, R) = 0.5$. The value q depends on the length of the considered entity. Bit reception probability is $b(x) = P(1, x)$, while, for instance, for packets 120 bits long the packet reception probability is $p(x) = P(2, x)$. The reason for using the approximation rather than the actual function is to reduce computation time at each node (if the protocol is used in practice) and in order to simplify the analyses and simulation of the protocol.

First, consider the case of routing with fixed-size packets and acknowledgments using the same packet size. If the acknowledgments are of a different packet size, the algorithms described are still applicable by changing only the corresponding formulas involving acknowledgments. Let C be the node currently holding the message, D be destination node, A the forwarding neighbor considered, $c = |CD|$, $a = |AD|$, and $x = |CA|$ (see Fig. 12.9). Several localized position-based algorithms are described in ref. [42]. The following discussion describes only the best performing ones, which also apply a general design principle. The progress made by

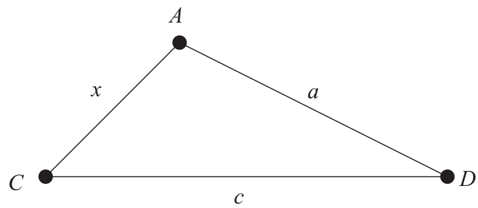

Figure 12.9 Several physical-layer optimized localized routing schemes can be defined by considering the probability of a successful transmission $p(x)$ and the progress $c - a$.

forwarding from C to A is $c - a$, and this progress is probabilistic. In the *aEPR* (expected progress routing) algorithm [42], the node C currently holding the packet will forward it to a neighbor A (closer to the destination than itself), which maximizes the expected progress, which is the product of the probability of successful delivery and acknowledgment of the packet from C to A (which is $p^2(x)$) and the progress made $(c - a)$ by forwarding to A. Thus in aEPR, the neighbor A that maximizes $p^2(x)(c - a)$ is selected.

The progress that can be made by sending a packet to A can also be considered with respect to the cost measure for making this progress. The cost measure considered is the expected hop count. The expected hop count depends on the distance and the selected number u of acknowledgments. The progress made could be measured in different ways. In the aEPR-1 algorithm [42], the node C currently holding the packet will forward it to a neighbor A (closer to the destination than itself), which maximizes the ratio of expected progress and the cost of the progress made. Since the considered cost, the expected hop count, is $1/p(x)^2 + 1/p(x)$, aEPR-1 will select the neighbor A, which maximizes $(c - a)/(1/p(x)^2 + 1/p(x))$. This derivation is based on a single acknowledgment for each packet, which is best only if packet reception probability is over 0.5. The optimal number of acknowledgment retransmissions u is approximated as $u \approx 1/p(x)$. The expected hop count is then $f(u, x) = 2/(p(x)(1 - (1 - p(x))^u))$. This variant, called *aEPR-u*, selects the neighbor that maximizes $(c - a)/f(u, x)$.

The *iterative EPR* (*IaEPR*) algorithm is an improved variant of aEPR-u. The algorithm can be described as follows. As in aEPR-u, the node C currently holding a message will first find a neighbor A that maximizes $(c - a)/f(u, x)$. Then, an intermediate common neighbor node B (closer to the destination than C, if it exists, $b = |BD|$) is found, which minimizes $f(u_1, |CB|) + f(u_2, |BA|)$, where $u_1 \approx 1/p(|CB|)$ and $u_2 \approx 1/p(BA|)$. If $f(u_1, |CB|) + f(u_2, |BA|) < f(u, x)$, then B becomes the new forwarding neighbor, taking the role of A. This process is iteratively repeated until no improvement is possible. Node C will forward the message to the selected neighbor A, which then again applies the same scheme for its own forwarding.

Consider now the model that does not have hop-by-hop acknowledgments. Localized protocols for this model are described in ref. [43]. It was proved in ref. [43] that the packet delivery rate approaches 1 if a large number of intermediate nodes is placed between the source and the destination nodes at distances between consecutive hops approaching 0. Following this observation, a localized algorithm can be described as in ref. [43]: The node C currently holding a message will forward it to its nearest neighbor A, which is closer to the destination than itself. The process continues until the destination is reached or a node is reached that has no neighbor closer to the destination.

A somewhat better performance is obtained by the following alternative scheme [43]. The progress made by forwarding from C to A is $c - a$. This progress is probabilistic. In the *nonacknowledged EPR* (*nEPR*) algorithm [43], the node C currently holding a message will forward it to a neighbor A (closer to the destination than itself), which maximizes the expected progress, which is the product of the probability of successful delivery $p(x)$ of the message from C to A and the progress $(c - a)$ made by forwarding to A. Therefore, the neighbor A that maximizes $p(x)(c - a)$ is chosen to forward the message.

The *iterative EPR* (*InEPR*) algorithm [43] is an improved variant of nEPR. The algorithm can be described as follows. As in nEPR, the node C currently holding a message will first find a neighbor A that maximizes $p(|CA|)(|CD| - |AD|)$. Then, if it exists, an intermediate node B (closer to the destination than C and a neighbor to both C and A) is found that satisfies $p(|CB|)p(|BA|) > p(|CA|)$ and has the maximum $p(|CB|)p(|BA|)$ measure. If found, then B becomes the new forwarding neighbor, taking the role of A. This process is iteratively repeated until no improvement is possible. Node C will forward the message to the selected neighbor A, which then again applies the same scheme for its own forwarding.

Now consider the case of variable packet lengths on each hop, and routing with hop-by-hop acknowledgments [44]. The localized algorithms described remain the same, with the following differences. Instead of the expected hop count in terms of packets, the schemes measure the expected number of transmitted bits. The expected hop count $f(u, x)$ in aEPR-u and IaEPR is replaced by the expected bit count $g(b, k)$ for routing with acknowledgments. If the aEPR variant is considered, then the-criterion maximizing $p^2(x)(c - a)$ is replaced by the criterion maximizing $g(b, k)$ $(c - a)$. Observe here that k, the packet length corresponding to the optimal expected bit count $g(b, k)$ (determined in ref. [45]), is not a constant, since each neighbor, being at a different distance, has its own optimal value for k. The case of variable packet length and routing without hop-by-hop acknowledgments was also considered in ref. [44].

The algorithms described so far are physical layer–based solutions for greedy position–based routing. Routing with guaranteed delivery for the unit-graph model and an ideal MAC layer, as described in ref. [13], applies greedy routing as long as possible, and when a node has no neighbor closer to the destination than itself, it resorts to face-recovery mode until a node closer to it is found. The recovery procedure is based on a planar graph locally defined. This procedure can be adapted to the physical layer in a straightforward manner. The edges of the planar graph are normally short ones, and therefore have relatively high reception probabilities. They are therefore good choices for edge selection. Thus, the recovery mode for the physical-layer impact routing may proceed in the same way as in the unit-graph model. Only greedy mode needs to be changed.

Finally, beaconless routing can be adapted to the physical layer by modifying the criterion for selecting the best forwarding neighbor and the appropriate time-out. The time-out can be based on the formulas already described here for selecting the best forwarding neighbor. If a given node announces the request for forwarding the packet several times, the best forwarding neighbors will receive it, and the best will respond a few times to make sure the response was received and it was selected.

EXERCISES

12.1 Show that any greedy routing algorithm that selects only nodes closer to the destination is always loop free. Find a representation of MFR in terms of the dot product and show in a similar way that MFR is also a loop-free routing scheme. Finally, construct an example where DIR will end in a packet loop [5].

12.2 Suppose a node configuration $S = (0, 0)$, $A = (1, 1)$, $B = (1, -1)$, and $D = (3, 0)$ ((x, y) represents the node position), while nodes S, A, B can mutually reach each other and node D is disconnected from all other nodes. Show that MFR forwarding from node S to D will end in a loop and that such a loop can also be constructed even when there is a path from source S to destination D. Explain why this does not contradict the proof of the loop-free property of MFR [5] and show how this loop can easily be repaired in a practical implementation.

12.3 The GEDIR [5] algorithm is an improvement of GREEDY that considers all (i.e., even those in the reverse direction) neighbor nodes and selects the node closest to the final destination. A message is dropped if it would be sent back to the node where it was previously sent from. Show that GEDIR is a loop-free routing algorithm and construct an example where GEDIR is successful while GREEDY is not.

12.4 The *nearest with forward progress* (*NFP*) algorithm [2] is a progress-based routing strategy which selects the neighbor with least forward progress as the next hop node. Investigate whether the method provides loop-free operation or if there is a node configuration where this routing strategy will produce a packet loop.

12.5 Derive an expression that estimates the expected forward progress of MFR applied on uniformly spatial distributed network nodes [46]. Use this expression to estimate a lower bound on the average number of hops produced by MFR when routing a packet over a distance d.

12.6 Greedy routing is often used as a single-path strategy, that is, at any time there is only one instance of the message in the network. In contrast, localized multipath strategies perform routing along a few recognizable paths simultaneously. Extend the GREEDY method presented to a multipath strategy that forwards the packet along c recognizable paths (however, paths can have edges in common). Define the different rules for nodes receiving the same greedy packet more than once [47].

12.7 Restricted directional flooding allows each intermediate node to forward a packet to possibly more than one neighbor lying in the general direction of the final destination [48,49]. Suppose the destination node D is located somewhere inside a circle C centered at the last known position of D. Apply the concept of restricted directional flooding and generalize distance-, progress-, and direction-based greedy routing methods in order to address all possible destination locations within the circle C [50].

12.8 Show that in a sensing covered network with the double-range property any node U lying within the Voronoi region $Vor(V)$ is also covered by node V. Use this result in order to prove that the Delaunay triangulation is always a subgraph of a sensing covered network with double-range property.

12.9 Show that the combination of GREEDY with the recovery strategy FACE is a loop-free routing strategy, while GEDIR and FACE may produce a packet loop.

12.10 Construct an example to show that face routing may not be successful in a connected unit-disk graph if it is applied in a subgraph that is not planar.

12.11 Suppose a weakened planar graph construction where the edges do not intersect in one intersection point but where collinear edges may intersect. Is face traversal always successful in such a graph construction?

12.12 Assume the following simple implementation of face routing. Face traversal is always performed in the clockwise direction. When face exploration encounters an edge intersected by the straight line connecting the last intersection point and final destination D, the next face is determined by simply skipping the intersected edge and continuing face exploration with the next edge clockwise from the intersected edge. Give an example where this simple implementation of face routing will lead to a packet loop. What additional condition must be checked in order to provide a loop-free operation of this algorithm?

12.13 It can be observed that face routing can produce a forwarding loop when the network topology changes due to node mobility. Show that both adding a new edge and removing an edge during face traversal may lead to a packet loop. Find a solution utilizing the creation time of both message and edges that will guarantee loop-free operation in the case where new edges are added during face traversal.

12.14 Assume a GFG implementation with the following simplified recovery strategy. The current face, which is traversed due to a packet recovery started at a node A, will never be changed. The recovery strategy will fall back to greedy mode when arriving at a node B, which lies closer to the destination D than the distance between A and D. Construct an example planar graph where this algorithm will end in a loop.

12.15 Construct a family of unit-disk graphs in order to show that any memoryless geographic routing algorithm with guaranteed delivery can produce a path of length $O(c^2)$, where c is the length of the shortest path [17].

12.16 Addressing all nodes lying within a certain geographical area is termed geocasting. Define an algorithm based on planar graph traversal that achieves geocasting with guaranteed delivery when the area is a circle centered around a given center position [51].

12.17 The concept of *Gabriel graphs* [14] can be used to define a localized planar graph construction for unit-disk graphs. By using information about all neighbor nodes, a node U preserves an edge to its neighbor V if and only if the circle with diameter $|UV|$ passing the nodes U and V does not contain any other neighbor. Show that for a connected unit-disk graph the resulting subgraph is planar and connected [13].

12.18 The localized planar graph construction from the previous exercise will not work correctly if the unit-graph property is missing. Give an example where two neighbor nodes U and V will produce an inconsistent view, that is, node U will preserve edge UV, while node V will remove that edge.

12.19 Suppose a generalization of the unit-graph concept where any node can have a sending range that varies within a maximum r and minimum $r/\sqrt{2}$ transmission range [52,53]. An edge exists between two nodes if they are mutually included in their sending ranges (i.e., only bidirectional connections are considered). Show that by additional message exchange a localized planar graph construction is also possible for this generalized network class. Investigate whether localized planar graph construction is also possible for variations in transmission range ratios that are larger than $\sqrt{2}$.

12.20 Face traversal needs exact location information about neighbor nodes and the final destination. Construct two examples where planar graph routing will end in a routing failure due to imprecise location information about the destination and the neighbor nodes, respectively [54].

12.21 Design and analyze a routing algorithm that will consider two types of errors: transmission failures (the receiver node is believed to be within the transmission radius, but it is not), and backward progress (the receiver node is believed to be closer to destination node than the sender node, but it is not) [55].

12.22 Planar graph routing applied on the network defined by connected geographical clusters may suffer from the fact that there are connected node configurations where any extracted planar graph will be disconnected. Give an example node configuration that proves this claim [23].

12.23 Assume a simplified power metric d^α with $\alpha \geq 2$ and d is the distance between the sender and the receiver. Show that it is always better to relay traffic along an intermediate collinear node. Does this proposition also hold if the power metric is extended by an additive constant $c > 0$?

REFERENCES

1. Hideaki Takagi and Leonard Kleinrock. Optimal transmission ranges for randomly distributed packet radio terminals. *IEEE Transactions on Communications*, **32**(3):246–257, March 1984.
2. Ting-Chao Hou and Victor O. K. Li. Transmission range control in multihop packet radio networks. *IEEE Transactions on Communications*, **34**(1):38–44, January 1986.
3. Gregory G. Finn. *Routing and Addressing Problems in Large Metropolitan-Scale Internetworks*. Technical Report ISI/RR-87-180, Information Sciences Institute (ISI), March 1987.
4. Evangelos Kranakis, Harvinder Singh, and Jorge Urrutia. Compass routing on geometric networks. In *Proceedings of the 11th Canadian Conference on Computational Geometry (CCCG '99)*, pages 51–54, Vancouver, Canada, August 1999.
5. Ivan Stojmenovic and Xu Lin. Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, **12**(10):1023–1032, October 2001.
6. Guoliang Xing, Chenyang Lu, Robert Pless, and Qingfeng Huang. On greedy geographic routing algorithms in sensing-covered networks. In *Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '04)*, pages 31–42, Tokyo, Japan, May 2004.
7. Xiaorui Wang, Guoliang Xing, Yuanfang Zhang, Chenyang Lu, Robert Pless, and Christopher Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (Sensys '03)*, Los Angeles, California, November 2003.
8. Chenyang Lu, John A. Stankovic, Tarek Abdelzaher, and Tian He. SPEED: A stateless protocol for real-time communication in sensor networks. In *Proccedings of the 23rd International Conference on Distributed Computing Systems (ICDCS 2003)*, Providence, Rhode Island, May 2003.
9. C. Huang, F. Dai, and J. Wu. On-demand location-aided QoS routing in ad hoc networks. In *Proceedings of the 2004 International Conference on Parallel Processing (ICPP)*, pages 502–509, Montreal, Canada, 2004.
10. Xu Lin, Mouhsine Lakshdisi, and Ivan Stojmenovic. Location based localized alternate, disjoint, multi-path and component routing schemes for wireless networks. In *Proceedings of the 2001 ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2001)*, pages 287–290, Long Beach, California, October 2001.
11. Rahil Jain, Anuj Puri, and Raja Sengupta. Geographical routing using partial information for wireless ad hoc networks. *IEEE Personal Communication*, pages 48–57, February 2001.
12. Ivan Stojmenovic, Mark Russell, and Bosko Vukojevic. Depth first search and location based localized routing and QoS routing in wireless networks. *Computers and Informatics*, **21**(2):149–165, 2002.

13. Prosenjit Bose, Pat Morin, Ivan Stojmenovic, and Jorge Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. In *Proceedings of the 3rd ACM International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM '99)*, pages 48–55, Seattle, Washington, August 1999.

14. K. R. Gabriel and R. R. Sokal. A new statistical approach to geographic variation analysis. *Systematic Zoology*, **18**:259–278, 1969.

15. Brad Karp and H. T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th ACM/IEEE Annual International Conference on Mobile Computing and Networking (MobiCom-00)*, pages 243–254, New York, August 2000.

16. Susanta Datta, Ivan Stojmenovic, and Jie Wu. Internal node and shortcut based routing with guaranteed delivery in wireless networks. In *Proceedings of the IEEE International Conference on Distributed Computing and Systems (Wireless Networks and Mobile Computing Workshop [WNMC])*, pages 461–466, Phoenix, Arizona, April 2001.

17. Fabian Kuhn, Roger Wattenhofer, and Aaron Zollinger. Asymptotically optimal geometric mobile ad-hoc routing. In *Proceedings of the 6th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM-02)*, pages 24–33, New York, September 2002.

18. Fabian Kuhn, Roger Wattenhofer, and Aaron Zollinger. Worst-case optimal and average-case efficient geometric ad-hoc routing. In *Proceedings of the 4th ACM International Symposium on Mobile Computing and Networking (MobiHoc 2003)*, pages 267–278, Annapolis, Maryland, 2003.

19. Fabian Kuhn, Roger Wattenhofer, Yan Zhang, and Aaron Zollinger. Geometric ad-hoc routing: Of theory and practice. In *Proceedings of the 22nd ACM International Symposium on the Principles of Distributed Computing (PODC)*, pages 63–72, Boston, Massachusetts, July 2003.

20. Khaled M. Alzoubi, Peng-Jun Wan, and Ophir Frieder. Message-optimal connected dominating sets in mobile ad hoc networks. In *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 157–164, Lausanne, Switzerland, 2002.

21. Jie Gao, Leonidas J. Guibas, John Hershberger, Li Zhang, and An Zhu. Discrete mobile centers. In *Proceedings of the 17th Annual Symposium on Computational Geometry (SCG)*, pages 188–196, 2001.

22. Yu Wang and Xiang-Yang Li. Geometric spanners for wireless ad hoc networks. In *Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS '02)*, pages 171–180, July 2002.

23. Hannes Frey and Daniel Görgen. Planar graph routing on geographical clusters. *Ad hoc Networks*, ~~2004~~.

24. Martin Mauve, Holger Füßler, Jörg Widmer, and Thomas Lang. *Position-Based Multicast Routing for Mobile Ad-Hoc Networks*. Technical Report TR-03-004, Department of Computer Science, University of Mannheim, Germany, 2003.

25. Le Zou, Mi Lu, and Zixiang Xiong. Pager: A distributed algorithm for the dead-end problem of location-based routing in sensor networks. In *Proceedings of the 13th International Conference on Computer Communications and Networks (ICCCN '04)*, pages 509–514, Chicago, Illinois, October 2004.

26. Volkan Rodoplu and Teresa H. Meng. Minimum energy mobile wireless networks. *IEEE Journal on Selected Areas in Communications*, **17**(8):1333–1344, August, 1999.

27. Li Li and Joseph Y. Halpern. Minimum-energy mobile wireless networks revisited. In *Proceedings of the 2001 IEEE International Conference on Communications (ICC 2001)*, Volume 1, pages 278–283, June 2001.

28. Ivan Stojmenovic and Xu Lin. Power-aware localized routing in wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, **12**(11):1122–1133, November 2001.

29. Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS-33)*, Volume 8, page 8020, Maui, Hawaii, January 2000.

30. Johnson Kuruvila, Amiya Nayak, and Ivan Stojmenovic. Progress based localized power and cost aware routing algorithms for ad hoc networks. In *Proceedings of the 3rd International Conference on AD-HOC Networks & Wireless (ADHOC-NOW '04)*, pages 294–299, Vancouver, British Columbia, July 2004.

31. Suresh Singh, Mike Woo, and C. S. Raghavendra. Power-aware routing in mobile ad hoc networks. In *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom-98)*, pages 181–190, New York, October 1998.

32. Ivan Stojmenovic and Susanta Datta. Power and cost aware localized routing with guaranteed delivery in unit graph based ad hoc networks. *Wireless Communications and Mobile Computing*, **4**:175–188, 2004.

33. Jie Wu, Fei Dai, Ming Gao, and Ivan Stojmenovic. On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks. *Journal of Communications and Networks*, **4**(1), March 2002.

34. Jie Wu and Hailan Li. On calculating connected dominating set for efficient routing in ad hoc wireless networks. In *Proceedings of the 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM '99)*, pages 7–14, Seattle, Washington, August 1999.

35. Marc Heissenbüttel and Torsten Braun. BLR: Beacon-less routing algorithm for mobile ad-hoc networks. *Computer Communications (Elsevier)*, **27**(11):1076–1086, July 2004.

36. Holger Füßler, Jörg Widmer, Michael Käsemann, Martin Mauve, and Hannes Hartenstein. Contention-based forwarding for mobile ad-hoc networks. *Ad Hoc Networks*, **1**(4):351–369, November 2003.

37. Brian M. Blum, Tian He, Sang Son, and John A. Stankovic. *IGF: A State-Free Robust Communication Protocol for Wireless Sensor Networks*. Technical Report CS-2003-11, Department of Computer Science, University of Virginia, April 21, 2003.

38. M. Zorzi. A new contention-based MAC protocol for geographic forwarding in ad hoc and sensor networks. In *Proceedings of the IEEE International Conference on Communications (ICC 2004)*, Volume 16, pages 3481–3485, Paris, 2004.

39. Ivan Stojmenovic. Beaconless Position Based Power Aware Routing and Routing with Guaranteed Delivery. In preparation.

40. R. Schmitz, M. Torrent-Moreno, H. Hartenstein, and W. Effelsberg. The impact of wireless radio fluctuations on ad hoc network performance. In *Proceedings of the 4th International IEEE Workshop on Wireless Local Networks (WLN 2004)*, Tampa, Florida, November 2004.

41. I. Stojmenovic, A. Nayak, J. Kuruvila, F. Ovalle-Martinez, and E. Villanueva-Pena. Physical layer impact on the design and performance of routing and broadcasting

protocols in ad hoc and sensor networks. *Computer Communications* (Elsevier), ~~forthcoming~~

42. Johnson Kuruvila, Amiya Nayak, and Ivan Stojmenovic. Hop count optimal position based packet routing algorithms for ad hoc wireless networks with a realistic physical layer. In *Proceedings of the 1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, Fort Lauderdale, Florida, October 2004.
43. Johnson Kuruvila, Amiya Nayak, and Ivan Stojmenovic. Greedy localized routing for maximizing probability of delivery in wireless ad hoc networks with a realistic physical layer. In *CD Proceedings of the 1st International Workshop on AlgorithmS for Wireless And mobile Networks (A-SWAN), Personal, Sensor, Ad-hoc, and Cellular Workshop* (at MobiQuitous), Boston, Massachusetts, August 2004.
44. Ivan Stojmenovic, Amiya Nayak, and Johnson Kuruvila. Design guidelines for routing protocols in ad hoc and sensor networks with a realistic physical layer. *IEEE Communications Magazine* (Ad Hoc and Sensor Networks Series), **43**(3):101–106, March 2005.
45. T. Nadeem and A. Agrawala. IEEE 802.11 fragmentation-aware energy-efficient ad-hoc routing protocols. In *Proceedings of the 1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, pages 90–103, Fort Lauderdale, Florida, October 2004.
46. Pedro Acevedo Contla and Milos Stojmenovic. Estimating hop counts in position based routing schemes for ad hoc networks. *Telecommunication Systems*, **22**:109–118, 2003.
47. Xu Lin and Ivan Stojmenovic. Location-based localized alternate, disjoint and multipath routing algorithms for wireless networks. *Journal of Parallel and Distributed Computing*, **63**:22–32, 2003.
48. Stefano Basagni, Imrich Chlamtac, Violet R. Syrotiuk, and Barry A. Woodward. A distance routing effect algorithm for mobility (DREAM). In *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom-98)*, pages 76–84, Dallas, Texas, October 1998.
49. Young-Bae Ko and Nitin H. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. In *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom-98)*, pages 66–75, Dallas, Texas, October 1998.
50. Ivan Stojmenovic, Anand Prakash Ruhil, and D. K. Lobiyal. Voronoi diagram and convex hull based geocasting and routing in wireless networks. In *Proceedings of the 8th IEEE Symposium on Computers and Communications (ISCC)*, pages 51–56, Kemer-Antalya, Turkey, July 2003.
51. Ivan Stojmenovic. Geocasting with Guaranteed Delivery in Sensor Networks. Paper presented at the International Workshop on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless and Peer-to-Peer Networks, Fort Lauderdale, Florida, February 2004.
52. Lali Barriere, Pierre Fraigniaud, Lata Narajanan, and Jaroslav Opatrny. Robust position-based routing in wireless ad hoc networks with unstable transmission ranges. In *Proceedings of the 5th ACM International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM '01)*, pages 19–27, Rome, Italy, July 2001.

53. Fabian Kuhn, Roger Wattenhofer, and Aaron Zollinger. Ad-hoc networks beyond unit disk graphs. In *Proceedings of the 2003 Joint Workshop of Foundations of Mobile Computing (DIALM-POMC)*, pages 69–78, San Diego, California, September 2003.

54. Karim Seada, Ahmed Helmy, and Ramesh Govindan. *On the Effect of Localization Errors on Geographic Face Routing in Sensor Networks*. Technical Report 03-797, University of Southern California USC, 2003.

55. S. Kwon and N. B. Shroff. Geographic Routing in the Presence of Location Errors. Submitted for publication.