PWave: A Multi-source Multi-sink Anycast Routing Framework for Wireless Sensor Networks

Haiyang Liu[1], Zhi-Li Zhang[1] ⋆, Jaideep Srivastava[1], and Victor Firoiu[2]

[1] Dept. of Comp. Sci. and Eng., Univ. of Minnesota, Minneapolis, MN 55414
[2] Advanced Information Technologies, BAE Systems, Burlington, MA 01803

Abstract. We propose a novel routing framework called PWave that supports multi-source multi-sink anycast routing for wireless sensor networks. A distributed and scalable potential field estimation algorithm and a probabilistic forwarding scheme are proposed to ensure low overhead and high resilience to network dynamics. Key properties of this framework are proved through theoretical analysis and verified through simulations.Using network lifetime maximization problem as one example, we illustrated the power of this framework by showing a 2.7 to 8 times lifetime extension over Directed Diffusion and up to 5 times lifetime extension over the energy-aware multipath routing.

1 Introduction

Wireless sensor networks (WSNs) are generally deployed to support specific missions or applications such as habitat monitoring, object tracking. Traffic are generated from a number of sensing sources and collected by (any one of) a few sinks [1]. Hence data communications in WSNs exhibit *multi-source*, a *multi-sink anycast* pattern, which is fundamentally different from that in general-purpose communication networks (whether wired or wireless), where any two nodes may serve as the two ends of an end-to-end communication. In addition, WSNs often operate in challenging environments and are subject to frequent disruptions and node failures. These unique settings and constraints call for a robust routing framework for WSNs that can quickly adapt to changes in traffic pattern, network conditions and environments.

Existing routing schemes (e.g., [2]) used in WSNs are either variations, or even direct adoption, of routing algorithms for general-purpose wired networks, or mobile wireless ad hoc networks (MANETs), which are typically designed using the single shortest path *unicast* routing paradigm[3]. While several multi-path routing schemes (see, e.g., [3–5]) have been proposed, they tend to be extension of the single shortest-path routing paradigm with use of additional paths – choice of these alternative paths is often decided based on somewhat *ad hoc* mechanisms. More importantly, these techniques

⋆ This work was supported in part by the NSF grant CNS-0435444 and a DoD Army High Performance Computing Research Center grant.

[3] While offering some desirable local properties, the more recent geographic or trajectory-based routing paradigm is still unicast based, using “shortest distance” paths in an Euclidean or metric space. In addition, it requires some type of location information, which may not be easy to obtain.

only support *single-sink* configuration (i.e., unicast routing) with no direct and easy extension to support *multi-source, multi-sink anycast* routing.

In this paper we propose a novel framework – referred to as *PWave* to support multi-source, multi-sink anycast routing that is inherent in WSNs. Inspired by the analogy between WSNs and electric networks, PWave constructs a *potential field* by assigning a "potential" (analogous to "voltage" in an electric network) to each node: a source or an intermediate node routes traffic (proportionally) to neighboring nodes with lower potentials towards the sinks, which have the lowest (zero) potentials. The PWave framework is designed with strong theoretical underpinnings. First of all, the constructed potential field realizes scalable, robust, proportional multi-source, multi-sink traffic allocation that optimizes a *customizable* quadratic function (based on appropriate definition of "link costs"). Furthermore, it guarantees that there is no local minima (i.e. packets are never stuck in a local dead-end) and thus ensures loop-free routing. It adapts to local changes rapidly, while dampening their global impact. These features enable compact and efficient protocol design with low execution overhead.

We develop a fully distributed algorithm for constructing the potential field and implement PWave using probabilistic forwarding to achieve the properties described above. PWave scales to the density of the network because only one-hop neighborhood information exchange is needed. In addition this algorithm is resilient to network dynamics in that local perturbations only have local effect. These features make PWave a suitable routing framework for WSNs. In a nutshell, the research contribution of this paper lies in the proposal of a novel routing framework that supports global optimization of custom objectives in a multi-source multi-sink anycast routing settings via fully localized computations. To the best of our knowledge, this is the first systematic routing framework for WSN with this capability.

The remainder of this paper is organized as follows. We detail the PWave framework design in section 2. The potential field estimation algorithm is described in section 3, followed by the experimental evaluations in section 4. We describe related work in section 5 and conclude in section 6.

2 The PWave Routing Framework

2.1 System Model and Problem Formulation

In order to formally define the PWave routing framework, we first introduce the system model, notations and assumptions. We assume that a WSN can be represented as a weighted (undirected) graph **G** = (**N**, **E**), where **N** is the set of nodes and **E** the edges (i.e., links) between nodes which are assumed to be *symmetric*. Asymmetric links are blacklisted as suggested in literatures. We use $R_{x,y}$ denote the weight of an edge *e* connecting nodes x and y, which is strictly positive, i.e., $R_{x,y} > 0$. This weight represents some measure of *unit cost* for transmitting one bit of information between x and y defined in certain manners depending on the applications and routing design objectives. (We will provide some examples of $R_{x,y}$ later.) Hence if $I_{x,y}$ amount of data is transmitted from x to y, the total cost would be $I_{x,y}R_{x,y}$. For simplicity, we assume that data traffic only flow along one direction of an edge. The cost for acknowledgements is implicitly accounted for in $R_{x,y}$. Under this assumption, $I_{x,y}$ denote the data rate flowing from x to y, we define $I_{y,x} = -I_{x,y}$. The same relation also holds if $I_{y,x}$ is the data rate flowing from y to x.

Let $\mathbf{S} \subset \mathbf{N}$ denote the set of source nodes, and $\mathbf{D} \subset \mathbf{N}$ be the set of sink nodes. For each $s \in \mathbf{S}$, I_s denote the data rate that may be generated by source node s. More generally, to account for potential in-network processing at intermediate nodes in a WSN that may increase or decrease the data rate flowing through them, for each $x \in (\mathbf{N} - \mathbf{D})$, we use I_x to denote the (internal) data generation/consumption rate at node x. Note here $I_x > 0$ means that data is generated at node x while $I_x < 0$ means that data is consumed at node x. For each node $x \in \mathbf{N}$, we use $Z(x)$ to denote the set of its neighboring nodes. Then the *flow conservation law* requires that for any node x that is not a sink, the total of data rates flowing into node x is equal to the total of data flowing out of node x plus or minus the data rate generated or consumed at node x itself. Namely,

$$\sum\nolimits_{y \in Z(x)} I_{x,y} = I_x, x \in (\mathbf{N} - \mathbf{D}). \tag{1}$$

Given the graph $\mathbf{G}$ representing a WSN and $\mathbf{I} := \{I_x | x \in \mathbf{N}\}$, we refer to the tuple ($\mathbf{G}$, $\mathbf{I}$) a network configuration **NC**. Given a **NC**, routing for a WSN can be casted as a global *multi-source multi-sink anycast flow allocation optimization problem* to determine the flows $\{I_{x,y}\}$ along the links under the flow conservation constraints (1) and boundary conditions (2) such that certain global objective function $\mathcal{F}(\mathbf{G}, \mathbf{I}, \{I_{x,y} | (x, y) \in \mathbf{E}\})$ can be optimized. Single path (or minimum cost) routing that computes a minimum cost path for each source to one of the sinks is such a flow allocation scheme that allocates flows based *only* on the cost of the paths, but not on the flow rates.

2.2 PWave Routing Framework

Intuitions and Principles From physics, it is well known that if the energy level of a physical system is minimized, the system would be in most stable state, i.e. the system will tend to go back to this state after disturbances. A routing framework designed this way will thus be robust. With this intuition, we design our PWave routing framework to solve this optimization problem by minimizing a natural quadratic objective function (4), which is equivalent to the total energy of a corresponding electric network system (see Fig. 1).

PWave solves the flow allocation optimization by assigning a potential field to the nodes in a WSN, namely, a function $V : \mathbf{N} \rightarrow R^+$, where R^+ denote the set of nonnegative real numbers. The potential function V satisfies the following *boundary conditions* at the *sink* nodes

$$V_d = 0, d \in \mathbf{D} \tag{2}$$

and the *flow distribution* conditions at *non-sink* nodes

$$I_{x,y} = \frac{(V_x - V_y)}{R_{x,y}} \tag{3}$$

as well as the flow conservation constraints (1) at the non-sink nodes.

The equation (3) specifies a *localized rule* on how data are routed at each node x based on local information, namely, its and its neighbors' potentials: data only flow from node x towards one of its nodes y with lower potentials ($V_y < V_x$) and the amount of data routed along the edge (x, y) is inversely proportional to $R(x, y)$, or equivalently, proportional to $g_{x,y} := 1/R_{x,y}$, which is referred to as the *conductance* of edge (x, y). The boundary conditions (2) ensure that the sink nodes have the lowest possible potential (namely, zero potential) so that data will always flow towards the sinks and are eventually "absorbed" at the sinks. As will be shown shortly, the potential field defined above guarantees the existence of a *unique* flow allocation $\{I_{x,y} : (x, y) \in \mathbf{E}\}$ such that it minimizes the following *global* objective function:

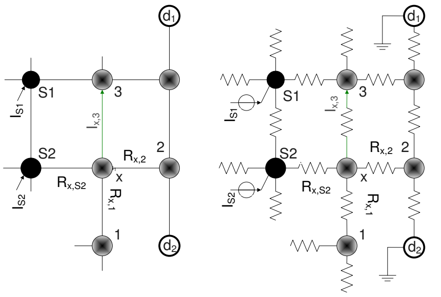
$$E = \frac{1}{2}\sum_{x,y} I_{x,y}^2 R_{x,y} = \frac{1}{2}\sum_{x,y} g_{x,y}(V_x - V_y)^2. \tag{4}$$

Moreover, we will see that the potential-based PWave routing framework also allows for a *probabilistic routing/forwarding* implementation at the packet level. More specifically, equation (3) allocation can be achieved in practice through forwarding a packet from node x to one of its lower potential neighbors y with probability given by

$$p_{x\to y} = \frac{I_{x,y}}{\sum\limits_{i:i\in Z(x)\wedge V(i)<V(x)} I_{x,i}} \tag{5}$$

where $I_{x,i}$ is computed from local potential values using eq.(3).

In summary, the potential-based PWave routing framework enables us to achieve goals at three different levels: i) at the *network-wide* macroscopic level, it minimizes a natural *global* objective function (4); ii) at the intermediate flow level, it provides a *localized* rule to determine how data flows are routed; and iii) at microscopic packet level, it allows for a simple probabilistic packet forwarding mechanism to achieve both flow-level routing and network-wide design objectives.



(a) Communication Network

(b) Equivalent Electric Network

Fig. 1. Analogy between WSNs and Resistive Electric Networks.

Example Applications Before we leave this section, we remark that by choosing different interpretations (and thus different values) for the edge costs R_{xy} (or equivalently, the edge conductance, g_{xy}), we can use eq.(4) to optimize different design objectives in routing for WSNs. For example, if we set $R_{xy} = 1$, then data flows are approximately proportionally distributed and routed based on path hop counts. If we set R_{xy} equal to the data loss rate on an edge, then minimizing eq.(4) would yield a flow allocation/routing strategy that attempts to approximately equalize the data losses among different paths. As another example, if we set $R_{x,y}$ as a combination of the per unit power consumed by transmitting one bit of data, denoted as CE_i, and the current energy level, denoted as EN_i, as follows:

$$R_{i,j} = \frac{1}{g_{i,j}} = (\frac{CE_i}{EN_i} + \frac{CE_j}{EN_j}) \tag{6}$$

the PWave routing framework yields a solution that approximately equalizes the power consumption among various paths to maximize the network lifetime. See [6] for detailed derivations.

3 Potential Field Estimation

3.1 Principles and Algorithm

The potential field estimation problem is governed by eqs.(1, 3) under boundary conditions specified in eq. (2). The existence and uniqueness of the solution to this problem are well known from electric network theory [7]. Traditional way of solving this problem is to rewrite the problem in matrix form and compute the inverse of the matrix. This is obviously infeasible in WSN environment as centralized data collection and processing are needed. Inspired by the random walk interpretation of electric networks from [7, 8], we propose an iterative and localized algorithm based on random walk games to progressively estimate the potential field .

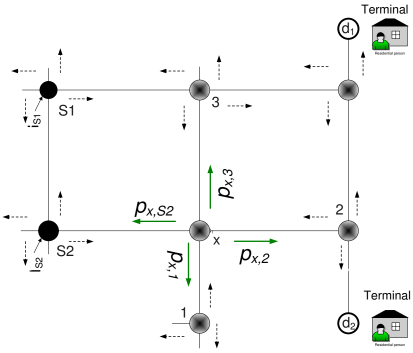


Fig. 2. Illustration of the Random Walk Game

Consider experiments of random walks in graph **G**, illustrated in Fig. 2, where every node is marked with a fixed face value $m.$ Starting from an arbitrary node x, a walker, initially with 0 money in hand, goes to one of its adjacent nodes with probability $p_{x,i}$ for $i \in Z(x)$. This walker keeps going in the same fashion inside the graph until it reaches one of the sink nodes, i.e. d_1 or d_2. Every time the walker passes one node i, it collects a fixed amount of money equal to m_i. After a large number of experiments, central limit theorem guarantees that the expectation of the total money collected by the walker, $f(x)$, converges [8] to:

$$f(x) = \sum_{i=1}^{\deg(x)} p_{x,i} f(i) + m_x \tag{7}$$

Obviously $f(i) = 0, i \in \mathbf{D}$ as a walker starting from a sink node immediately stops. After rewriting eqs. (1, 3) to yield the expression of the voltages, we observe that eq. (7) is equivalent to eqs. (1, 3) when $p_{x,i}$ and m_x are designed as follows:

$$p_{x,i} = \frac{g_{i,x}}{\sum_{k=1}^{\deg(x)} g_{k,x}} \tag{8}$$

$$m_x = \frac{I_x}{\sum_{k=1}^{\deg(x)} g_{k,x}} \tag{9}$$

This mathematical equivalence warrants that the expectation of the total money collected by a walker starting from node x converges to the potential value at node x. While this analogy provides a distributed way for individual nodes to estimate its potential value, this method is still impractical in a WSN environment because of the need for extra routing infrastructure support, due to route-back of the final collected amount, and the huge communication overhead due to required large number of experiments and long duration for each experiment.

Motivated by the Relaxation Method [9], we address the issues listed above by restricting the random walk game to be within one-hop neighborhood of the starting point and apply equation (7) iteratively on all nodes until the whole network reaches equilibrium state. With this method, only local broadcasting among adjacent nodes is needed. The overall communication overhead can be reduced through adjusting of accuracy requirement. In addition, the nature of potential field being a smooth harmonic function ensures localized effect of perturbations , which we prove it in section 3.2, that further reduces potential field maintenance overhead. The pseudo code of this algorithm is presented in Algorithm 1. This pseudo code describes how the entire network reaches global equilibrium. Each node only needs to periodically execute the steps from line 12 to line 19. Though *PotentialFieldConstruct* only specifies a uniform absolute tolerance threshold for all nodes. In practice, non-uniform relative thresholds may be used to better reflect estimation accuracies at different potential levels. Our experimental results in section 4 show that significantly low overhead can be achieved by combining a coarse-grained estimation of the entire potential field with on-demand localized potential field refinements.

Algorithm 1 PotentialFieldConstruct (**N**, **I**, $Sinks$, $tolerance$)

Require: Set of all nodes **N**, array of packet generation rate **I**, set of sink nodes $Sinks$, error tolerance $tolerance$

Ensure: The equilibrium potential field

```
Variables: $P$ nodePotentials, x nodeId,
*for* each $x \in \mathbf{N}$
  if $x \in Sinks$
    then $P[x] \leftarrow 0$
    else $P[x] \leftarrow$ arbitrary non-negative number
  endif
*endfor*
$equilibrium \leftarrow false$
*while* $(equilibrium \neq true)$
  $equilibrium \leftarrow true$
  *for* each $x \in \mathbf{N}$
    if $x \notin Sinks$
      then $newP \leftarrow$ apply eq. (7)
      else $newP \leftarrow 0$
    endif
    if $|P[x] - newP| \geq tolerance$
      then $equilibrium \leftarrow false$
    endif
    $P[x] \leftarrow newP$
  *endfor*
*endwhile*
return $P$
```

3.2 Properties

We now summarize the key properties of this algorithm that are of great value in ensuring efficient routing protocol design and executions in WSN. All formal proofs of these properties are described in [6] and omitted in this paper due to space constraint.

Properties: 1) **Convergency**:*The* PotentialFieldConstruct *algorithm, when* $tolerance \to 0$*, converges to the unique solution of the potential field estimation problem with arbitrary non-negative initial guess of potential values and in any iteration order.* 2) **Loop Free**: *Forwarding over equilibrium potential field is loop free.* 3) **No Local Minima**: *The equilibrium potential field does not have local minima in that node x has the minimum potential value among its neighbor nodes if and only if x is a sink node.* 4) **Proportional Traffic Allocation**:*Given two disjoint paths sharing only the starting and ending nodes,* $Path_1$ *and* $Path_2$*, with corresponding effective resistances* Re_1 *and* Re_2*. If* $Re_1 < Re2$*, the flow rate allocated on* $Path_1$ *will be higher than that allocated on* $Path_2$*.* 5) **Rich Links Attract More Traffic**:*When more parallel links are added to a path, higher flow rate will be allocated to that path.* 6) **Perturbation Spatial Decay**:*In a dense network, the absolute potential value changes caused by event at node x are attenuated to zero over the number of hops from node x.*

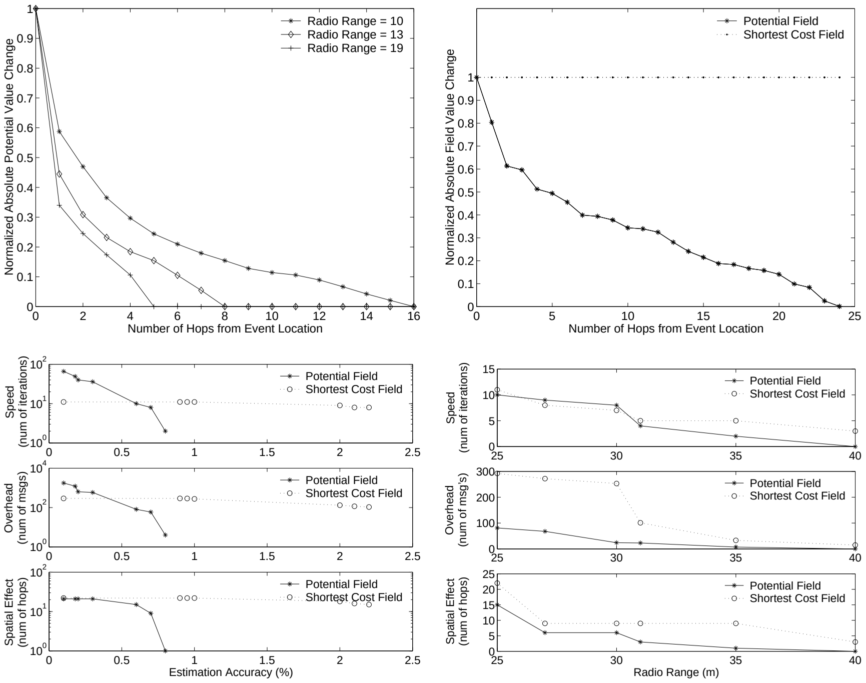


Fig. 3. Spatial Decaying of Perturbations for NS1 (a) and NS2 (b); PWave Sensitivity to Accuracy (c) and Radio Range (d)

Property 1) guarantees the correctness of the algorithm. Property 2) and 3) enable efficient protocol design in PWave as there is no need for loop-preventing mechanisms and last resort of flooding for getting packets out of a local dead-end. Property 4) comes directly from Ohm's Law and effectively guarantees that capable paths take more traffic. Property 5) comes from Rayleigh's Monotonicity Law [7] and warrants that any new nodes or links always help to take more traffic share. Property 6)is the key property that ensures the scalability and network dynamics resilience of PWave since it guarantees that only local areas of the potential field need to be updated in react to local events.

4 Experimental Evaluations

4.1 Experimental Setup and Performance Metrics

We have implemented the PWave routing algorithm and protocol in TinyOS and evaluated its performance using TOSSIM [10] simulation environment. Recognizing the asynchronous nature of PWave protocol and the lack of precise MAC timing and interference models in TOSSIM, we developed our own plug-in modules to count the protocol iteration steps.

We conducted our experiments with a network setting with grid layout, denoted as NS1 and a setting with random layout, denoted as NS2. Both layouts contain 400 nodes.

In NS1, the nodes are placed on grid locations of a (200m X 200m) square. In NS2, the nodes are randomly distributed in a (400m X 400m) square. The regular structure of NS1 enables better illustration of key concepts, while NS2 simulates a sensor layout closer to environmental monitoring applications.

We evaluated the performance of PWave in terms of convergence time, communication overhead, and locality of effects of network changes (spatial effect). The convergence time is measured by the number of iterations of the PWave algorithm until equilibrium is reached (all changes in potential are below the given threshold). The overhead is measured by the total number of messages broadcasting needed to reach equilibrium. The spatial effect of a single network change is measured by the maximum number of hops from the point of change where there is a node with potential value affected by the initial network change.

4.2 Evaluation of the PWave Protocol

Here we investigate the dynamic performance of PWave protocol in reacting to network dynamics. The other aspect of PWave's performance, the initialization of the potential field, has lesser impact on the overall performance of PWave since it is a one-time cost that can be minimized through pre-computed distribution guess and amortized through long communication session.

We first evaluate the locality of impact following a network event and then the sensitivity of PWave performance to the estimation accuracy and network density.

Locality of Perturbations (Spatial Effect) Fig. 3(a) shows the spatial decay of a 10% data rate increase at (50,50) inside network NS1 with source at (90,90), sink at (0,0) and the radio range of 13m. We observe that this change has an *exponentially spatial decay*, essential for the scalability and resilience of PWave. We also observe that the event perturbation *decays faster with larger radius range*, which corresponds to higher network density. Thus better robustness through locality can be achieved with denser deployment of senor nodes. Fig. 3(b) shows the spatial decay effect for a network event of 1/3 link cost increase around node x at (193,190) inside NS2 network with source at (389,379), sink at (32,1) and the radio range of 25m. Observe that PWave still shows exponential decay while the event in Smallest Cost Field (SCF) propagates with constant value throughout the network. This result shows the resilience of PWave to network event. Last, observe that the link const change event is general in that it can include link outages and new nodes joining the network, and thus the above observations hold for a wide range of network events.

PWave Sensitivity Using the same setting from which Fig.3(b) was generated, Fig. 3(c) shows the PWave and SCF performance for a range of relative accuracy requirements. First observe that SCF is insensitive to the relative accuracy requirement since SCF has constant event propagation. On the other hand, PWave shows a clear decrease in convergence time, communication overhead, and reduced spatial effect for higher levels of error tolerance. In particular, for relative tolerance greater than 0.6%, PWave incurs lower overhead and faster convergency than SCF, primarily due to the tighter affected region. From our experiments, this accuracy is sufficient for traffic allocation purpose. Using the same setting with a fixed relative accuracy of 0.6%, Fig. 3(d) shows that denser networks (larger radio range) have decreased convergence times, require smaller refreshing overhead, and have a smaller number of nodes impacted by a network change, in both PWave and SCF. But PWave incurs less overhead and converges much faster at higher density than SCF. We also observe that SCF's impacted area (spacial effect) is larger than PWave.

4.3 Network Lifetime Maximization

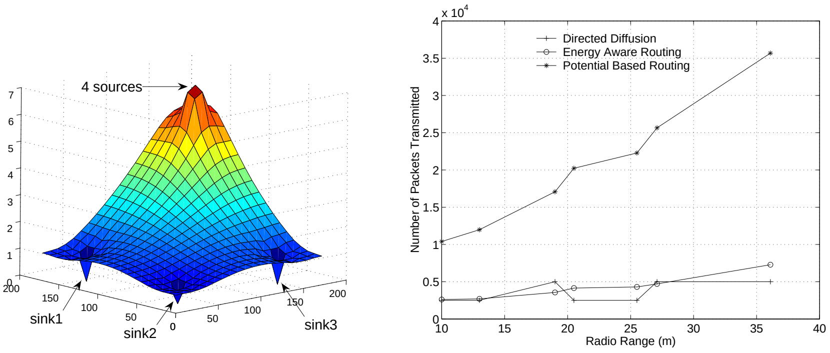


Fig. 4. (a)Estimated Potential Field; (b) Network Lifetime vs Network Density

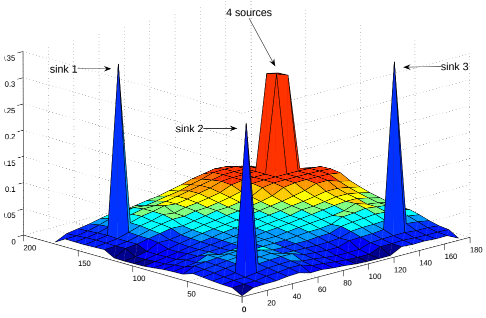


Fig. 5. Traffic Spatial Distribution

In the following we illustrate performance of PWave when applied in the context of network lifetime maximization (see section 2.2) and compare with other two state-of-the-art protocols, Directed Diffusion [2] and Energy-aware Routing [4].

Our experiments use the NS1 network setting with four source nodes located closely at (144,135), (153,135), (153,144) and (144,144). Three sink nodes are located at (27,27), (144,27) and (27,144). All nodes in network have the same amount of battery energy level initially except sink nodes and source nodes which are set to have high energy level to guarantee that source and sink nodes will not run out of battery before intermediate nodes given that the purpose of this experiment is to evaluate the efficiency of the traffic balancing obtained from PWave and its competitors. To reduce simulation time, the battery energy level of each non-sink, no-source node is set to be able to handle(either receive or transmit) at most 10,000 packets. The life time of the network is defined as the duration from the start of experiment until the first node runs out of energy.

We illustrate in Fig. 4(a) the potential field constructed by PWave under relative accuracy of 0.6% with 31 iteration rounds and 2314 message overhead, and observe its properties of monotonicity and lack of local minima. Fig. 4(b) shows that PWave achieves 2.7 to 8 times longer lifetime compared to the baseline shortest path Directed Diffusion. and up to 5 times longer lifetime over the energy-aware routing scheme. PWave achieves the major lifetime extension through better traffic balancing and take advantage of existence of multiple sinks none of the existing schemes are exploring. Fig.5 shows the traffic distribution normalized to the total traffic. Observe that less traffic is allocated on nodes that are along longer paths from sources to sinks. For example, the sink at (27,27) received less traffic than the other two sinks due to its longer distance from the sources. While all nodes around a sink received traffic allocation, the ones that face the source nodes received more allocation due to their smaller distance to sources. This experiment thus verifies that PWave achieve balanced traffic allocations with more traffic allocated on shorter paths.

5 Related Work

Well-studied network optimization is generally related to our work. The classic work in this area can be traced back to [11, 9]. All of these work formulated the optimization problem over traditional unicast/multicast communication pattern with multiple commodity flow model. PWave address a single commodity flow between multiple sources and multiple sinks with anycast communication pattern. The key mathematical difference between these two settings lies in the end-to-end flow rate constraints. Traditional mode requires that traffic (one commodity) generated by a source must be equal to the traffic received by the corresponding destination while PWave only requires the sum of the traffic generated by all sources is equal to the sum of the traffic received by all destination nodes. Many multipath routing schemes for WSNs [3–5] are related to our work. But these schemes are based on heuristic alternative paths selection without clear global optimization objective. And they did not address the anycast problem. The traffic-aware routing scheme proposed in [12] is also relevant. [12] builds local potential field (using a taut elastic membrane analogy), around shortest path as an accessory to route data around hot spots. This scheme builds on top of link-state routing protocol and aims at Internet type networks. It requires substantial message exchanges and intensive computations, thus not applicable in WSNs.

6 Conclusion and Future Work

In this paper, we presented a novel anycast routing framework that supports global optimization of custom objectives via a fully distributed, highly scalable and resilient protocol. Key properties of this framework are proved through theoretical analysis and verified through simulations. Using network lifetime maximization problem as one example, we illustrated the power of this framework by showing a 2.7 to 8 time lifetime extension over Directed Diffusion and up to 5 times lifetime extension over the energy-aware multipath routing scheme proposed in [4]. As our immediate future work, we plan to pursue systematic performance evaluation in real world settings.

References

1. Intel: Intel Heterogeneous Sensor Networks Project (2004) `http://www.intel.com/research/exploratory/heterogeneous.htm`.
2. Intanagonwiwat, C., etc.: Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. In: Proc. of ACM Mobicom'00. (2000)
3. Ganesan, D., etc.: Highly-Resilient Energy-Efficient Multipath Routing in Wireless Sensor Networks. ACM Mobile Computing and Communication Review **5**(4) (2001)
4. Shah, R.C., Rabaey, J.: Energy Aware Routing for Low Energy Ad Hoc Sensor Networks. (In: Proc. of WCNC'02)
5. Ye, F., Zhong, G., Lu, S., Zhang, L.: GRAdient Broadcast: A Robust Data Delivery Protocol for Large Scale Sensor Networks. J. of Wireless Networks **11**(3) (2005) 285–298
6. Liu, H., Zhang, Z., Srivastava, J., Firoiu, V.: PWave: Flexible Potentila-based Routing Framework for Wireless Sensor Networks. Technical report, (`http://www-users.cs.umn.edu/~hliu/pwave.pdf`)
7. Doyle, P.G., Snell, J.L.: Random Walks and Electric Networks. Mathematical Assn. of America (1984)
8. Qian, H., Nassif, S., Sapatnekar, S.: Power Grid Analysis Using Random Walks. IEEE Tran. on CAD of Integrated Circuits and Systems **24**(8) (2005) 1204–1224
9. Stern, T.: A Class of Decentralized Routing Algorithms Using Relaxation. IEEE Trans on Comm. (**25**(10))
10. Levis, P., Lee, N., Welsh, M., Culler, D.: TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications. In: Proceedings of SenSys'03. (2003) 126–137
11. Gallager, R.: A Minimum Delay Routing Algorithm Using Distributed Computation. IEEE Trans on Comm. (**25**(1))
12. Basu, A., Lin, A., Ramanathan, S.: Routing Using Potentials: A Dynamic Traffic-Aware Routing Algorithm. Proceedings of ACM SIGCOMM'03. (2003)