Loss-Aware Geographic Routing for Unreliable Wireless Sensor Networks

Euhanna Ghadimi*, Ahmad Khonsari*†, Mohammad S. Talebi† and Nasser Yazdani*

* ECE Department, University of Tehran, Tehran, Iran

† School of Computer Science, IPM, Tehran, Iran

Emails: e.ghadimi@ece.ut.ac.ir, {ak, mstalebi}@ipm.ir, yazdani@ut.ac.ir

***Abstract*—The research community has recently witnessed the emergence of densely deployed Wireless Sensor Networks (WSNs) consisting of a large number of battery-operated sensor nodes. As a candidate for monitoring remote faulty regions, WSNs may suffer from hardware and software faults which may cause misbehavior of a portion of sensor nodes. Geographic routing algorithms aims at traversing data packets in such environments with admissible communication complexity. In this paper, we propose a novel region routing algorithm that addresses message loss tolerability in harsh and hostile environments by assigning higher weights to higher harsh regions and then we present a suboptimal routing in dense WSNs. Finally, we present extensive simulation experiments to validate the accuracy of the proposed algorithm.**

***Index Terms*—Geographic Routing, High Loss Region, Optimization, Routing algorithm, Wireless Sensor Network.**

I. Introduction

Wireless sensor networks with high capabilities of interaction among sensor nodes have expedited the methods of getting knowledge and controlling the environment. Within the last decade, several studies have explored the utilization of sensor networks in many aspects of human life, such as habitat monitoring, pervasive computing, security and tracking [1]. Geographic routing protocols [2][3][4] are very attractive choices for routing in wireless sensor networks. In contrast with the other analogous methods [5][6], geographic based routing protocols conserve energy and bandwidth. Further, in such protocols nodes' storage overhead is low. A Plethora of localization systems such as GPS (Global Positioning System), infrastructure-based localization systems [7], and ad-hoc localization systems [8] have been proposed in the literature to obtain the location information.

Most of geographic routing schemes are lying on nodes sending packets to a neighbor that is closer to the destination. such a greedy forwarding mechanism repeated until packets reach the destination. In [3], authors proposed Greedy perimeter stateless routing (GPSR) protocol, which makes greedy forwarding decisions. In this protocol when a packet is stuck at a node where routing is impossible, the algorithm recovers by routing around the perimeter of the region. Face (perimeter) routing is a common solution when greedy routing fails due to obstacles or dead-ends [4][9]. These algorithms require the communication graph to be planar. In planarization algorithms each node excludes an edge to a neighbor from the planar graph if there is another path through a different neighbor called witness. As an example of methods without planarization, GDSTR [10] is a geographic routing algorithm which has proposed a spanning tree structure called hull tree and suggested an associated convex hull to keep the location of descendant nodes. GDSTR routes packets on the hull tree when greedy forwarding is not possible.

It has been investigated that forwarding methods may suffer from unpredictable conditions as a natural result of node malfunctioning or temporal unavailability of nodes while the nodes in the network are connected [1]. Assuming that all nodes likely suffer from random node failure caused by battery depletion, internal hardware or software faults with equal probability, the whole nodes in a particular geographic region may experience high loss rate conditions. This is mostly caused by harsh environment such as fire, bomb explosion, typical security attacks, e.g. Denial of Service and so on. The region in which such situations occur are usually referred as *high loss region*.

Distributed fault-tolerant event region detection is studied by Krishnamachari et al [11], which uses a Bayesian approach to distinguish single node faults from area faults. Luo et al [12] Suggests a majority voting process between nodes to decide about fault event happening. In our previous study [13], we extended the majority voting scheme, wherein each sensor propagates its assigned confidence value of fault event occurrence to its neighbors. Then, in contrast with binary voting in [12], a weighted averaging process has been done at nodes claiming fault event.

All of the above studies have considered only routing without capturing the harsh environment circumstances and only a few ones have investigated the characteristics of harsh environments. To the best of our knowledge, this is the first work that considers both harsh environment features and routing constraints in such environments and tackles this problem by proposing a reliable routing algorithm. The proposed algorithm aims at reducing overall energy consumption of the network by performing perceptive forwarding decisions so as to minimize the total number of transmitted packets performs through the network.

The rest of the paper is organized as follows. In Section II, we provide some preliminaries for high loss region detection and define the problem of weighted region routing in wireless sensor networks. In Section III, we present our routing algorithm with complexity analysis. In Section IV, we present our simulation results which confirm our theoretical analysis. Finally, Section V ends with concluding remarks.

II. Preliminaries and Problem Definition

We assume a dense sensor network which is deployed in a physical field. Each sensor is equipped with sensing, processing and communication components. Moreover, sensor nodes communicate one another in a uniform topology all over the network. Also, caused by a variety of reasons, some parts of the network may encounter with the fault. Fig. 1 shows a high loss rate region which is detected as a fault event. In this figure, gray nodes are out of the loss region whereas black ones and light ones are lossy nodes and boundary nodes, respectively.

Boundary nodes are responsible for maintaining topological changes in the loss region and thereby they must know each others location. Other nodes only need to know their direct neighbors location to make greedy forwarding decisions. We want to route packets somewhere out of the high loss region (say A in Fig. 1) to the other place in the network (say B).

We model the loss region as a weighted region where each packet is sent hop by hop with a definite probability to send packets occasionally within the weighted region rather than rotating traversing all border nodes (as an alternative for face routing). Thus, we calculate a consensus-based weight for a high loss region and solve the problem theoretically to find the shortest path in a weighted region. Let p be the average massage loss probability in the faulty region obtained via a consensus-based voting. We simply define the weight of the region as following

$$w = \frac{1}{1-p}; \quad p < 1 \tag{1}$$

The problem of finding the shortest path through a weighted region is still an open problem [14]. Considering non-convex polygon, either probable exact or approximate algorithms to find the shortest path in such polygons would have exponential complexity, making them inappropriate for resource limited WSNs. Therefore, we model the weighted region as a convex polygon. The minimal convex set which is the intersection of all convex sets containing all points of a set S is referred to as a *convex hull* [15].

As a local forwarding strategy, in our algorithm, each boundary node upon receipt of a packet whose destination is exterior of the weighted region, decides whether to route the packet through the loss region. We refer to this node as *forwarding node*. The parts of the hull which can forward packets to the destination in a direct path out of the weighted region are called *visibility chain for destination*. Also, two boundary nodes which intersect the tangent line of the destination are called *top tangent* and *down tangent* nodes. For the sake of clarity, these issues are depicted in the Fig. 2.

Lemma 1: Each shortest path entirely within the weighted region crosses only two points on the hull.

Due to space limit we omit the proofs. For a detailed discussion, we refer the interested reader to [20]. As a result of Lemma 1, for the routes entirely through the weighted region, *forwarding node* assumes only the nodes on the *visibility chain*. However, there are other possible shortest paths which consist of routes through the weighted region combined with paths via the perimeter of the area. We call the latter routing strategy *mixed type*. Now, we formulate these two kinds of routes as follows:

For the first type, initially the polygon must be triangulated with *forwarding node* as a common vertex. Each individual edge on the *visibility chain* is viewed as a side of triangle (consisting of *forwarding node* and two adjacent boundary nodes on the *visibility chain*). Then, in each triangle the problem of finding the shortest path must be solved.

The problem of finding the shortest path through a triangle is depicted in the Fig. 3. The main constraint in this problem is the weight w. Every route inside the triangle Δxyz has weight $w > 1$, whereas paths outside the triangle have the weight $w = 1$. The goal is to find point t which minimizes the cost

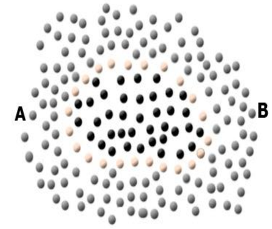

Fig. 1. A high loss rate region. Black nodes are lossy nodes where Gray ones are healthy nodes. The nodes in light red as the boundary region separate loss event from the other parts of the network.

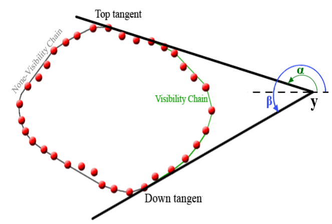

Fig. 2. Visibility chain of a given node y consists of the nodes which are located between the two tangents and the destination.

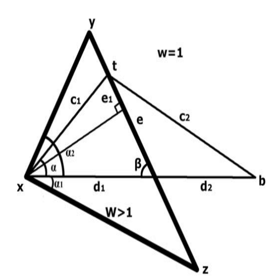

Fig. 3. Shortest path between x, b through a triangle with weight $w > 1$ inside the triangle and $w = 1$ outside the triangle.

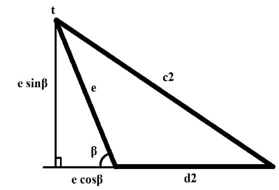

Fig. 4. The triangle which shows the shortest path between point t and destination d (the section of path that is out of the weighted region) which is derived from the Fig. 3

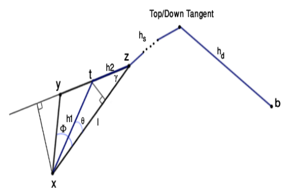

Fig. 5. Mixed type Route. x is forwarding node and b is the destination. The edge yz is an edge on the none-visibility chain.

of the path between x, b. Such a cost is a function of both distance and weight w. Finding t is equivalent to finding the angle α, for the sake of convenience, we aim at determining the angle α. As depicted in Fig. 3, given α, the weighted path comprises of two disjoint segments, namely c_1 and c_2. Both of the aforementioned segments are functions of α. Doing some trigonometric operations, for c_1 we get

$$c_1(\alpha) = \frac{d_1 \sin\beta}{\sin(\alpha+\beta)} \tag{2}$$

Calculation of c_2 needs more elaborations. To do this, we first obtain e as follows:

$$e(\alpha) = \frac{c_1(\alpha)\sin\alpha}{\sin\beta} = \frac{d_1 \sin\alpha}{\sin(\alpha+\beta)} \tag{3}$$

Now c_2 can be expressed as (see Fig. 4.):

$$\begin{aligned} c_2(\alpha) &= \sqrt{e^2(\alpha)\sin^2\beta + (d_2 + e(\alpha)\cos\beta)^2} \\ &= \left(\frac{d_1^2 \sin^2\alpha}{\sin^2(\alpha+\beta)} + d_2^2 + 2d_1 d_2 \frac{\sin\alpha\cos\beta}{\sin(\alpha+\beta)}\right) \end{aligned} \tag{4}$$

In order to find the shortest path, we aim at determining it as the solution to minimization problem, whose cost function is the corresponding weighed distance as following:

$$\min_{\alpha} wc_1(\alpha) + c_2(\alpha) \tag{5}$$

subject to:

$$-\alpha_1 < \alpha < \alpha_2 \tag{6}$$

On the other hand, in *mixed type* routes, for each edge on the *non-visibility chain* exists a route through the weighted region between *forwarding node* and that edge. The other parts of the route are the faces of the polygon toward the border of the *visibility chain* (i.e. *top tangent* or *down tangent*). A mixed type route is demonstrated in Fig. 5.

Form Fig. 5 it is clear that the goal is to determine point t. As with above, we can equivalently determine the angle θ. The *mixed-type* route comprises of many smaller segments, introduced as h_1, h_2, and h_s, etc.

Regarding Fig. 5, through trigonometric manipulations, h_1 and h_2 are given by

$$h_1(\theta) = \frac{l \sin\gamma}{\sin(\theta+\gamma)} \tag{7}$$

$$h_2(\theta) = \frac{h_1(\theta)\sin\theta}{\sin\gamma} \tag{8}$$

As with the weighted distance, we elaborate on finding the shortest mixed-type route by considering it as the solution to the following minimization problem

$$\min_{\theta} wh_1(\theta) + h_2(\theta) + \sum_{s\in NVC} h_s + h_d \tag{9}$$

subject to:

$$0 < \theta < \phi \tag{10}$$

where the lower bound in constraint (10) is established so as to convexify the cost function of (9), and is determined by setting the second derivative of its cost function greater than zero. Due to space limit, we omit the detailed analysis and refer the interested reader to [20].

In the cost function of problem (11), term $\sum_{s\in NVC} h_s$ are the edges of the hull from *forwarding node* to the *top/down tangent*. Besides, term h_d is the line segment between top/down tangent and the destination node. Both of the aforementioned terms are independent from θ and consequently would be omitted from the optimization process.

Both problem (5) and (9) has convex cost functions with linear constraints, hence they admit a unique minimal point which can be achieved using either iterative methods, such as Gradient Projection, Newton's Method, etc. or possibly existing closed form solutions [21]. Since the two aforementioned minimization problems associated with each node are independent from other nodes, the corresponding minimal points would be achieved quite fast, using only per node computations, and there is no need for any coordination amongst other nodes. However, in some scenarios with restricted resources, additional constraints might exist which necessitate balancing the communication load of the nodes on the mixed-type route. Such a case is beyond the scope of this paper and we will pursue it as a future direction.

III. Algorithms

To detect fault regions and boundary nodes we have applied contour tracking algorithm [16]. Also for convex hull algorithm, the most popular algorithms are the Graham scan algorithm [17] and the divide-and-conquer algorithm [18]. Although construction of convex hull and finding visibility chain are two independent jobs, owing to decreasing computational overhead we combine these two actions. We use Lemma 2 to construct convex hull of boundary nodes.

Lemma 2: Top tangent and down tangent points are two vertices of the convex hull.

In our scheme, after boundary region changes, contour network stabilizes then during the first routing through the weighted region, two tangent points are discovered and then convex hull will be constructed. For the convex hull construction algorithm which is based on Andrew's *"Monotone Chain"* algorithm [19] refer to [20]. When convex hull of border nodes constructed, it will be unchangeably used as an approximation of high loss environment until next contour network changes.

Algorithm 1. Pseudo Code for Selecting Minimum Cost Path

Initialization
Initialize the following items:
Input: forwarding node a, destination node d.
Let W be the ordered array of nodes on the hull.
Let *MinPoint* be the point with the minimum cost.
If (*visibility chain* of destination d is not known)
{
  Get the points with min or max angle between given destination d and positive direction of x-axis
  h_{Top} = index of W with min angle.
  h_{Down} = index of W with max angle.
  VC = the indexes of W between h_{Top} and h_{Down} which make visibility chain
  $NVC = W - VC$ }
For $i = 0$ to length of VC (from h_{Top} to h_{Down})
{
  //cost of the route through the triangle (in (7))
  Point t = *ComputeType1MinCost*$(a, VC[i], VC[i+1], d)$
  If cost of the path to t is less than min cost
    MinPoint = t
}
For $j = 0$ to length of NVC
{
  //cost of the mixed type route (in (11))
  Point k = *ComputeMixedTypeMinCost*$(a, NVC[j], NVC[j+1], d)$
  If cost of the path to k is less than min cost
    MinPoint = k
}
Output: *MinPoint*

Each *forwarding node* selects minimum cost path between the local shortest path through the weighted region and *mixed type* paths. For this objective it executes the pseudo-code shown in the Table I. In our routing algorithm, when a *forwarding node* finds shortest path through the weighted region, each node forward packets to the closer neighbor to the predefined shortest path Hop by Hop in a greedy manner.

IV. Experimental Results

In this section we conduct experiments to evaluate our routing algorithm. The experiments are implemented in MATLAB and C++ programming language.

A. Network Setup

We consider the network as a square of side length 50 units where the nodes are deployed using 2-D Poisson process. Each node has transmission radius of 3 units. Base station is located in the center of the network and fault pattern is exerted randomly within the arbitrary shape as inputs similarly to the approach in [16]. The source node is the sink whereas destination node is some where out of the loss region. We use this network setup to run the experiments.

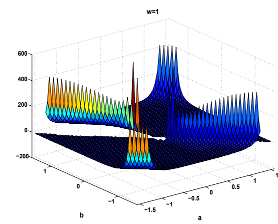

Fig. 6. The Cost Function of The First Optimization Problem with $w = 1$

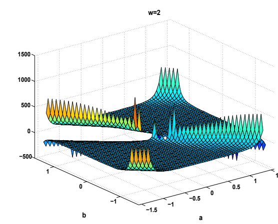

Fig. 7. The Cost Function of The First Optimization Problem with $w = 2$

B. Convexity of Cost Function

In the first experiment, we have investigated the issue of convexity of the cost functions. For this purpose we must make sure that the second derivative of its cost function is greater than zero. Fig. 6,7 depict the curve of the second derivatives of the optimization problem of formula (5). The results show that for different values of weight w, d_1 and d_2 when the independent variables of the problem, i.e. α falls within $(-\frac{\pi}{2}, \frac{\pi}{2})$, then the cost function is convex. Similarly, for the second problem (9) with the same initial conditions, the marginal bounds for θ is $[0, \frac{\pi}{2}]$ (see Fig.8 and Fig.9).

C. Message Count

As the second experiment, we have compared our method with loss unaware greedy routing and GPSR [3]. The comparison is done based on the number of sent messages via

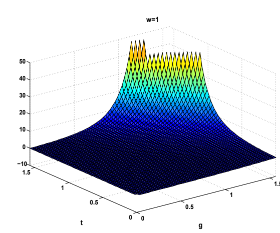

Fig. 8. The Cost Function of The Second Optimization Problem with $w = 1$

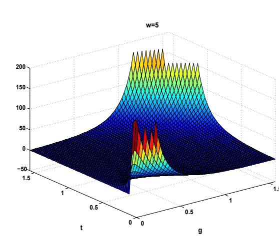

Fig. 9. The Cost Function of The Second Optimization Problem with $w = 5$.

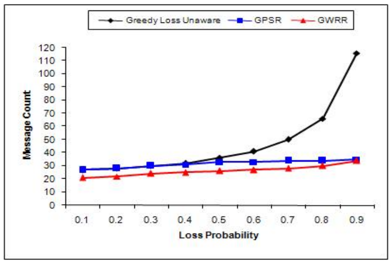

Fig. 10. Our routing algorithm vs. GPSR and loss unaware routing based on number of transmitted messages in different loss probabilities.

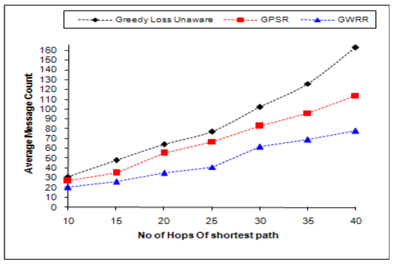

Fig. 11. Our routing algorithm vs. GPSR and loss unaware routing based on average number of transmitted messages for the routes with different length.

these three routing algorithms through the paths between sink and the destination node. We have considered message retransmission whenever message loss occurs in the network. The probability of message loss out of the high loss region is considered moderate (around 5%) whereas the message loss probability in the high loss region increases from 10 to 90 percent. The results are depicted in Fig. 10. We can see that the suggested method outperforms the Greedy forwarding and GPSR in terms of the number of transmitted messages.

The comparison of performance in these three routing algorithms is depicted in Fig. 11. The figure reveals that the number of total transmitted messages in the proposed algorithm in short distance and long distance conditions(e.g., when the source and destination are 10 and 40 hops away.) are about 26% and 32% respectively, lower than results previously reported in companion algorithm in [3]. Also at the similar conditions it performs nearly 35% and 52% better than [5]. One factor leading to this observation is the better ability of the proposed algorithm in deciding between to route through the loss region in low loss situations and then route around the perimeter when the loss probability is excessively high.

V. Conclusion and Future Work

In this paper, we presented a weighted region routing algorithm as a new routing algorithm for lossy wireless sensor networks which is superior to DSR and GPSR algorithms. In this algorithm, the high loss environment is treated as a weighted region and minimal most paths are found toward the destination through the weighted region. Moreover, boundary nodes decide forwarding strategy. As a future direction, we would like to modify the algorithm so that the traffic load on each lossy nodes to be taken into accounts. Comparing the performance of the algorithm with existing fault tolerant routing algorithms is the other future line of this study.

References

[1] K. H. Willig, Protocols and Architectures for Wireless Sensor Networks, John Wiley, New York, 2005.

[2] K. Akkaya, M. Younis,"A survey of routing Protocols in Wireless Sensor Networks", *Elsevier Ad Hoc Network Journal*, Vol. 3, No. 3, pp. 325-249, 2005.

[3] B. Karp, H. T. Kung,"GPSR: greedy perimeter stateless routing for wireless network", *In Proceedings of the 6th annual international conference on Mobile computing and networking (MobiCom '00)*, pp. 243-254, New York, NY, USA, 2000.

[4] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger, "Geometric ad-hoc routing: Of theory and practice", *In Proceeding of 22nd ACM Symposium on the Principles of Distributed Computing (PODC)*, pp.63 - 72, Boston, USA, 2003.

[5] D. B. Johnson, and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks". *In Proceeding of the Mobile Computing conference*, Vol.5, No. 5, pp. 153-181, 1996.

[6] C. E. Perkins, and E. M. Royer. "Ad-hoc on-demand distance vector routing". *In Proceeding of 2nd IEEE Workshop on Mobile Computing Systems and Applications*, 1999.

[7] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Location-Support System", *In Proceedings of the 6th annual international conference on Mobile computing and networking*, pp. 32 - 43, Boston, Massachusetts, USA, 2000.

[8] N. Bulusu, J. Heidemann, D. Estrin, and T. Tran, "Self-configuring Localization Systems: Design and Experimental Evaluation", *ACM Trans on Embedded Computing Systems TECS*, Vol. 3, No. 1, pp. 24 - 60, 2004.

[9] B. Greenstein, D. Estrin, R. Govindan, S. Ratnasamy, and S. Shenker. "DIFS: A Distributed Index for Features in Sensor Networks". *In Proceeding of the First IEEE International Workshop on Sensor Network Protocols an Applications*, pp. 163- 173, 2003.

[10] B. Leong, B. Liskov, and R. Morris. "Geographic routing without planarization". *In Proceedings of the 3rd conference on 3rd Symposium on Networked Systems Design and Implementation*, 2006.

[11] B. Krishnamachari, S. Iyenger, "Distributed Bayesian Algorithms for Fault-Tolerant Event Region Detection in Wireless Sensor Networks", *IEEE Transactions on Computers*, Vol. 53, No. 3, pp.241-250, 2004.

[12] X. Luo, M. Dong, Y. Huang, "On Distributed Fault-Tolerant Detection in Wireless Sensor Networks", *IEEE Transactions on Computers*, Vol. 55, No. 1, pp. 58-70, 2006.

[13] A. Abbasi, E. Ghadimi, A. Khonsari, N. Yazdani, M. Ould-Khaoua, "A Distributed Clustering Algorithm for Fault-Tolerant Event Region Detection in Wireless Sensor Networks",*In Proceeding Of Frontiers of High Performance Computing and Networking ISPA 2007 Workshops*, pp. 493-502, 2007.

[14] E. Demaine, and J. O'Rourke: "Open Problems from CCCG 2006". *In proceeding of the Canadian Conference on Computational Geometry (CCCG 2007)*, pp. 277-280, 2007.

[15] J. O'Rourke, Computational Geometry in C (2nd Edition), Chap. 3 "Convex Hulls in 2D", 1998.

[16] R. Sarkar, X. Zhu, J. Gao, and Joseph S. B. Mitchell. "Light-weight Contour Tracking in Wireless Sensor Networks", *In Proceedings of 27th Conference on Computer Communications. (IEEE INFOCOM'08)*, 2008.

[17] R. Graham, "An Efficient Algorithm for Determining the Convex Hull of a Finite Point Set", *Information Processing Letters*, Vol. 1, pp. 132-133, 1972.

[18] F. Preparata, S.J. Hong, "Convex Hulls of Finite Sets of Points in Two and Three Dimensions", *Communications of the ACM*, Vol. 20, No. 2, pp. 87-93, 1977.

[19] A. M. Andrew, "Another Efficient Algorithm for Convex Hulls in Two Dimensions", *Information Processing Letters*, Vol. 9, No. 5, pp. 216-219, 1979.

[20] E. Ghadimi, A. Khonsari, N. Yazdani "Geographic Routing in Presence of the Loss in Wireless Sensor Networks", *Technical Report*, TRCSE2008-25, University of Tehran, 2008.

[21] S. Boyd, L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.