Stochastic Routing in Wireless Sensor Networks

Fikret Sivrikaya, Thomas Geithner, Cuong Truong, Manzoor Ahmed Khan, and Sahin Albayrak

Technische Universität Berlin, DAI-Labor, 10587 Berlin, Germany

***Abstract*—We propose a new location-based stochastic routing approach that is well-suited for wireless sensor networks deployed for public safety applications such as emergency evacuations or search and rescue operations. We first introduce a new modeling and evaluation framework based on Markov chains for randomized routing. Based on this evaluation framework, we study the load balancing and routing performance aspects of i) a near-optimal solution using the complete topology information, ii) a heuristic algorithm that uses only local neighborhood information. Numerical studies using the evaluation framework show that our heuristic routing approach scales well with both network size and density, considering the combined problem of routing and load balancing.**

I. Introduction

Public safety applications, such as emergency evacuation (of buildings or urban areas) or search and rescue operation after a large-scale natural disaster, can significantly benefit from an earlier deployment of a wireless sensor network (WSN). In this context, the major use of the deployed WSN would be to allow gathering location-specific information from the field and transmitting data to nodes at specific locations for directives and announcements. For instance, in an emergency evacuation application, some sensors may be equipped with appropriate actuators, and the overall information about the field can be compiled and conveyed to those for location-specific dynamic evacuation directives.

Location-based, or geographic, routing is well-suited for such sensor network deployments, where nodes are stationary and the communication paradigm is location-centric as opposed to traditional node-centric addressing. Though the nodes are stationary, there may be frequent node failures, especially during emergency situations, which renders proactive routing methods inefficient and even inapplicable due to stringent energy and latency requirements.

There are two extremes in geographic routing; the greedy strategy and the robust strategy. Greedy strategies may suffer from failures to route packets to destination, while robust strategies need very high flooding rates to ensure guaranteed and rapid delivery of data. Our approach is a combination of the two, where we utilize an adjusted level of greediness as packets travel from source to destination. We provide an overview of related work and contrast those to our approach in the next section.

A. Related Work

Geographic routing protocols are mostly based on the greedy routing principle. Many of these propose methods to overcome the mentioned drawback of greedy routing, i.e. to work around holes (voids) and avoid *dead-ends* in the topology to reach the destination. GRAViTy [1] is one such approach, which uses backtracking from dead-ends and learning from previous routes to optimize subsequent paths in greedy routing. GPRS [2] and GOAFR$^+$ [3] utilize face (or perimeter) routing to go around voids in the topology. All of these protocols use single path routing with deterministic selection of the next hop, and they may suffer from an unbalanced distribution of energy dissipation among sensor nodes in the network. Iterative use of the same routing path, due to the greedy and deterministic selection of next hop, increases the stress on those relaying nodes. Generally speaking, randomized routing is a well-studied approach to overcome this problem, but is mostly overlooked in the geographic routing domain.

Those that utilize randomized routing mostly consider an unknown destination location and employ a pure random walk to discover the destination. Among those, [4] provides an analysis of pure random walks on sensor networks with regular deployment, such as triangular, hexagonal, or square-based topologies. In [5], authors also consider unbiased random walk on a regular deployment of nodes, forming a hexagonal lattice pattern. Another use of unbiased random walks is presented in [6] in order to detect outlier data in sensor networks. Zhang et al. [7] utilize the random walk approach in a rather different context in sensor networks, i.e. in order to enhance the source-location privacy by introducing *phantom* sources in between the actual data source and the sink. A random walk is initiated at the actual source and terminated after a predefined hop count, where the phantom source is created. Data is then sent to the sink by the phantom source using a given routing protocol. An overview and comparison of different random walk strategies for ad hoc networks is given in [8]. None of those approaches consider a given destination location, hence there is no bias toward a target in the probabilities of the random walk.

Geographical Random Forwarding (GeRaF) [9] introduces a new concept of *receiver contention* for packet forwarding. In this scheme, the relaying node does not specify the next hop but the receiving neighbors decide which one should relay the packet based on the location, similar to the greedy approach. The paper presents an analysis of this scheme, but does not fully address how contention among receivers is resolved in a distributed manner. Probabilistic Geographic Routing (PGR) [10] is similar to our approach in the sense that it assigns probabilities to a few candidate relaying nodes, but the assignment is uniform along the routing path, unlike our approach where the probability assignment scheme (bias) changes as packets get closer to the destination. Barrett et al. introduce a family of routing protocols based on probabilistic flooding in [11]. Our approach can be viewed as an additional member to this family, albeit as one not utilizing flooding in order to ensure low power operation.

B. Model and Assumptions

We consider a large multi-hop wireless network consisting of m stationary sensors (nodes), where each node has a fixed circular transmission range, determining the set of nodes it can communicate directly. Such a network is generally abstracted by a graph $G = (V, E)$, where each vertex $v \in V$ represents a node in the network, and each edge $e \in E$ represents the existence of a direct wireless link between two nodes in the network.

We assume that all nodes are aware of their locations at network deployment time (e.g. through GPS receivers), or shortly after deployment by employing a distributed location discovery algorithm, such as in [12]. Moreover, the destination address is assumed to be known to the source node. Recall that a destination address in this context is usually a geographical location rather than an explicit node address, hence this assumption is purely realistic. More specifically data communication in the considered emergency application scenario is used to gather location-specific information from the field or to transmit data to nodes at specific locations. Even though the communication destination might be a set of nodes around a given target location, we can assume without loss of generality that the destination is a hypothetical node at the exact target location.

II. A New Stochastic Routing Approach

Existing geographic routing (GR) algorithms exploit the fact that the shortest path between source and destination nodes in a network gets increasingly closer to the straight line connecting the two nodes in the Euclidean space as node density in the network increases. In wireless sensor networks where limited battery power is a vital resource, frequent use of such *straight line* paths would quickly exhaust the energy of those *on-line* nodes along the path. Thus, critical sensor network applications should employ a load balancing strategy for relaxing the energy consumption in order to maximize the network life-time, while keeping the path lengths for active flows at a reasonable level compared to the shortest paths.

In this paper we present a novel routing approach that integrates the load balancing aspect as an intrinsic property to address the described problem. We model the movement of each packet in the network as a random walk, whose transition behavior is influenced by intermediate nodes along the paths from the packet's source to the destination. Upon receiving a packet to be forwarded, an intermediate node *greedily* assigns transition probabilities to all of its 1-hop neighbors and randomly forwards the packet to a neighbor based on those probabilities.

The novelty in our approach stems from the adjustment of the level of *greediness* in assigning probabilities of the random walk, which we formulate as a set of requirements:

- *Load distribution* – In the first steps of the random walk, the packet should tend to discover many different paths, rather than biasing too much toward the shortest path.
- *Convergence* – In later steps, the bias should be increased toward the nodes that are closer to the destination.
- *Guaranteed delivery* – All neighbors of a node should be assigned nonzero probabilities.

These requirements describe a generic randomized routing approach that can be realized or implemented in different ways according to the assignment method of probabilities. In Section IV we present one such implementation as a heuristic algorithm and analyze its performance, before which we first provide a generic analysis framework using Markov chains.

III. Modeling Framework based on Markov Chains

A Markov chain can be specified by a set of *states*, $S = \{s_1, s_2, \ldots, s_r\}$, and a *transition probability matrix* P, whose entry $p_{ij} \in P$ represents the probability of the chain to be in state s_j at the next step given that it is currently in state s_i. A state s_i of a Markov chain is called *absorbing* if it is not possible to leave it once reached (i.e., $p_{ii} = 1$). A Markov chain is *absorbing* if it has at least one absorbing state and if from every state it is possible to reach an absorbing state. In an absorbing Markov chain, a state which is not absorbing is called a *transient state*.

We model the routing process for a given flow (i.e. a source-destination pair) as an *absorbing Markov chain*, where network nodes represent the set of states in the Markov chain. A destination d for a flow in the network corresponds to an absorbing state d of the Markov chain.

The average number of hops for packets to travel from source s to destination d corresponds to the number of steps for the Markov chain, starting from state s, to be absorbed. For a given routing process, we construct the transition probability matrix P of the Markov chain by assigning each entry p_{ij} the probability of a packet to be forwarded from node i to node j for a given source-destination pair. In order to compute the number of steps for the Markov chain to be absorbed, we first write the probability transition matrix P in its *Canonical form* as in (1), where I is the identity matrix, Q is an $(m-1) \times (m-1)$ matrix, and R is a column vector with $m - 1$ entries.

$$P = \begin{pmatrix} Q & R \\ 0 & I \end{pmatrix} \tag{1}$$

Then the *fundamental matrix* of P is computed as $N = (I - Q)^{-1}$, whose entry $n_{ij} \in N$ represents the expected number of times the process is in the transient state s_j, if it started in state s_i. Thus, adding all entries in the row i of N yields the expected number of steps required before the chain is absorbed. This can be written as $T = Nc$, where c is a column vector of 1s. Then the entry t_i of T gives the expected number of steps until the Markov Chain reaches an absorbing state.

Therefore, the expected number of hops for packets to be routed from source s to destination d is given by t_s and the average node utilization of any node j is given by n_{sj}, for a given source and destination pair. This Markov-based framework allows a flow-based analytical evaluation of any memory-less random walk approach.

A. Evaluation Metrics

In this paper we are interested in two objectives for a routing protocol. One is efficiently delivering packets to their destination in terms of the number of hops required, and the other is spreading the energy dissipation among nodes in the network. There is an apparent trade-off between these two objectives. On the one hand, repeated use of shortest paths provide efficient routing with a poor load balancing aspect. On the other hand, using many alternative paths provides a fairer load distribution while requiring longer paths.

As a metric for the first objective, we use the *flooding rate* f, which is the ratio of the number of message transmissions to the shortest possible hop count between two nodes, providing a measure of the routing efficiency [13]. Since t_s is the expected path length for a given source s and destination d in our framework, f can be expressed as

$$f = \frac{t_s}{h_{sd}}, \tag{2}$$

where h_{sd} is the length (in hops) of the shortest path between nodes s and d.

As radio communications constitute the major source of battery usage in sensor networks, we use node utilization as a measure of the energy dissipation. Given that n_{sj} represents the utilization of node j in our framework, the mean utilization among all nodes for the given flow can be expressed as

$$C = \frac{1}{m-1}\sum_{i=1}^{m-1} n_{si},$$

where m is the number of nodes in the network. Then the normalized average deviation from the mean utilization, the metric for the second objective, is given by

$$\delta_C = \frac{1}{(m-1)C}\sum_{i=1}^{m-1} |n_{si} - C|. \tag{3}$$

The normalization by the term C in the denominator ensures that deviation valuaes are comparable between different topologies. For an integrated approach that addresses both objectives, we introduce a linear combination of the two metrics, controlled by a parameter $\alpha \in [0,1]$. For a given value of α, a routing protocol should aim at minimizing the objective function F.

$$F = \alpha \cdot \delta_C + (1-\alpha)f. \tag{4}$$

B. Optimal Transition Probability Matrix

In this section we study the optimal transition probability matrix, which minimizes the objective function F. This centralized theoretical solution provides a means to quantify the performance of a distributed heuristic approach, as we present in the following sections. An obvious first attempt in obtaining the optimal transition probabilities would be to use an optimization method such as linear programming, which requires bringing (4) into the form $max\{c^T n \mid Ax \geq b, x \geq 0\}$. However, the term δ_C cannot be written as a linear combination of N, preventing the use of Linear Programming.

We observe that any set of probability assignments results in a valid solution as long as the sum of all probabilities for each node is exactly one. Thus, a set of valid solutions can easily be generated by randomly assigning forwarding probabilities that fulfill this condition. Moreover, a non-optimal solution can mostly be improved by adjusting the probabilities of one or more nodes in the network, without affecting the others. Based on these observations, we use a *genetic algorithm* to compute near-optimal solutions for the optimization problem defined by (4), where a *gene* consists of the probability set of one node (which corresponds to the vector $G_i = (p_{i1}, \ldots, p_{im})^T$ in the Markov chain model). Thus, one gene can be modified independently, affecting the result (value of the objective function) but not the validity of the solution. The set of all genes of a solution forms the *genome* of the *individual* (solution). To compare different individuals, the Markov-based evaluation framework as described in Section III is used.

Our genetic algorithm consists of the following steps:

1) *Initialization:* Create the first generation of individuals by randomly generating their genomes.
2) *Comparison and Selection:*
   - Compare the quality of all individuals using the objective function.
   - Select a specified number of individuals for the reproduction phase based on the comparison.
3) *Reproduction:*
   - Create a new generation of individuals by combining the genomes of the parent generation.
   - Apply some random modifications (mutations) to the genes of the new generation.
4) *Iteration:* Repeat steps 2 and 3 until the termination condition (here the number of iterations) is reached.

As in any genetic algorithm, the quality of the results generated by this algorithm depends on the choice of its parameters, i.e. population size, number of selected individuals per iteration, mutation rate, and number of iterations. We fine tune the parameters by some empirical adjustments, while some special cases are also validated with respect to the actual optimum solutions (e.g. for computing the shortest paths by setting the parameter α in (4) to zero). The algorithm was implemented in *GNU Octave* [14], a *MATLAB* [15] compatible language.

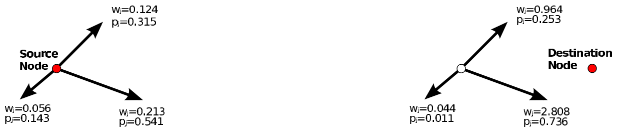

Fig. 1. Example: heuristic probability assignment

IV. Distributed Heuristic Algorithm

We conceptualize and demonstrate our approach described in Section II with a simple heuristic for assigning the forwarding probabilities. Consider a source or relaying node i in a transmission session with packets destined to destination d. Let $\Delta(u, v)$ represent the Euclidean distance between the two nodes u and v and π_i the set of neighbors of node i. For any neighbor j of i ($j \in \pi_i$), a weight w_j is assigned using eq. (5).

$$w_j = \frac{\Delta(i, d) + \max\limits_{k \in \pi_i}\{\Delta(i, k)\} - \Delta(j, d)}{\Delta(j, d)}. \tag{5}$$

Then, all weighs of node i are normalized using Eq. (6), which results in a set of forwarding probabilities for node i. Effectively, this heuristic algorithm constructs the transition probability matrix P described in Section III in a distributed manner.

$$p_j = \frac{w_j}{\sum\limits_{k \in \pi_i} w_k}. \tag{6}$$

Fig. 1 illustrates the probability assignment for two nodes, the source node (left) and an intermediate node close to the destination (right), each with three neighbors at identical relative positions. We observe that all neighbors are assigned non-negligible forwarding probabilities at the source, while those probabilities converge toward the destination direction as the packet gets closer to the destination node. This heuristic uses only local information in computing the probabilities and satisfies all three requirements we outlined in Section II.

Voids may emerge in the network topology especially when there is a sparse deployment of nodes. In some cases this may result in a poor selection of paths by the heuristic algorithm, which is a common problem for geographic routing protocols, especially for those employing the greedy strategy. In pure greedy strategies, the destination may not even be reached unless there is an extra mechanism to overcome the voids problem. In our approach, non-zero probability assignment to all neighbors ensures that the destination will be reached, as long as the source and destination is connected. This can be easily proven by the Markov chain analysis framework. On the other hand, resulting routing paths in the existence of voids may be very ineffective in terms the number of hops to reach the destination. In this work, we consider sensor networks for emergency applications with a dense deployment of nodes for increased robustness; hence, we do not explicitly address route optimizations to efficiently surpass voids in topology. Those techniques covered in Section I-A are complementary to our approach and can be employed in highly irregular sensor deployment scenarios.

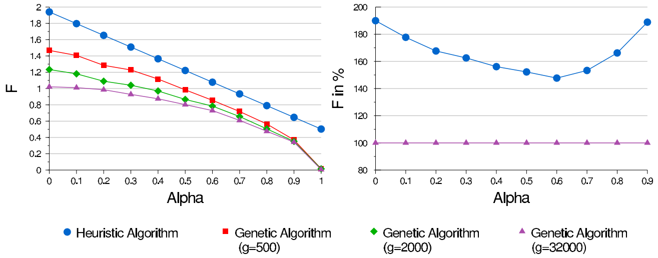

Fig. 2. Comparison between heuristic and genetic algorithm.

V. Results and Analysis

In this section we use our modeling framework of Section III to obtain numerical results for assessing the performance of our heuristic routing algorithm. We utilize the near-optimal solutions obtained by the genetic algorithm as a benchmark for evaluating the results of the heuristic approach.

A. Comparison of Heuristic Approach and Optimal Solution

We first investigate the behavior of the heuristic and genetic algorithms for various linear combinations of the two objectives, controlled by the parameter α. Since the genetic algorithm requires high computation time for obtaining close-to-optimal results, we use a moderate network size of 100 nodes for the results in this section. Nodes are randomly scattered in a unit square area with a transmission range of 0.2 units. Scalability results for the heuristic approach using larger network sizes are presented in the next section.

Fig. 2 shows the comparison results for varying α values over the range of [0,1]. Note that α does not affect the operation of the heuristic algorithm, but the objective function value is naturally affected by different values of α. For the genetic algorithm, each selected α value requires a separate optimization process. The parameter g in the figure represents the number of generations used for the genetic algorithm. The diagram on the left shows the absolute value of F depending on the selected value for α. It also illustrates the convergence of the genetic algorithm result toward the optimal solution.

The diagram on the right shows the same values normalized with respect to the best known solution, such that the results of the genetic optimization with $g = 32000$ are fixed at 100%. We observe that the performance of the heuristic algorithm is at its best compared to the optimal solution when α is around 0.6. This demonstrates that the heuristic approach is effective in simultaneously addressing both objectives of efficient routing and load balancing. It is important to note that, for all values of α, the heuristic algorithm always achieves a performance figure that is less than twice that of the best known solution, which uses the centralized genetic algorithm.

B. Scalability

We now investigate the behavior of the heuristic algorithm in more detail using different network topologies with respect to parameters such as network size, node density, and network diameter. Two scenarios are evaluated for the results in this section; i) a varying number of nodes with fixed network

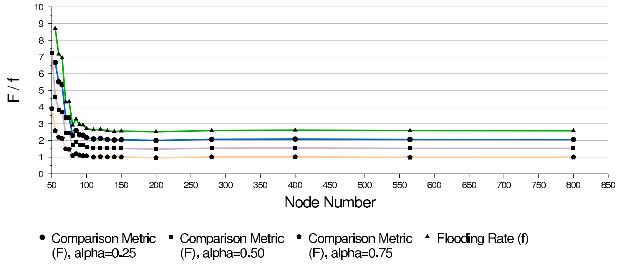

Fig. 3. Heuristic algorithm performance with increasing node density in a fixed network area.

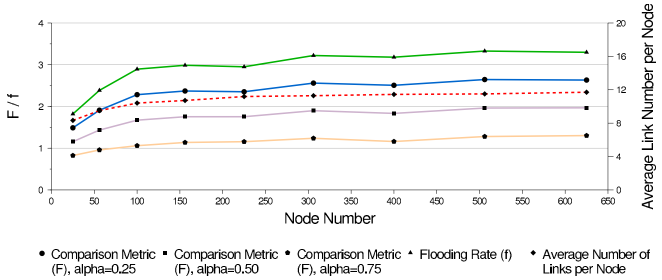

Fig. 4. Heuristic algorithm with different network sizes for a fixed node density.

dimensions to analyze the influence of node density, and ii) a growing network size (in terms of both network dimensions and the number of nodes) while keeping node density constant in order to investigate scalability. In the first case, we use a unit square again as the network dimensions while the number of nodes is varied between 25 and 800. The second scenario uses a fixed node density with the network dimensions being varied by setting the edge size to values between 0.5 and 2.5 units (which results in network sizes between 25 and 625 nodes). In both scenarios, every data point is an average over 100 random networks. For each random network, source and destination nodes are selected apart from each other so that the distance between the nodes scales with the network diameter.

Fig. 3 shows the behavior of the heuristic algorithm in a network with constant dimensions and a varying number of nodes. For very small node densities we observe a rather poor performance for the heuristic, but then it quickly stabilizes and stays constant regardless of the network size or density. In Fig. 4, we proportionally increase the network dimensions and the number of nodes together, thus keeping node density constant. Note that for small networks, the availability of different choices at intermediate nodes are limited, which narrows down the spectrum of possible performance figures of different routing algorithms. Hence the heuristic approach also performs better in small networks as expected. This behavior can be observed by the correlation between the objective function F and the average number of links per node (dotted line in Fig. 4), which depends on node density and transmission range as well as edge effects (that have a higher influence on smaller networks).

VI. Conclusion

In this work we introduced a new approach to stochastic routing in wireless sensor networks that is suitable for disaster recovery and emergency applications, and proposed a distributed heuristic algorithm to implement this approach. We presented a generic evaluation framework for modeling and analyzing random walk based routing approaches, and defined the metrics for the integrated objectives of efficient routing and load balancing. We also implemented a genetic algorithm to find the near-optimal transition probability matrix that minimizes a given linear combination of the two objectives. Numerical results demonstrate that our distributed heuristic approach performs well using as a benchmark the centralized solutions of the genetic algorithm and that it scales well with respect to network size and density.

References

[1] T. Dimitriou and I. Krontiris, "Gravity: Geographic routing around voids in sensor networks," *International Journal of Pervasive Computing and Communications*, vol. 2, pp. 351–361, 2006.

[2] B. Karp and H. T. Kung, "Gpsr: greedy perimeter stateless routing for wireless networks," in *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2000, pp. 243–254.

[3] F. Kuhn, R. Wattenhofer, and A. Zollinger, "An algorithmic approach to geographic routing in ad hoc and sensor networks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 1, pp. 51–62, 2008.

[4] H. Tian, H. Shen, and T. Matsuzawa, "Randomwalk routing for wireless sensor networks," *Parallel and Distributed Computing, Applications and Technologies, 2005. PDCAT 2005. Sixth International Conference on*, pp. 196–200, Dec. 2005.

[5] I. Mabrouki, X. Lagrange, and G. Froc, "Random walk based routing protocol for wireless sensor networks," in *ValueTools '07: Proceedings of the 2nd international conference on Performance evaluation methodologies and tools*, 2007, pp. 1–10.

[6] K. Padmanabh, A. Vanteddu, S. Sen, and P. Gupta, "Random walk on random graph based outlier detection in wireless sensor networks," *Wireless Communication and Sensor Networks, 2007. WCSN '07. Third International Conference on*, pp. 45–49, Dec. 2007.

[7] L. Zhang, "A self-adjusting directed random walk approach for enhancing source-location privacy in sensor network routing," in *IWCMC '06: Proc. of the 2006 International Conf. on Wireless Communications and Mobile Computing*. New York, NY, USA: ACM, 2006, pp. 33–38.

[8] S. S. Dhillon and P. V. Mieghem, "Comparison of random walk strategies for ad hoc networks," in *Proceedings of the Sixth Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net 2007)*, Corfu, Greece, June 2007, pp. 196–203.

[9] M. Zorzi and R. R. Rao, "Geographic random forwarding (geraf) for ad hoc and sensor networks: multihop performance," *IEEE Transactions on Mobile Computing*, vol. 2, no. 4, pp. 337–348, 2003.

[10] M. M. T. Roosta and S. Sastry, "Probabilistic geographic routing protocol for ad hoc and sensor networks," in *Proc. Int. WorkshopWireless Ad Hoc Networks*, 2006.

[11] C. L. Barrett, S. J. Eidenbenz, L. Kroc, M. Marathe, and J. P. Smith, "Parametric probabilistic routing in sensor networks," *Mob. Netw. Appl.*, vol. 10, no. 4, pp. 529–544, 2005.

[12] S. Meguerdichian, S. Slijepcevic, V. Karayan, and M. Potkonjak, "Localized algorithms in wireless ad-hoc networks: location discovery and sensor exposure," in *MobiHoc '01: Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*. New York, NY, USA: ACM, 2001, pp. 106–116.

[13] I. Stojmenovic and X. Lin, "Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 12, no. 10, pp. 1023–1032, Oct. 2001.

[14] "GNU Octave." [Online]. Available: http://www.gnu.org/software/octave/

[15] "MATLAB." [Online]. Available: http://www.mathworks.de/