Fast Progress-Based Routing
in Sensing-Covered Networks

Tarek El Salti
Department of Computing and Information Science
University of Guelph, Guelph, Ontario, Canada, N1G 2W1
Email: telsalti@uoguelph.ca

Thomas Fevens and Alaa E. Abdallah
Department of Computer Science and Software Engineering
Concordia University, Montréal, QC, Canada, H3G 1M8
Email: {fevens, ae_abdal}@cse.concordia.ca

***Abstract*—A sensing-covered network is wireless sensor network where every single point in the geographic area is covered by the sensing range of at least one sensor node. For routing on such a 2-D sensing-covered sensor network topology, we propose new greedy forwarding-type progress-based routing protocols, and hybrid routing algorithms which are based on a faster simplified version of the *BVGF* routing protocol of Xing *et al.* (2006) and the new progress-based routing protocols. All these algorithms guarantee delivery on sensing-covered networks. We demonstrate through simulations that our proposed routing protocols perform fewer calculations while maintaining comparable average path lengths, compared to some existing routing protocols.**

I. INTRODUCTION

A wireless *sensor* network integrates systems for processing, sensing and wireless communication. A sensor network is a system of wireless autonomous hosts that can communicate with each other over wireless links without having any fixed infrastructure or centralized control. Each host in this network can communicate with all other hosts within its transmission range [1], [2], which we will assume to be a fixed range R_c for all hosts. There will be hosts that are not within the transmission range of each other. Since the hosts are considered as endsystems (receive, send and process data) and as routers (data forwarding), multihop routing will be used where intermediate hosts are used to transmit the message. The absence of an infrastructure, together with possible dynamic topology changes and resource constraints, make routing efciently in such sensor network becomes a challenging task.

In addition, since sensor nodes typically have limited amount of memory, *local* routing algorithms are preferred whereupon a packet is forwarded based only on information gathered from the neighbors of the current node at most a fixed number of hops away (assumed to be one hop for this paper). We are specifically interested in *position-based* routing (also known as online routing [3], [4] or geographic routing [5]). In position-based routing protocols, a node forwards packets based on the location of the node, its neighbors up to a fixed number of hops way, and the destination [6]. These routing protocols use the fact that frequently sensor nodes lie in a plane (or to only a limited extent in the third dimension) and know their coordinates relative to a global coordinate system.

The sensing applications for sensor networks may need additional requirements to be included. For example, many applications (e.g., distributed detection [7]) present a sensing coverage requirement that is not included in the usual ad hoc networks. A *sensing-covered* network is a sensor network where every point in the geographic area is covered by the sensing range of at least one sensor node. The sensor nodes are used to monitor conditions, e.g., environmental conditions (temperature, pollutants, etc.).

In this paper, we present several guaranteed-delivery local position-based routing algorithms for sensing-covered networks. Two of these protocols are hybrid routing algorithms incorporating a simplified, faster alternative to *BVGF*. For the rest of the paper, we will assume that the nodes of the wireless sensor network exist in the Euclidean plane $\mathbf{R}^2$ and that the nodes know their coordinates in this space, acquired by a GPS system or some other technique. We also make the simplifying assumption that the nodes are static (not mobile).

II. BACKGROUND

A common model for a position-aware network is a geometric graph, which is a graph embedded in a d-dimensional Euclidean space such that its vertices are points with coordinates and its edges are straight-line segments. The set of n wireless hosts is represented as a point set S in $\mathbf{R}^2$, each point possessing a geometric location. For node u, we denote the set of its neighbors by $N(u)$. The number of the neighbors of u is the degree of u. On S, a (Euclidean) graph can be modeled as a weighted (undirected or directed) graph $G(S, E)$ where E is a subset of the pairs of nodes of S and the weight of an edge uv between nodes u and v is the Euclidean distance in the plane between the nodes which we denote as $|uv|$. The weight of a graph is the sum of its edge weights. Between any pair of nodes, there are two types of path lengths. One refers to the Euclidean length which is the sum of all the hops' Euclidean distances in a path. The other one refers to the network length which is the hop count of a path. Define a subgraph of G, $P(G)$, as a t-spanner of G if the length of the shortest path between any two nodes in $P(G)$ is not more than t times longer than the shortest path connecting them in G, where t is the stretch factor.

In our wireless host model, two nodes are connected by an undirected edge if the Euclidean distance between them is at most R_c, the transmission range of the nodes. The resulting graph is called a unit disk graph (*UDG*). A *UDG* is a common geometric graph to represent sensor networks and ad hoc networks. The *UDG* is considered poor for some routing protocols for various reasons. The 2-D *UDG* graph is typically a non-planar graph which means that it has crossing edges. Also since the *UDG* may be a dense graph, it can have a high average node degree, the routing protocols performed on it will take considerable time. One frequently used approach to reduce this work is for the routing strategies use a spanning subgraph of the unit disk graph such that only the edges in the subgraph are used for routing. Therefore, much research effort has gone into the development of algorithms for topology control for ad hoc networks (see [8], [9] for surveys). In this paper, we will use two different subgraphs of the sensor network. All distance used are Euclidean distances. For the definition of a primary routing algorithm, we need to use a Voronoi diagram [10] which is the partition of the plane into cell, one cell per network node, such that any point in a cell is closer to it's network node than any other network node. To further reduce the node degree during simulations we will use the Adaptive Yao (*DAAY*) subgraph [11] which is locally incrementally defined by connected directed edges to nearest neighbors. Each choosen neighbor defines a cone with its apex at the current node that is centered on the neighbor. The cone angle measured about the current node from the side of the cone to the center of the cone is θ. All the other neighbors in the cone are discarded. This process continues until all the neighbors have been chosen or discarded, creating a directed *DAAY* subgraph. An undirected *DAAY* subgraph is created by ignoring the direction of the edges. The *DAAY* subgraph is strongly connected, has bounded out-degree, is a t-spanner with bounded stretch factor, and orientation-invariant.

A. Routing on UDGs

Routing research in sensor networks and related ad hoc networks is a huge field. We focus on the unicast approach to delivering a packet between a single source and a single destination. Several unicast routing protocols have been presented based on 2-D graphs during the past few years, of which we will focus on position-based routing algorithms. Giordano *et al.* [6] distinguished several classes of position-based routing protocols. We are interested in the class that includes Basic Distance, Progress, and Direction Based Methods. In this type of class, a node A will forward the packet either based on the Euclidean distance to the destination, projected distance to the destination or the direction to the destination. till reaching, if possible, the node D. The main goal is to always choose one of its neighbors that have the most positive progress towards the destination. We refer to such protocols simply as progress-based routing protocols [3], [12]–[14]. These progress-based routing algorithms are locally distributed algorithms since they do not need to gather the global topology information of the network as a whole. Thus, the bandwidth and limited storage resources can be more efficiently utilized. These characteristics also make the position-routing algorithms quick to adapt to network topology changes.

One popular example of progres-based routing algorithms is the *Greedy Forwarding* routing algorithm (denoted by *GF*). There are three main types of *GF*. One is based on the Euclidean distance to the destination [15] which is referred to as geographic distance (*GEDIR*). Another one is based on the projected distance to the destination (the projection is perpendicularly onto the straight line joining the current node and the destination node) [14], which is referred to Most Forward within Radius (*MFR*). The last one is based on the angle between the neighbor and the destination (the angle is measured between the line joining the current node and destination node and the line joining the current node and the neighbor node) [16], which is referred to Compass routing (*DIR*).

In general, progress-based routing protocols may suffer from local minima [3], that is, a current node has no other neighbors that make better progress than itself, or two adjacent neighbors that are equally close to the destination. Hence a loop can occur. Even if *GEDIR* protocol can stop forwarding a packet once detecting a loop, there will be a failure in delivering the packet. To bypass these local minimum during routing, [17] (also see [18]) and [13] use a hybrid routing algorithm that combines greedy routing with face routing. Face routing [3] (or perimeter routing [13]) perform routing on the faces of an extracted planar subgraph of the UDG to guarantee packet delivery. The idea is to use greedy routing until a local minimum is reached whereupon face routing is used until greedy routing can be used again, and so on.

B. Sensing-Covered Networks

Because of the impact of the distribution of sensor nodes in a region on performance, additional issues concerning sensor networks are required and these include the following: coverage and connectivity. Coverage refers to the fact that every single point in a region is at least covered by the sensing range of one node. A sensor node covers a point in a region by its sensing range (sensing radius) that keeps track of the surrounding area. Such a network is called a sensing-covered network. All the nodes in the network will be assumed to have the same sensing range R_s. Connectivity refers to the fact that any node can communication with any other node in the network. Xing *et al.* [19] show that if a sensing-covered network is modeled by an UDG, then to maintain connectivity then $R_c \geq 2R_s$.

There are several routing protocols that have been studied in sensor networks. Some of these include: *GEDIR*, *DIR*, *MFR* (and their 2-hop versions [12]) and Bounded Voronoi Greedy Forwarding (*BVGF*) [20] routing protocols. We will focus on the *BVGF* routing protocol.

The *BVGF* routing algorithm works as follows. A current node checks if the Voronoi cells of its neighbors intersect the line segment joining the source and the destination. See Fig. 1. After having this set of nodes, the algorithm picks the one which is the closest one to the destination. The algorithm continues like this until reaching the destination. This algorithm is based on a strongly connected graph. Define a neighbor of the current node as making positive projected progress if, when both nodes are perpendicularly projected onto the source-destination line, the neighbor is closer to the destination. According to the Theorem 5 in [20], *BVGF*

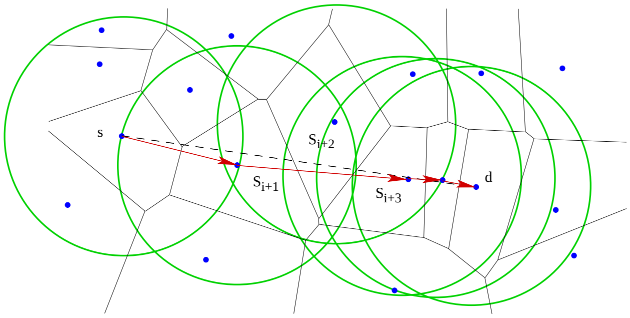

Fig. 1. *BVGF* algorithm. Arrowed path indicates path choosen by algorithm.

always finds a next-hop neighbor with positive progress and thus this algorithm guarantees that the packet is delivered to the destination since the next-hop node has positive projected progress.

III. NEW POSITION-BASED ROUTING ALGORITHMS

To reduce the computation cost of *BVGF*, we introduce a simplified version of *BVGF*, called Sensing Circles Close to the Line Routing Algorithm (*SCL*) replacing the Voronoi cells with the sensing ranges.

A. Sensing Circles Close to the Line Routing Algorithm (SCL)

As mentioned by Xing *et al.* [20], the Voronoi region of a node is contained in its sensing circle. Because of this fact, we introduce a simplified version of *BVGF* routing protocol called *SCL* routing protocol by replacing the Voronoi cells with the sensing ranges of the nodes. The routing algorithm works as follows. A current node checks if the sensing circles of its neighbors intersect the line segment joining the source and the destination. After determining this set of nodes, the current node picks the one which is the closest to the destination. The algorithm continues like this until reaching the destination.

Note that since all the sensing ranges are the same, no direct intersection line intersection calculations are needed. All is needed is to determine the subset of neighboring nodes that are in a corridor centered on the source-destination line of width $2R_s$. With k neighboring nodes, at most $2k$ inside/outside line tests, using implicit line equations for the boundaries of the corridor, are needed. All the nodes in this subset then have sensing ranges that intersect the source-destination line. Compare this to Voronoi cells defined by an average number of Voronoi vertices of at most six [21]. With k neighboring nodes, the *BVGF* algorithm requires at most $6k$ inside/outside line tests (to determine whether the Voronoi vertices of a cell lie on different sides of the source-destination line) determine which Voronoi cells intersect the source-destination line. In addition, the Voronoi cells must be precomputed in $O(n \log n)$ time for the *BVGF* algorithm, whereas for *SCL*, no precomputation is needed.

Because this algorithm uses the sensing ranges instead of the Voronoi cells, this algorithm does not guarantee packet delivery. See the counter-example in Fig. 2. In this figure, the current node (labeled as such) is surrounded by its neighbors. For all nodes, their corresponding sensing ranges are shown. The boundary of the sensing circle of the current node barely crosses the source-destination line. The sensing circle of itsi nearest neighbor (stripped pattern) crosses the source-destination line but the node is assumed to make very small negative progress toward the destination. Although some of its neighbors (lightly shaded) do make positive progress towards the destination, their sensing circles do not intersect the source-destination line. Finally, the remaining nodes (criss-crossed pattern) do not make positive progress towards the destination even though some of their sensing circles intersect the line joining the source and the destination.

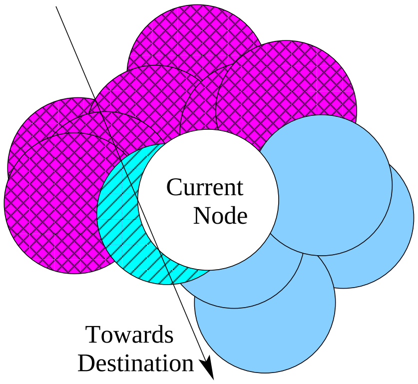

Fig. 2. Example where *SCL* fails to progress. The current node (white) is surrounded by its neighbors. The source-destination line is indicated.

Since the *SCL* routing algorithm does not guarantee delivery, we also propose two new hybrid routing algorithms which combine *SCL* with *GF*-type progress-based protocols specifically designed for dense sensing-covered networks. The progress-based algorithms and hybrid algorithms proposed here guarantee the delivery of packets.

B. New GF-type Progress-based routing algorithms

In this subsection, we introduce two *GF*-type progress-based protocols specifically designed for dense sensing-covered networks. The aim of these two algorithms is to continue to make positive progress toward the destination while remaining as close as possible to the line between source and destination. These routing protocols will be used instead of the *GEDIR* routing protocol as a recovery mode in hybrid protocols discussed later in this Section. The reason for this change is that although *GEDIR* routing protocol always picks a neighbor that is the closest neighbor to the destination, this neighbor might be far away from the line joining the source and the destination.

The first algorithm is the GreedyClose2 routing algorithm, *GC2*. In this algorithm, the current node considers its two neighbors that are closest to the destination and picks the one with the shortest perpendicular projection of the neighbor to the source-destination line (ties are broken arbitrarily). Here the distance of the projection refers to the minimum distance from the neighboring node to the source-destination line. The

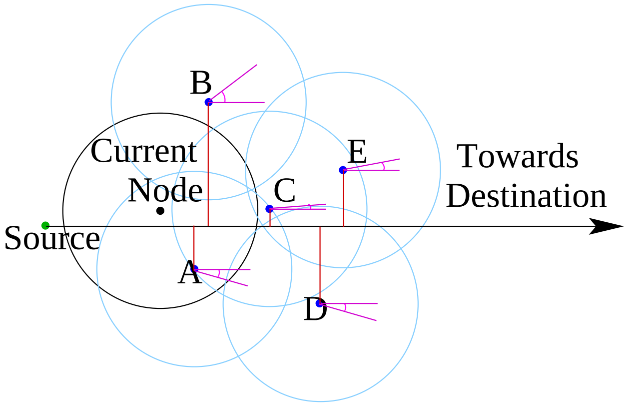

Fig. 3. Comparison of *GC2* and *SAL* algorithms. For *GC2*, the two neighboring nodes closest to the destination are D and E, and E is choosen since its perpendicular projection distance is smallest. For *SAL*, C has the smallest angle (as shown by the wedges) and would be choosen. Sensing circles are also shown.

algorithm continues like this until reaching the destination. Although the *GC2* routing protocol is similar to *MFR* routing protocol, it also tries to stay close to the line joining the source and the destination. See Fig. 3.

Theorem 1: In a sensing-covered network with $R_c \geq 2R_s$, *GC2* guarantees that the packet is always delivered to the destination.

Proof: Consider the point p on the boundary of the sensing circle closest to the destination. In a sensing-covered network, for the immediate region around p to be sensing-covered, p must be in (or on the boundary of) the sensing circles of at least two other nodes at unique positions different from that of the current node. Then the node for a second sensing circle would lie within a circle of radius R_s centered at p, any position within would have positive progress to the destination. And since $R_c \geq 2R_s$ this second node would be a neighbor of the current node. Further, since a neighboring node with positive progress is contained within a circle centered on the destination whose boundary passes through the current node, this node also makes positive projected progress. ■

The next protocol is called the Smallest Angle To The Line Routing Algorithm, *SAL*. The idea behind this protocol is to choose the neighbor with positive progress towards the destination whose angle from the source-destination line is the smallest possible (ties are broken arbitrarily). The angle for one neighbor is between the line joining this neighbor and the source node, and the line joining the source and the destination, with the angle measured from the destination end of the source-destination line. See Fig. 3. The Compass routing protocol is similar to this protocol except for the manner in which the angle is measured.

Theorem 2: In a sensing-covered network with $R_c \geq 2R_s$, the *SAL* routing algorithm guarantees that the packet is always delivered to the destination.

Proof: The proof is similar to the proof presented for the *GC2* routing protocol. ■

C. Hybrid Versions of SCL

As mentioned previously, the *SCL* routing algorithm chooses the next hop if this hop's Voronoi region intersects the line joining the source and the destination; and that has a positive progress towards the destination. To overcome the counterexample presented in Fig. 2, *SCL* routing protocol is combined with either *SAL* algorithm or *GC2* algorithm as hybrid versions of the original *SCL* routing protocol.

For the *SCL*:*GC2* protocol, when *SCL* encounters a failure in the packet delivery at a particular node, it switches to the *GC2* algorithm once and then continues as before with *SCL*. The *GC2* mode allows *SCL* to remain as close as possible to the line segment joining the source and the destination in order to avoid any failure in packet delivery.

Theorem 3: In a sensing-covered network with $R_c \geq 2R_s$, *SCL*:*GC2* will guarantee that the packet is always delivered to the destination.

Proof: Since we shown above that there is always at least one neighbor with positive progress, there are two cases depending on whether a least one of these neighbors with positive progress has a sensing circle that intersects the source-destination line. If there such a neighbor, then the *SCL* routing protocol is used and by Theorem 3 the next-hop neighbor has positive projected progress towards the destination. If there is no such neighbor, the *GC2* routing protocol is used and by the above proof of guaranteed delivery the next-hop neighbor has positive projected progress towards the destination. Thus, the *SCL*:*GC2* routing protocol guarantees that the packet is always delivered to the destination. ■

The *SCL*:*SAL* hybrid algorithm is defined similarly and also guarantees delivery by a similar argument.

D. Hybrid Versions of BVGF

In our simulation of the sensing covered networks, we consider reducing the average node degrees by using subgraphs of the original UDG for the network. A consequence of using a subgraph is that there may now exist nodes where none of the remaining neighbors that make progress to the desination has a Voronoi cell that intersects the source-destination line. To correct for this situation, which would not occur in the original UDG, we introduce two hybrid versions of *BVGF*, *BVGF*:*GC2* and *BVGF*:*SAL*, defined analogously to the hybrid versions of *SCL*.

IV. EVALUATING THE PERFORMANCE OF THE NEW ROUTING ALGORITHMS.

The experiments are done based on a stochastic communication model. This model assumes that $R_c \geq 2R_s$. To create a random sensing-covered network, the model uses an incremental approach using a grid of lines (other approaches have been proposed [19], [22] based on starting off with a dense set of sensor nodes and selectively turning off nodes while maintaining sensing-coverage). Initially, there are no sensing nodes and the grid of lines covers the domain. The parallel spacing between the lines is βR_s which is proportional to the desired sensing radius. Then new sensing nodes with sensing radii $(1-\beta)R_s$ are incrementally generated at random positions in the domain. If the new sensing circle covers a previously uncovered line segment in the grid of lines, it is added to the network, otherwise it is discarded. When the entire grid is covered by the new sensing circles, the radii of the sensing nodes are increased to R_s thus ensuring that any regions between the grid lines are also covered.

Using this model, we place around 1000 nodes randomly in a 500m×500m region in 2-D using $\beta = 1/10$. This region is covered by a set of active nodes creating a sensing-covered network. The transmission radius R_c of each host is 40m with a sensing range R_s of 20m. The largest connected component is exactly equal to the number of nodes since the graph is strongly connected. We repeat this simulation with five different networks using 965, 1012, 973, 960 and 1002 nodes, respectively. For each network, every single node sends a packet to every other node in the network (for a total of 965*964 = 930,260 packets for the first network, and a total of 4,822,790 packets over all five networks). Since the average node degree over the five networks is about 17.5, we perform the routing protocols based on the *UDG* as well as directed *DAAY*s, and undirected *DAAY* subgraphs. As the cone angle in the *DAAY* subgraphs increases, the average node degree decreases. We evaluated the performance of our new routing protocols and compared their performance with *GEDIR* and each other in terms of the following performance measures: running time, hop dilation, and Euclidean dilation. The hop *dilation* is the maximum ratio between any pair of nodes u, v, $u \neq v$, of the hop count of the route path from u to v returned by the algorithm to $\lceil |uv|/R_c \rceil$, and the Euclidean *dilation* is the maximum ratio between any pair of nodes u, v, $u \neq v$, of the Euclidean length of the route path to $|uv|$. All running times for each routing protocol are measured over all packets sent on a networks, summed over the five experimental networks. The cone angle θ in *DAAY* in both directed and undirected versions has the values 0°(i.e., the *UDG*), 10°, 20°, 30°, and 40°(the latter subgraphs having an average node degree of about 7). To ensure that any subgraph still retains at least one neigbor that makes progress to the destination for each node, we use angles less than 45°. The routing protocols running on these graphs achieve 100% delivery rates. The simulation results for the undirected *DAAY* subgraphs where very similar to those for the directed *DAAY*s so we will present just the undirected *DAAY* results.

From the Fig. 4, even though *GEDIR* routing protocol outperforms other routing protocols, the new *SCL* hybrid routing protocols have better hop dilations than the *BVGF* hybrid routing protocols. You can see from the figure also, that the two hybrid versions of *BVGF* are almost the same in their behavior. The same applies for the two hybrid versions of the *SCL* routing protocol. The performance of the *SAL* is similar to that of *GC2* and was not included in the figure. The more θ increases, the more the hop dilation increases for all routing algorithms. The reason is that when θ increases, the nodes in the network will have smaller degrees. Thus, we will start losing some edges which might be part of shorter routing paths to some destination nodes resulting in longer paths.

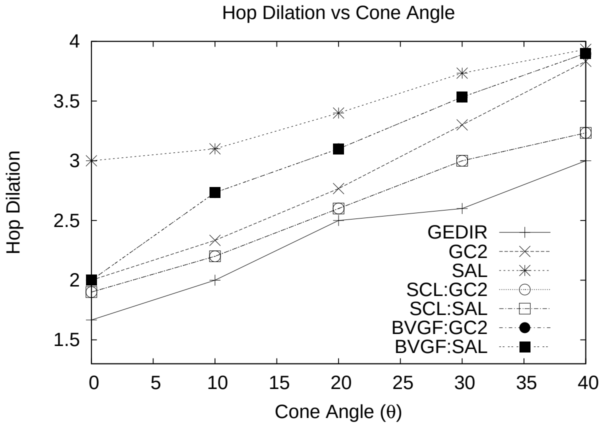

Fig. 4. The hop dilation for different routing algorithms.

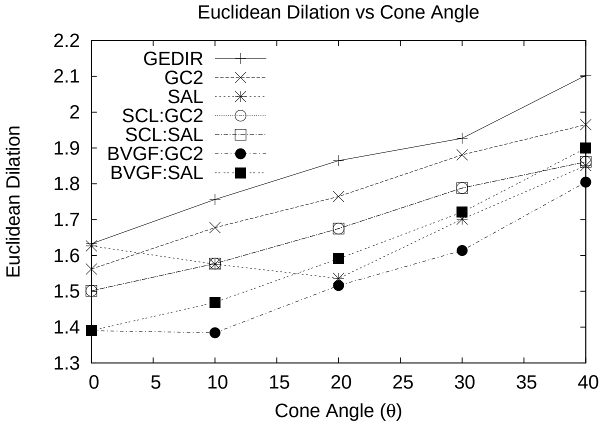

Fig. 5. The Euclidean dilation for different routing algorithms.

Fig. 5 shows the Euclidean dilation for various routing protocols. It is clear that even though the hybrid versions of *BVGF* routing protocol have lower Euclidean dilation compared to other routing protocols; the *SCL*:*GC2* and *SCL*:*SAL* routing protocols are close to *BVGF* versions especially as the angle of the *DAAY* graph increases. This figure also shows that the two hybrid versions of *BVGF* are almost the same in their behavior. The same applies for the two hybrid versions of *SCL* routing protocol. All hybrid routing protocols achieve much better performance than the *GEDIR* routing protocol. As with the hop dilation, the Euclidean dilation increases when θ increases for similar reasons.

As seen in Fig. 6, when θ increases, the number of switches in hybrid algorithms to the *SAL* and *GC2* routing protocols increases, since a node will start losing some edges with neighbors which may have had intersection with the line segment by their Voronoi cells or their sensing circles. The two hybrid versions of *SCL* routing protocol are almost equal in the number of switches to recovery modes. Note that when θ=0, both hybrid versions of *SCL* switch to recovery mode 56 times out of 4,822,790 packet deliveries.

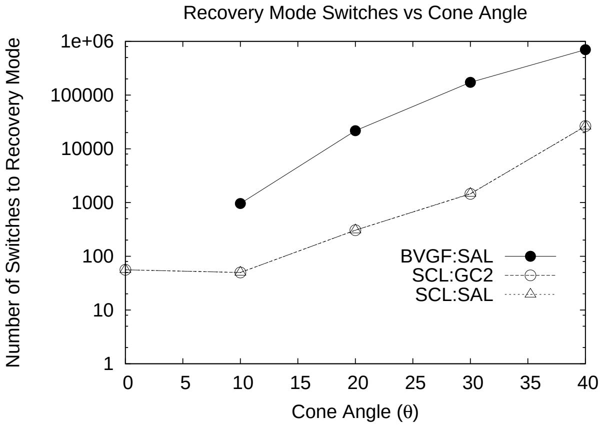

Fig. 6. The number of switches to recovery modes that hybrid versions of *BVGF* and *SCL* can make. Summed over all packets sent over all five experimental networks.

V. CONCLUSION

We demonstrate two hybrid routing protocols using *SCL*, a simplified version of *BVGF*, with guaranteed delivery and nearly equivalent path selection. By avoiding the potential expensive calculation of the Voronoi diagram and by routing using a third the number of inside/outside line test (the most expensive calculation), the overall energy consumption is greatly reduced leading to longer lived sensors. In addition, we demonstrate the benefits of reducing average node degrees in sensing-covered networks using the *DAAY* subgraph, particularly in terms of quicker path determinations by reducing the average degree of each node.

REFERENCES

[1] L. Barrière, P. Fraigniaud, L. Narayanan, and J. Opatrny, "Robust position-based routing in wireless ad-hoc networks with irregular transmission ranges," *Wireless Communications and Mobile Computing Journal*, vol. 3/2, pp. 141–153, 2003.

[2] S. Basagni, I. Chlamtac, V. Syrotiuk, and B. Woodward, " A distance routing effect algorithm for mobility (DREAM)," in *Proc. of 4th ACM/IEEE Conference on Mobile Computing and Networking (Mobicom '98)*, 1998, pp. 76–84.

[3] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, " Routing with Guaranteed Delivery in Ad Hoc Wireless Networks," *Wireless Networks*, vol. 7, no. 6, pp. 609–616, 2001.

[4] P. Bose and P. Morin, "Online routing in triangulations," in *Proc. of 10th Annual Inter. Symposium on Algorithms and Computation (ISAAC '99)*, 1999, pp. 113–122.

[5] R. Jain, A. Puri, and R. Sengupta, "Geographical routing using partial information for wireless ad hoc networks," *IEEE Personal Comm. Magazine*, vol. 8, no. 1, pp. 48–57, 2001.

[6] S. Giordano, I. Stojmenovic, and L. Blazevic, "Position based routing algorithms for ad hoc networks: A Taxonomy," in *Ad Hoc Wireless Networking*, X. Cheng, X. Huang, and D. Du, Eds. Kluwer, 2003, pp. 103–136.

[7] P. Varshney, *Distributed Detection and Data Fusion*. New York: Springer-Verlag, 1996.

[8] R. Rajaraman, "Topology control and routing in ad hoc networks: A survey," *SICACT News*, vol. 33, pp. 60–73, 2002.

[9] X. Li, "Topology control in wireless ad hoc networks," in *Ad Hoc Networking*, S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, Eds. IEEE Press, 2003.

[10] S. Fortune, "Voronoi diagrams and delaunay triangulations," in *Computing in Euclidean geometry*, D. Du and F. Hwang, Eds. World Scientific, 1992, pp. 193–233.

[11] T. Fevens, A. Abdallah, T. El Salti, and L. Harutyunyan, " A Class of Orientation-Invariant Yao-type Subgraphs of a Unit Disk Graph," in *DialM-POMC 2007, The Fourth ACM SIGACT-SIGOPS International Workshop on Foundation of Mobile Computing*, August 2007.

[12] I. Stojmenovic and X. Lin, " Loop-Free Hybrid Single-Path/ Flooding Routing Algorithms with Guaranteed Delivery for Wireless Networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 12, no. 10, pp. 1023–1032, 2001.

[13] B. Karp and H. Kung, "GPSR:greedy perimeter stateless routing for wireless networks," in *Proc. of Sixth ACM Int'l Conf. Mobile Computing and Networking (MobiCom '00)*, 2000, pp. 243–254.

[14] H. Takagi and L. Kleinrock, " Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals," *IEEE Trans. Comm.*, vol. 32, no. 3, pp. 246–257, 1984.

[15] G. Finn, " Routing and Addressing Problems in Large Metropolitan-Scale Internetworks," *Technical Report ISI Research Report ISU/RR-87-180, Inst. for Scientific Information*, 1987.

[16] E. Kranakis, H. Singh, and J. Urrutia, " Compass Routing on Geometric Networks," in *Proc. 11th Canadian Conf. Computational Geometry*, 1999, pp. 51–54.

[17] S. Datta, I. Stojmenovic, and J. Wu, "Internal node and shortcut based routing with guaranteed delivery in wireless networks," *Cluster Computing*, vol. 5, pp. 169–178, 2002.

[18] I. Stojmenovic and S. Datta, "Power and cost aware localized routing with guaranteed delivery in wireless networks," in *Proc. Seventh IEEE Symposium on Computers and Communications ISCC*, Taormina, Sicily, Italia, July 2002, pp. 31–36.

[19] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, and C. Gill, " Integrated Coverage and Connectivity Configuration for Energy Conservation in Sensor Networks," *ACM Trans. Sensor Networks*, vol. 1, no. 1, pp. 36–72, 2005.

[20] G. Xing, C. Lu, R. Pless, and Q. Huang, " Impact of Sensing Coverage on Greedy Geographic Routing Algorithms," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 4, pp. 348–360, April 2006.

[21] X.-Y. Li, G. Călinescu, P.-J. Wan, and Y. Wang, "Localized delaunay triangulation with application in ad hoc wireless networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 14, no. 10, pp. 1035–1047, Oct. 2003.

[22] F. Ye, G. Zhong, S. Lu, and L. Zhang, " Peas: A robust energy conserving protocol for long-lived sensor networks." in *Proc. of 23nd International Conference on Distributed Computing Systems (ICDCS)*, 2003.