Technical Report

Department of Computer Science
and Engineering
University of Minnesota
4-192 EECS Building
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 06-026

PWave: Flexible Potential-based Routing Framework for Wireless Sensor Networks

Haiyang Liu, Zhi-li Zhang, Jaideep Srivastava, Victor Firoiu, and Brian Decleene

August 07, 2006

PWave: Flexible Potential-based Routing Framework for Wireless Sensor Networks

Haiyang Liu, Zhi-Li Zhang and Jaideep Srivastava
Department of Computer Science and Engineering
University of Minnesota, Minneapolis, MN 55414
{hliu, zhzhang, srivasta}@cs.umn.edu

Victor Firoiu and Brian Decleene
Advanced Information Technologies
BAE Systems, Burlington, MA 01803
{victor.firoiu, brian.decleene}@baesystems.com

***Abstract*— Inspired by the potential theory in resistive electric networks, we present a routing framework called PWave that is fundamentally different from existing routing paradigms. PWave provides a systematic framework that guarantees proportional traffic allocation and supports global optimization of custom objectives. A fully distributed and highly scalable potential estimation algorithm and protocol were designed to ensure PWave to have 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.**

I. 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 to one or a few sinks where sensor data are aggregated and sent to an (off-site) central station. Hence unlike general-purpose communication networks (whether wired or wireless) where any two nodes may serve as the two ends of an end-to-end communication and thus communication patterns can be arbitrary, communication patterns in WSNs are more confined and regular, namely from a set (or subset) of sources to one or a few (pre-specified) sinks. Moreover, the data rates of sensor nodes are either specified in queries or constrained by their physical capacities; this, combined with the knowledge of the application they are deployed to support, imply that the data generation rate of sources, although unpredictable, can be largely bounded. On the other hand, WSNs often operate in challenging or even hostile environments, with limited bandwidth and lower power, and they are subject to more frequent communication disruptions and node failures. These unique settings and constraints make routing for WSNs very different from general-purpose wired or wireless networks – in particular, the (all-pair) *shortest path* (or *minimum cost*) routing paradigm frequently used in wired (as well as in many wireless) networks, where a single shortest path between any two source-destination pair is computed for routing. In particular, to account for and accommodate the resource constraints and frequent failures/disruotions, routing for WSNs must be carefully designed and optimized, ideally with the abilities to *locally adapt* to changes in data rates, network conditions and environments.

However, many existing routing schemes (e.g., [1]) 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, as stated earlier, are typically designed using the single shortest path routing paradigm. The shortest paths are often constructed based purely on abstract graph-theoretical notions, while oblivious of traffic load or other constraints. Data-driven directed diffusion [2] provides an alternative paradigm for route construction, but it is still implicitly single shortest-path based. While offering some desirable local properties, the more recent geographic or trajectory-based routing paradigm uses "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.) Although shortest-path routing minimizes the average energy consumption per packet, it may suffer from routing instability due to network dynamics, for example, node or link failures may require network-wide re-computation of shortest paths, resulting in more routing overheads and less reliable communications. Furthermore, use of a single shortest-path often leads to hotspots that can rapidly drain the energy of individual nodes, thereby shortening the network lifetime. While several multi-path routing schemes (see, e.g., [3], [4], [5], [6]) 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. In addition, these techniques only support *single-sink* configuration, with no direct and easy extension to multi-sink configuration. However, to increase reliability and lifetime of a WSN, multiple sinks are often used where data from a source node can be routed to any of the sink nodes [7].

In this paper we present a novel and flexible routing framework that systematically allocate traffic proportional to path conductance. Drastically different from the traditional routing paradigms, this new routing framework – referred to as *PWave* – takes its inspiration from potential theories [8] and principles in resistive electric networks [8] and assigns a *potential field* to a WSN: routing is determined by the potentials of nodes in a WSN just like how electric current flows in a resistive electric network. The multiple sinks configuration is naturally addressed by assigning multiple ground nodes

in equivalent electric networks. PWave guarantees that traffic allocation is inversely proportional to the effective cost along a path based on Ohm's Law [9]. The traffic allocation in turn provides global minimization of a quadratic function based on Thompson's Law [9]. This enables flexible objectives to be optimized in PWave via different link cost design. In addition, PWave has many salient features and desirable properties that it inherits from electric networks. For example, the potential field built by PWave guarantees loop-free routing and packets are never stuck in a local dead-end (i.e. no local minima). These features enable compact and efficient protocol design with low execution overhead.

The key challenge in designing such a routing framework lies in how to build the potential field efficiently in a *distributed* fashion so that it can rapidly adapt to local changes in the network dynamics using *localized* rules. Based on the connection between the potential field and random walk [9], We take a probabilistic approach and develop a fully distributed algorithm that solves the potential field estimation problem efficiently. This algorithm scales to the density the network because only one-hop neighborhood information exchange is needed. In addition this algorithm is highly resilient to network dynamics in that local perturbations only have local effect. These features make this algorithm (and thus PWave) a suitable routing framework for WSNs.

The contributions of this paper is summarized as follows.

- *Flexible Mutlipath Routing Framework:* A novel routing framework that guarantees proportional traffic allocation and supports global optimization of custom objectives via fully localized computations. To the best of our knowledge, this is the first systematic routing framework for WSN with this capability.
- *Fully Distributed Potential Estimation Algorithm:* A novel fully distributed and highly scalable potential estimation protocol with low overhead and high resilience to network dynamics

The remainder of this paper is organized as follows. We detail the PWave framework design in section 2. The principles, properties and protocol design for the potential field are discussed in section 3. This is followed by the experimental evaluations in section 4. We describe related work in section 5 and conclude in section 6.

II. The PWave Routing Framework

In this section, we describe the motivations and details of the PWave framework and show how it supports flexible multipath routing in WSN.

A. System Model

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*, i.e., both ends of a link/edge can communicate with each other[1]. 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[2], namely, either from x to y, or from y to x. 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,

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

Given the graph **G** representing a WSN and $\mathbf{I} := \{I_x | x \in \mathbf{N}\}$, we refer to the tuple (**G**, **I**) a network configuration **NC**. Given a **NC**, routing for a WSN can be casted as a global *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. For a very general objective function $\mathcal{F}(\mathbf{G}, \mathbf{I}, \{I_{x,y} | (x,y) \in \mathbf{E}\})$, the flow allocation optimization problem can be difficult to solve, and even if it is solvable, it may not be stable, and moreover, it may not allow for an efficient distributed implementation of the solution.

[1] Although some studies have shown that communication channels between nodes can be asymmetric [1] in WSN, i.e., one node can communicate with the other node but not vice versa. The justification comes from the communication graph used in problem formulation is not necessarily a complete mimic of the physical network and asymmetric links are usually blacklisted from being used for data routing [1], [10].

[2] In other words, we ignore the acknowledgement of data traffic that flows along the other direction. In a sense, the cost for transmitting acknowledgements is implicitly accounted for in $R_{x,y}$.

B. PWave Routing Framework

1) Intuitions and Principles: Our potential-based PWave routing framework solves the flow allocation optimization problem under a natural quadratic objective function that not only always guarantees a *unique* solution, but also allows for efficient distributed solution and implementation as routing is done based on a *localized rule* as described below. 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

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

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.

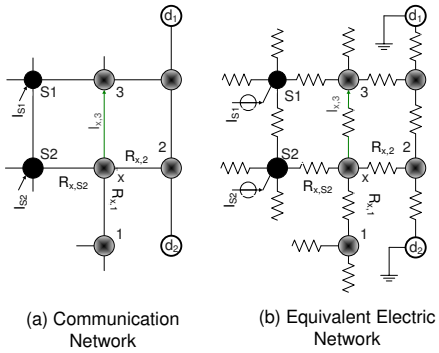

Fig. 1. Analogy between WSNs and Resistive Electric Networks.

2) Properties of PWave Routing Framework: The nice properties of the potential-based PWave routing framework follows directly from its analogy and equivalence to resistive electrical networks (see Fig. 1): if we interpret $R_{x,y}$ as resistance on edge (x, y), $I_{x,y}$ the electric current flowing through edge (x, y) and V_x the voltage at node x, eq.(3) is exactly the Ohm's law, eq.(1) exactly the Kirchhoff's law, while the boundary conditions (2) corresponding to the voltage grounding at the sinks. From the resistive electric network theory, the *energy* dissipated by a current flow $I_{x,y}$ over a wire (edge) of resistance $R_{x,y}$ is then given by $E_{x,y} = I_{x,y}^2 R_{x,y} = g_{x,y}(V_x - V_y)^2$. Then Thompson's principle from electrical theory states that the flow pattern of current within an electrical network is that which minimizes the total energy dissipation $\sum_{x,y} E_{x,y}$ – namely, the objective function in eq.(4) – over all flow patterns achieving the same total current. Or equivalently, its Lagrangian dual, Dirichlet's principle states that the voltages (i.e., potentials) taken (by nodes) within an electrical network minimize the total energy dissipation given by eq.(4).

The proof of Thompson's (or Dirichlet's) principle [9] depends nothing but on the eqs. (3), (2), and (1), using the Lagrangian method (note that the objective function in eq.(4) is a convex function). Hence the same principles apply to our potential-based PWave routing framework: the data flows routed through the network using PWave minimizes the global objective function in eq.(4). The convexity of the objective function implies that the potential field that induces the optimal flow routing and minimizes eq.(4) under the constraints eqs.(3), (2) and (1) is unique [11]. Moreover, from eqs.(1) and (3), we have

$$V_x = \sum_{y\in Z(x)} \frac{g_{x,y}}{g_x} V_y + \frac{I_x}{g_x} \text{ where } g_x = \sum_{y\in Z(x)} g_{x,y}, x \in \mathbf{N}/\mathbf{D} \tag{6}$$

In other words, for a non-sink node x, its potential is a weighted average of its neighbors' potentials, while at the sink nodes d, the potentials satisfy the boundary conditions, $V_d = 0$. Hence the potential field is a *harmonic* function that satisfies the boundary conditions in eq.(2). As a result, any local changes in edge costs or data rates would induce *localized, smooth* changes in node potentials that are eventu-

ally averaging out. These nice properties of the potential field, when constructed in a distributed manner, would lead to fast convergence, low overheads and global stability via localized actions, as we will show in the next section.

To help understand what it means to minimize the global objective function in a data network, consider a simple case where there is only one source s (with I_s as its source data rate) and one sink d, and the topology of the WSN is such that it consists exactly of K *disjoint* paths Then the flow conservation constraints become trivial except at the source, where we have $\sum_{k\in Z(s)} I_{s,k} = I_s$, and $|Z(s)| = K$. Let P_k denote each of the K disjoint paths, and $R_k = \sum_{(x,y)\in P_k} R_{x,y}$, namely, the sum of edge costs along path P_k, in other words, the path cost. Then the global objective function becomes

$$1/2 \sum_k I_k^2 R_k. \tag{7}$$

A differentiation yields that eq.(7) is minimized when $I_k R_k = I_j R_j$ for $1 \le j, k \le K$. Combining this with the constraint $\sum_k I_k = I_s$ yields that $I_k = \frac{I_s}{R_k} R_*$, where $R_* = 1/(\sum_j \frac{1}{R_j})$ Hence by minimizing the global objective function in eq.(7), the PWave routing framework yields a multi-path proportional routing scheme where the amount of data routed along each of K disjoint paths is inversely proportional to the total cost of the path. Hence the paths with the minimum costs will automatically have most amount of data routed along them. (For a more general network topology with non-disjoint paths, similar results hold – based on the analogy from electrical networks, it is possible to construct an "equivalent" network with disjoint paths only. The details are omitted, and will be included in a future version of the paper.) We see that without resort to any *ad hoc* mechanism for choosing either a minimum cost path (out of possibly many) or multiple paths, the PWave routing framework provides a more elegant and unified framework to perform (proportional) multi-path routing that relies only on a localized routing rule (i.e., eq.(3)) while achieving a network-wide global objective. Moreover, as stated earlier, changes in data rates, edge costs, or node failures would have only localized effect on the potential field that is gradually smoothed (see eq.(6)). Hence routing updates would converge fairly quickly with relatively low overhead.

3) 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 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 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:

$$R_{i,j} = \frac{1}{g_{i,j}} = (\frac{CE_i}{EN_i} + \frac{CE_j}{EN_j}) \tag{8}$$

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

III. POTENTIAL FIELD CONSTRUCTION

In this section, we tackle the potential field estimation problem that is core to the PWave framework proposed in section II. We first describe the principles behind our algorithm. The iterative algorithm is then presented with proved convergency and theoretical analysis of converging speed, followed by the protocol design. Finally properties of the potential field that are key to the PWave routing framework are presented with proofs.

A. Principles

The potential field estimation problem is governed by eq.(6), which is referred to as Kirchhoff equation in electric network theory, under boundary conditions specified in eq. (2).

Wrapping the problem expression in form of matrix yields:

$$\mathbf{M}\bar{V} = \bar{I} \tag{9}$$

where $\mathbf{M}$ is the conductance matrix, $\bar{V}$ the potential vector and $\bar{I}$ the vector of internal data generation/consumption rates. The existence and uniqueness of the solution to this problem are well known from electric network theory [9].

Traditional way of solving this problem via computing the inverse of $\mathbf{M}$ is obviously infeasible in WSN environment as centralized data collection and processing are needed. Inspired by the random walk interpretation of electric networks from [9], [11], we propose an iterative and localized algorithm based on random walk games to progressively approach the equilibrium potential field .

Consider experiments of random walks in graph $\mathbf{G}$, 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 [11] to:

$$f(x) = \sum_{i=1}^{\deg(x)} p_{x,i} f(i) + m_x \tag{10}$$

Obviously $f(i) = 0, i \in \mathbf{D}$ as a walker starting from a sink node immediately stops. Observe that eq. (10) is equivalent to eq. (6) when $p_{x,i}$ and m_x are designed as follows:

$$p_{x,i} = \frac{g_{i,x}}{\sum_{k=1}^{\deg(x)} g_{k,x}} \tag{11}$$

$$m_x = \frac{I_x}{\sum_{k=1}^{\deg(x)} g_{k,x}} \tag{12}$$

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 without the need of computing the inverse of the matrix **M**, this method is still impractical in a WSN environment for the following reasons:

- *Extra Routing Infrastructure Support:* The final amount of money collected after each random walk will have to be routed via multiple hops back to the originators from sinks. Thus an extra non-trivial routing infrastructure is needed. Besides, the low reliability of wireless communication links and high network dynamics in WSNs make packet route-back highly vulnerable.
- *Communication Overhead:* Too much communication overhead will be incurred in a large scale WSN environment where many nodes are far away from sink nodes which are the only places where random walk can stop. In addition, by the central limit theorem, a large number of random walk experiments are required for accurate estimation of a single potential value.

B. Algorithm

1) Intuitions and Pseudo Code: Motivated by the Relaxation Method [12], 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 (10) 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 III-D, 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.

2) Convergency: The correctness of this algorithm is guaranteed by the following convergency theorem.

Theorem 1 (Convergency): The *PotentialFieldConstruct* algorithm, when $tolerance \rightarrow 0$, converges to the unique solution of eq. (9) with arbitrary non-negative initial guess of potential values and in any iteration order.

The basic idea of the proof is to show that every time eq. (10) is applied, the total energy around a node is minimized thus the total *energy* of the whole network is reduced. This process continues until the whole network is in minimum energy state which corresponds to the solution due to Thompson's Law and Direchlet's Law. The detailed proof is presented in Appendix I.

3) Convergence Speed: After the proof of correctness, we now analyze the convergence speed of the algorithm in order to gain insights on factors affecting its performance.

Assuming that the potential vector $\bar{V}$ is arranged in such a way as the iterative application of eq. (10) goes in order of nodes corresponding to $\bar{V}[1], \bar{V}[2]$ ... $\bar{V}[n]$, where $n = |\bar{V}|$. Let $\mathbf{M} = L + D + U$, where L denotes the lower triangular matrix, U the upper triangular matrix and D the diagonal matrix of **M**.

We can show that the convergence speed is dictated by the eigenvalues of matrix $(-(L + D)^{-1}U)$ and the initial guess of the potential field $\bar{V}^{(0)}$. Details of derivation are in Appendix II. Both network topology and the order of the iterative estimations may affect **M**, thus $(-(L + D)^{-1}U)$. Experimental convergence speed evaluation will be discussed in section IV.

4) Practical Considerations: 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 IV 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

```
1: Variables: P nodePotentials, x nodeId,
2: for each x ∈ N
3:     if x ∈ Sinks
4:         then P[x] ← 0
5:         else P[x] ← arbitrary non-negative number
6:     endif
7: endfor
8: equilibrium ← false
9: while (equilibrium ≠ true)
10:    equilibrium ← true
11:    for each x ∈ N
12:        if x ∉ Sinks
13:            then newP ← apply eq. (10)
14:            else newP ← 0
15:        endif
16:        if |P[x] − newP| ≥ tolerance
17:            then equilibrium ← false
18:        endif
19:        P[x] ← newP
20:    endfor
21: endwhile
22: return P
```

C. PWave Protocol

We now describe the fully distributed PWave protocol that instantiate the PotentialFieldConstruct algorithm on individual nodes.

1) Messages: PWave is a broadcasting based protocol that uses only one message, referred as beacon message, for the sake of potential value advertising and existence maintenance. This message has the structure of ¡SenderID, SenderPotential, StateInfo¿. The StateInfo field is reserved for link conductance computation based on node-specific information such as energy levels. Other fields are self-explaining.

2) State Tables: Each node maintains a state table with an entry for each one-hop neighbor. The state table has

the structure of ¡NeighborID, NeighborPotential, LinkConductance, Timestamp¿ The Timestamp records when the latest beacon message from NeighborID was received in receiver's clock. This ensures that no clock synchronization is needed for the sake of state table maintenance. The LinkConductance is computed from the receiver's StateInfo and the received StateInfo from neighbor NeighborID.

3) Processes: PWave consists of the following key processes for proper operations.

- *Initialization:* All source nodes are identified by the initial discovery process using other protocols such as directed diffusion [2]. The data generation rates are assigned accordingly from queries. All sink nodes are initialized with potential value of 0. All non-sink nodes are initialized with arbitrary positive potential values.
- *State Table Maintenance:* Receiving a new beacon message triggers this process. New entries or field updates of existing entries are carried out on this table. In addition, an expiration timer periodically fires up to remove expired entries.
- *Beacon Message Broadcasting:* Beacon messages are broadcasted both periodically and asynchronously. The periodical broadcasting ensures entry refreshing, node failure and new node joing the network. Asynchronous broadcasting is triggered by large estimation discrepancies. When the difference between two estimations of the potential value is large enough (greater than a pre-set tolerance level), a beacon message will be immediately broadcasted for faster convergence.
- *Potential Re-estimation:* This process is usually triggered by the state table changes and internal packet generation rate change. Eq.(10) is applied for this process. In the case of a packet gets stuck in a non-sink node due to inaccurate potential field estimations. One time application of this equation guarantees a next hop for forwarding.

D. Properties

We now summarize the key properties of this potential field that are of great value in ensuring proper traffic allocations and efficient routing protocol design and executions.

Theorem 2 (Loop Free): Forwarding over equilibrium potential field is loop free.

Proof: The forwarding rules described in eq. (5) require that packets are always forwarded from high potential nodes to low potential nodes. Thus the potential values of each node along a path decreases monotonically when the potential field is in equilibrium state. Thus forwarding loops can not be formed. ■

Theorem 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.

Proof: We prove it through contradiction. Assume a non-sink node, x, possesses minimum potential value in its neighborhood, thus we have

$$f(x) < \sum_{i=1}^{\deg(x)} p_{x,i} f(i) \tag{13}$$

This is contradictory with the equation (10) that holds for all non-sink nodes when the potential field is in equilibrium state. We conclude the theorem holds. ■

The above two theorems enable efficient protocol design in PWave as there is no need for loop-preventing mechanisms and last resort flooding for getting packets out of a local dead-end.

Theorem 4 (High Traffic Over Low Resistance Path): 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 $Path_2$.

Proof: The proof comes directly from Ohm's Law:

$$|I_{Path_1}| = \frac{|V_{start} - V_{end}|}{\mathrm{Re}_1} > \frac{|V_{start} - V_{end}|}{\mathrm{Re}_2} = |I_{Path2}| \tag{14}$$

■

This theorem effectively guarantees the proportional traffic allocation pattern we described in section II.

Theorem 5 (Rich Links Attract More Traffic): When more parallel links are added to a path, higher flow rate will be allocated to that path.

Proof: Rayleigh's Monotonicity Law [9] states that new links always reduce the effective resistance of a path. Application of Ohm's Law immediately results in higher flow rate along that path. ■

This theorem warrants that areas with denser node deployment take more traffic loads than sparsely deployed areas in WSNs. Unlike in shortest path routing where route selection is independent of node density, PWave framework is less likely to experience hot spot problems.

Theorem 6 (Perturbation Spatial Decay): The absolute potential value changes caused by event at node x are attenuated to zero over the number of hops from node x.

The main idea is that any perturbation only has partial influence on each one of its neighbors since each neighbor usually has multiple neighbors. Thus the further away from the event center, the less effect the perturbation has. The formal proof to this theorem is detailed in Appendix III. This theorem ensures the scalability of PWave in large scale WSNs since it guarantees that only local areas of potential field need to be updated in react to local events.

IV. Experimental Evaluations

In this section we illustrate the benefits of PWare routing through simulation experiments. First, we evaluate the performance of the potential field construction protocol and compare it with commonly used Shortest Cost Field (SCF) construction protocol [2] under a set of common network dynamics. Then we evaluate an application of PWave with the objective to maximize network lifetime (described in Section II-B.3) and compare it with two popular protocols: Direct Diffusion [2] and Energy-aware routing protocol proposed in [4].

A. Experimental Setup and Performance Metrics

We have implemented the PWave routing algorithm and protocol in TinyOS and evaluated its performance using TOSSIM [13] 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. Besides resembling the regular layout type found in industrial monitoring applications [14], 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 broadcasted until 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.

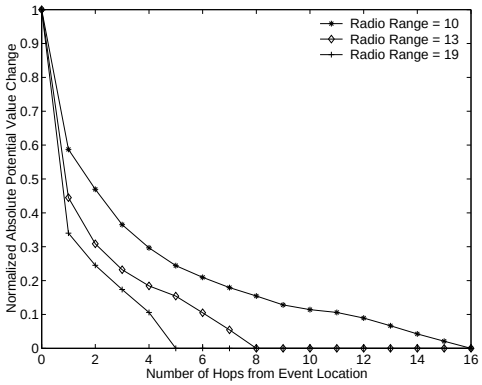

Fig. 2. Spatial Decaying of Perturbations in Grid Network

B. 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 its message exchange overhead can be reduced using similar technique proposed in [15].

In the following we evaluate the locality of impact following a change in the network and the sensitivity of PWave performance to the estimation accuracy and network density.

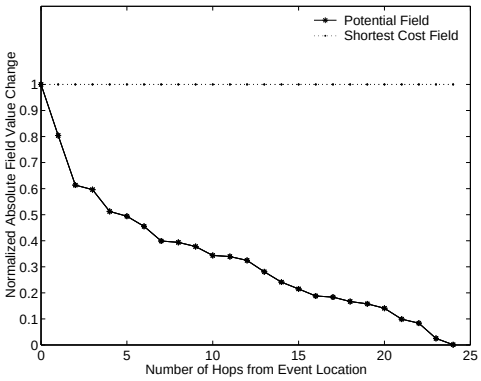

Fig. 3. Spatial Decaying of Perturbations in Random Network

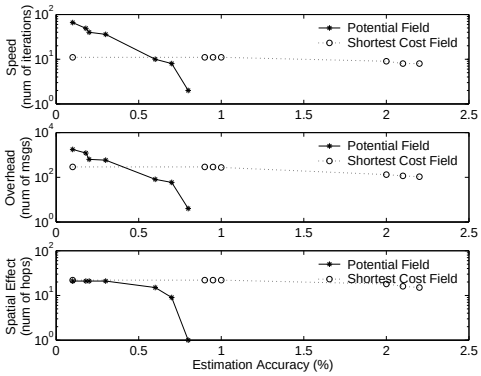

Fig. 4. Performance Effect of Relative Accuracy

1) Locality of Impact (Spatial Effect): First we evaluate the Locality of PWave protocol, a property that is key to its scalability.

Our experiments used the NS1 network setting with source node s at (81,90) and sink node d at(0,0). After the potential field is initialized at equilibrium state, we introduce a 10% increase in I_s simulating a data collection rate change event. This event directly results in a single node potential change, which subsequently propagates over the field. Fig. 2 shows the maximum absolute potential value change (normalized) over all nodes at a given distance (no. hops) from the event location. We observe that this change has an *exponentially spatial decay*, essential for the scalability of PWave. We also observe that the event perturbation *decays faster with larger radius range*, which corresponds to higher network density, where more nodes are likely to contribute to the averaging process. Thus better robustness through locality can be achieved with denser deployment of senor nodes.

Next we compare the PWave dynamic response with the SCF response. Here our experiments used the NS2 random network, where the source node s is at (389,379), the sink node d is at (32,1), and the introduced event is 1/3 link cost increase around node x at (193,190). We observe in Fig. 3 that PWave has a spatial effect decaying exponentially, while the cost change for the Smallest Cost Field (SCF) propagates with constant value throughout the network. This result shows

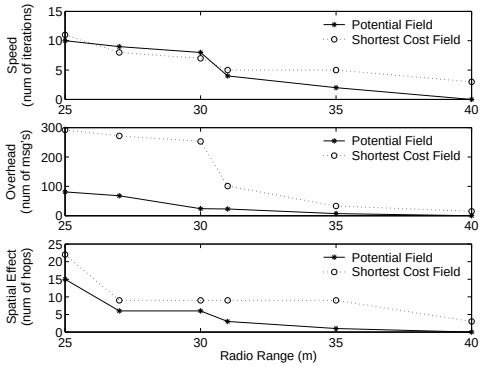

Fig. 5. Performance Effect of Network Density

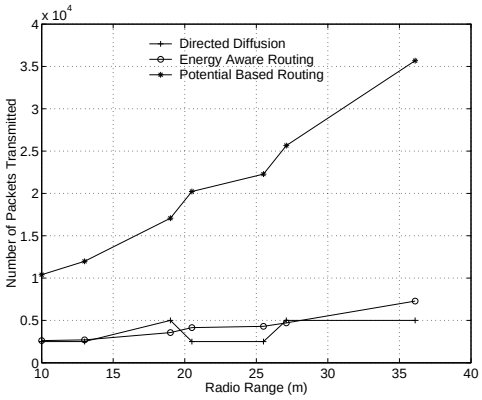

Fig. 6. Network Lifetime vs Network Density

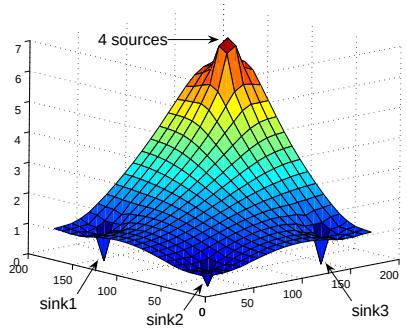

Fig. 7. 3D visualization of the Potential Fields

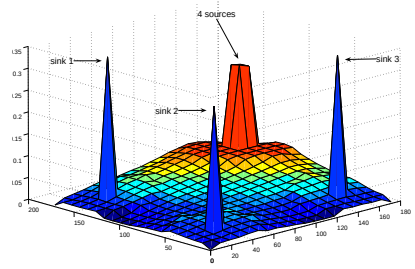

Fig. 8. Traffic Distribution for Potential Based Routing

the significant benefit of PWave in its much lower overhead following network perturbations.

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.

2) Effect of Estimation Accuracy: Next, we study the sensitivity of PWave to the threshold of estimation accuracy, that can give an indication of practical threshold levels. Here we use the same NS2 network setting and the same link cost increase event as in section IV-B.1.

Fig. 4 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. The slight drop in terms of field refreshing overhead, refreshing speed and affected regions is purely due to the larger base value in nodes further away from the event location.

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 preventing local potential minima, and thus successfully forwarding all packets to sinks.

3) Effect of Network Density: Here we evaluate the sensitivity of PWave to network density, given a fixed relative accuracy of 0.6%. Various network densities are achieved using different radio ranges and we use the same NS2 network setting and the same link cost increase event as in section IV-B.1.

We observe in Fig. 5 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.

C. Network Lifetime Maximization

So far we have shown the good scalability and robustness properties of the PWave protocol that is an implementation of a potential field over a Sensor Network. This potential field can be used with great benefit by routing protocols with a wide range of objectives. In the following we illustrate one such application that maximizes the network lifetime through multipath routing and show its superior performance compared with other two state-of-the-art protocols, Directed Diffusion 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 energy level initially except sink nodes and source nodes which are set to have high energy 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. The resistance is specified according to the design described in section II-B.3. To reduce simulation time, the 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.

First, we illustrate in Fig. 7 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.

Next, we plot in Fig. 6 the network lifetime resulting from the three protocols: PWave, Direct Diffusion and Energy-aware Routing. *We observe 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. This is a significant improvement over the state-of-the-art, possible only through the paradigm shift represented by the Potential Field model implemented by PWave.*

Both energy-aware routing and PWave routing schemes show a gain in lifetime over higher network density because both take a probabilistic approach in making routes selections. The shortest path routing employed in Directed Diffusion (DD) does not show this pattern due to its deterministic route selection. The lifetime variation in DD was purely determined by whether a unlucky node is selected by multiple routes for traffic relaying. These variations also show the inherent instability of shortest path routing.

PWave achieves the major lifetime extension through better traffic balancing and take advantage of existence of multiple sinks, which none of the existing schemes are exploring. Fig. 8 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.

V. Related Work

We have already reviewed both singlepath and multipath routing schemes for WSN in Section I. In this section, we focus our discussion on a couple of directly related recent work.

On the potential-based routing framework side, the routing scheme proposed in [16] is directly related. [16] 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.

On the methodology side, [11], [9], [17] provided most intuitions. [9] established a comprehensive connection between random walks and electric networks. [11] developed a centralized random walk based approach that is efficient in power grid analysis in integrated circuit design.

VI. Conclusion and Future Work

In this paper, we presented a novel routing framework that guarantees proportional traffic allocation and supports global optimization of custom objectives. A fully distributed and highly scalable potential estimation protocol was also presented to ensure the routing framework 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 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] A. Woo, T. Tong, and D. Culler, "Taming the Underlying Challenges of Reliable Multi-hop Routing in Sensor Networks," in *Proceedings of ACM SenSys'03*, Nov. 2003.

[2] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," in *Proceedings of ACM Mobicom'00*, June 2000.

[3] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly-Resilient Energy-Efficient Multipath Routing in Wireless Sensor Networks," *ACM Mobile Computing and Communication Review*, vol. 5, no. 4, Oct. 2001.

[4] R. C. Shah and J. Rabaey, "Energy Aware Routing for Low Energy Ad Hoc Sensor Networks," in *Proceedings of WCNC'02*, 2002.

[5] F. Ye, G. Zhong, S. Lu, and L. Zhang, "GRAdient Broadcast: A Robust Data Delivery Protocol for Large Scale Sensor Networks," *Journal of Wireless Networks*, vol. 11, no. 3, pp. 285–298, Jan. 2005.

[6] J. Gao and L. Zhang, "Load Balanced Short Path Routing in Wireless Networks," in *Proceedings of IEEE INFOCOM'04*, May 2004.

[7] "Intel Heterogeneous Sensor Networks Project," Dec. 2004, http://www.intel.com/research/exploratory/heterogeneous.htm.

[8] R. M. Bevensee, "Probabilistic Potential Theory Applied to Electrical Engineering Problems," *Proc. IEEE*, vol. 61, no. 4, p. 423437, Apr. 1999.

[9] P. G. Doyle and J. L. Snell, *Random Walks and Electric Networks*. Mathematical Assn. of America, 1984.

[10] G. Lu, N. Sadagopan, B. Krishnamachari, and A. Goel, "Delay Efficient Sleep Scheduling in Wireless Sensor Networks," in *Proceedings of IEEE INFOCOM'05*, May 2005.

[11] H. Qian, S. Nassif, and S. Sapatnekar, "Power Grid Analysis Using Random Walks," *IEEE Tran. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 8, pp. 1204–1224, May 2005.

[12] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, *Relaxation Methods for Boundary Value Problems*. Cambridge University Press, 1992.

[13] P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications," in *Proceedings of SenSys'03*, Nov. 2003, pp. 126–137.

[14] "Intel Harnesses Wireless Sensors For Chip-Equipment Care," Oct. 2003, techWeb, http://www.techweb.com/wire/26802594.

[15] F. Ye, A. Chen, S. Lu, and L. Zhang, "A Scalable Solution to Minimum Cost Forwarding in Large Sensor Networks," in *Proceedings of ICCCN'01*, May 2001.

[16] A. Basu, A. Lin, and S. Ramanathan, "Routing Using Potentials: A Dynamic Traffic-Aware Routing Algorithm," in *Proceedings of ACM SIGCOMM'03*, Aug. 2003.
[17] F. Kelly, "Network Routing," *Phil. Trans. Roy. Soc. Ser.*, vol. 337, no. 1, pp. 343–367, Apr. 1991.
[18] D. Estrin, A. Sayeed, and M. Srivastava, "Wireless Sensor Networks," Sept. 2002, tutorial at MobiCom'02.
[19] J. Park and S. Sahni, "Maximum Lifetime Broadcasting in Wireless Networks," *IEEE Trans. Computers*, vol. 54, no. 9, pp. 1081–1090, Apr. 2005.

APPENDIX I
PROOF OF THEOREM CONVERGENCY

Proof: We prove this theorem from physical laws well-known resistive electric networks given the equivalence of the potential field and the voltage distribution.

We study the energy dissipation around an arbitrary non-sink node x. Potential of a node x is the superposition of two Potential components. One component, denoted as $f'(x)$ is incurred due to the boundary condition of the internal packet generation rate I_x. the other component, denoted as $f''(x)$ is incurred due to the potentials of the neighborhood. Both V' and the energy incurred by it are fixed given a fixed I_x and $g_{x,i}$ for $i \in Z(x)$ since the flow distribution is fixed by the Ohm's law and Kirchhoff's Law described in equations (3)(1). We now show that $f''(x)$, given below,

$$f''(x) = \sum\nolimits_{i=1}^{\deg(x)} p_{x,i} f''(i) \tag{15}$$

will minimize the energy dissipation around x given by:

$$E(f''(x)) = \frac{1}{2} \sum\nolimits_{k=1}^{\deg(x)} g_{k,x} (f(k) - f''(x))^2 \tag{16}$$

$E(f''(x))$ is a convex function on f"(x). Minimization of it over variable f"(x) is achieved by making:

$$\frac{\partial E(f''(x))}{\partial (f''(x))} = 0 \tag{17}$$

Incorporation of equation (16) into equation (17) results in exactly equation (15).

Thus we have shown that every time the equation (10) is applied regardless of the order of nodes, the energy dissipation around node x is minimized regardless of the initial potential values. When $tolerance \rightarrow 0$, the iterative process will not stop until the energy around all nodes has been minimized, which put the whole network in minimal energy dissipation state or equilibrium state.

Given the uniqueness of the solution and the Dirichlet's principle and the Thomson's principle, the solution that *PotentialFieldConstruct* converges to is the unique solution of the potential field regardless of the initial guess.

■

APPENDIX II
CONVERGENCE SPEED ANALYSIS

Assuming that the potential vector $\bar{V}$ is arranged in such a way as the iterative application of eq. (10) goes in order of nodes corresponding to $\bar{V}[1]$, $\bar{V}[2]$ ... $\bar{V}[n]$, where $n = |\bar{V}|$. Let $\mathbf{M} = L + D + U$, where L denotes the lower triangular matrix, U the upper triangular matrix and D the diagonal matrix of $\mathbf{G}$.

The iterative potential estimation process can thus be expressed as:

$$(L + D)\bar{V}^{(k+1)} = I - U\bar{V}^{(k)} \tag{18}$$

where $\bar{V}^k$ stands for the potential estimation at k-th step. Rewriting equation (18) yields:

$$\begin{aligned} \bar{V}^{(k+1)} &= (L + D)^{-1}(I - U\bar{V}^{(k)}) \\ &= \{(L + D)^{-1} \sum_{j=0}^{k} (-(L + D)^{-1}U)^j\} I \\ &\quad + (-(L + D)^{-1}U)^{k+1} \bar{V}^{(0)} \end{aligned} \tag{19}$$

Thus the final solution of $\bar{V}$ is given by:

$$\begin{aligned} \bar{V}^{\infty} &= \lim_{k \rightarrow \infty} \bar{V}^{(k+1)} \\ &= \lim_{j \rightarrow \infty} (L + D)^{-1} \frac{1 - (-(L + D)^{-1}U)^j}{1 + (L + D)^{-1}U} I \\ &\quad + \lim_{j \rightarrow \infty} (-(L + D)^{-1}U)^i \bar{V}^{(0)} \end{aligned} \tag{20}$$

The convergence of the iterative process (proved in Theorem 1) guarantees that the absolute values of all the eigenvalues of matrix $(-(L + D)^{-1}U)$ are strictly less than 1. Equation (20) thus yields

$$\bar{V}^{(\infty)} = (L + D + U)^{-1} I = \bar{V} \tag{21}$$

The above analysis shows that the convergence speed depends the maximum eigenvalue of matrix $(-(L + D)^{-1}U)$ and the initial guess of the potential field $\bar{V}^{(0)}$.

APPENDIX III
PROOF OF THEOREM PERTURBATION SPATIAL DECAY

Proof: Consider the generic case of an initial potential change on a single node x. All other link/node changes are linear combinations of this type over different locations, at different times and with different values. For example, a link cost change between a and b results in initial node potential changes in both a and b and fixed topology with new costs. Thus this change can be decomposed to the linear superposition of the two individual node potential change cases.

Assume node x experienced an initial potential change $\Delta P_x^{(0)}$ in the middle of an infinite network. This is a more conservative consideration as boundary nodes tend to absorb potential change waves due to their fixed boundary conditions.

We define the maximum retaining factor as:

$$\alpha_{\max} = \frac{\max(C_i : 1 \leq i \leq \deg ree(x))}{\sum_{i=1}^{\deg ree(x)} C_i} \tag{22}$$

Note $\alpha_{\max} < 1$ if x has at least two neighbors. The degenerate case of node with only one neighbor (referred as single-link node) can be safely removed from this process because the potential change on it is exactly same as its only neighbor. In a potential change propagation path there can be at most two single-link nodes at the beginning and end of the path. Thus their existence would at most at two more hops to the propagation upper bound we derive with them removed.

Now assume that $C_{x,y} = \max(C_i : 1 \leq i \leq \deg ree(x))$. We thus have $|\Delta P_y^{(0)}| \leq \alpha |\Delta P_x^{(0)}|$ where $y \in Z(x)$. That is to say the potential changes propagated to the one-hop neighborhood of node x are bounded by $\alpha |\Delta P_x^{(0)}|$. We first bound the ultimate potential change on node x and then bound the potential changes j-hop away from x.

Recall that the initial change at x results in smaller changes in its neighbors which would reflect back changes to node x. This process goes on recursively. Let us assume the maximum retaining factor of all nodes in Z(x) to be $\alpha = \max(\alpha_i : 1 \leq i \leq \deg ree(x))$. The final potential change on node x $|\Delta P_x| = \lim_{k \to \infty} |\Delta P_x^{(k)}|$ is bounded, considering that $\alpha < 1$, by

$$\lim_{k \to \infty} |\Delta P_x^{(k)}| \leq \lim_{k \to \infty} (\sum_{i=0}^{k} \alpha^i |\Delta P_x^{(0)}|) = \frac{|\Delta P_x^{(0)}|}{1 - \alpha} \tag{23}$$

The same process can be applied on all neighbors of x, with different initial potential change. Thus we have the potential change in j-hop neighborhood of node x bounded by

$$|\Delta P_y| \leq \alpha^j |\Delta P_x| = \frac{\alpha^j |\Delta P_x^{(0)}|}{1 - \alpha} \tag{24}$$

Since plugging $\alpha < 1$ into equation (24) yields $\lim_{j \to \infty} |\Delta P_{y \in j-hop(x)}| = 0$, we conclude that the initial potential change propagation is spatially constrained, given a fixed tolerance level.

■

APPENDIX IV
EXAMPLE APPLICATION OF PWAVE FRAMEWORK – NETWORK LIFETIME MAXIMIZATION

In this appendix we demonstrate the power of the PWave framework through a comprehensive example of solving the problem of network lifetime maximization.

We adopt the definition of network lifetime defined as the time period until the first node running out of energy [4]. We adopt a homogenous energy consumption model that energy required for any node i to handle a packet (of fixed size) is constant, denoted as CE_i. This is justified since the energy used for receiving and transmission are on same level and energy cost in other operations such as computations is negligible compared to communication cost [18].

Let N denote the total number of nodes in the network. Let T_i denote the expected lifetime of node i from time 0.

Given this model, the lifetime of node i is governed by:

$$EN_i = \int_0^{T_i} (\sum_{k=1}^{\deg ree(i)} |I_{i,k}| \cdot CE_i) \cdot dt \tag{25}$$

This problem may not have unique solution and is NP-hard [19]. In order to utilize the PWave framework, we approximate $f(I) = |I|$ with the quadratic function $f(I) = I^2$, which results in the following approximation of the original lifetime model:

$$\tilde{EN_i} = \int_0^{T_i} (\sum_{k=1}^{\deg ree(i)} I_{i,k}^2 \cdot CE_i) \cdot dt \tag{26}$$

In a static design[3], maximizing the network lifetime is equivalent to keeping all expected T_i's balanced from time 0. Given a static design, traffic allocation pattern $I_{i,k}$ is time invariant. Thus Equation (26) yields

$$\frac{1}{T_i} = \frac{\sum_{k=1}^{\deg ree(i)} I_{i,k}^2 \cdot CE_i}{\tilde{EN_i}} \tag{27}$$

We design the resistance/conductance between node i and j as follow:

$$R_{i,j} = \frac{1}{g_{i,j}} = (\frac{CE_i}{\tilde{EN_i}} + \frac{CE_j}{\tilde{EN_j}}) \tag{28}$$

where $\tilde{EN_i}$ is obtained approximately from physical measurement of the energy level of node i. Plugging equation (28) into equation (4), the quadratic function PWave minimizes is given by:

$$\begin{aligned} E &= \frac{1}{2}(\sum_{i,j} I_{i,j}^2 \frac{CE_i}{\tilde{EN_i}} + \sum_{i,j} I_{j,i}^2 \frac{CE_j}{\tilde{EN_j}}) \\ &= \frac{1}{2}(\sum_i (\sum_j \frac{I_{i,j}^2 \cdot CE_i}{\tilde{EN_i}}) + \sum_j (\sum_i \frac{I_{j,i}^2 \cdot CE_j}{\tilde{EN_j}})) \\ &= \sum_{i=1}^{N} \frac{1}{T_i} \end{aligned} \tag{29}$$

Equation (29) shows that if the resistance between nodes is designed as in equation (28), traffic allocation in PWave framework would minimize equation (29). It is well known that the solution to the problem of minimization of

$$\min(\sum_{i=1}^{N} \frac{1}{T_i}) \tag{30}$$

subject to constraint

$$\sum_{i=1}^{N} T_i = const \tag{31}$$

[3]In dynamic design, we only need to periodically repeat the static design process at discrete time steps

yields $T_i = \frac{const}{N}$. The constraint given in eq. (31) is true as the total leftover energy at the design is fixed and we adopted a homogenous energy consumption model.

Thus we have shown that that if the resistance between nodes is designed as in equation (28), the balanced traffic allocation generated in PWave framework would approximate the solution to the network lifetime maximization problem.