Composable Information Gradients in Wireless Sensor Networks

Huijia Lin
Computer Science
Cornell University
huijia@cs.cornell.edu

Maohua Lu
Computer Science
SUNY Stonybrook
mlu@cs.sunysb.edu

Nikola Milosavljević
Computer Science
Stanford
nikolam@cs.stanford.edu

Jie Gao
Computer Science
SUNY Stonybrook
jgao@cs.sunysb.edu

Leonidas J. Guibas
Computer Science
Stanford
guibas@cs.stanford.edu

Abstract

In sensor networks we aim to achieve global objectives through local decisions at each node, based only on data available in the node's neighborhood. In this paper, we diffuse information away from source nodes holding desired data, so as to establish information potentials *that allow network queries to navigate towards and reach these sources through local greedy decisions, following information gradients. We compute these information potentials by solving for a discrete approximation to a partial differential equation over appropriate network neighborhoods, through a simple local iteration that can be executed in a distributed manner and can be re-invoked to repair the information field locally when links fail, sources move, etc. The solutions to this equation are classical* harmonic functions*, which have a rich algebraic structure and many useful properties, including the absence of local extrema, providing a guarantee that our local greedy navigation will not get stuck.*

Unlike shortest path trees, which can also be used to guide queries to sources, information potentials are robust to low-level link volatility as they reflect more global properties of the underlying connectivity. By exploiting the algebraic structure of harmonic functions such potentials can be combined in interesting ways to enable far greater path diversity and thus provide better load balancing than is possible with fixed tree structures, or they can be used to answer range queries about the number of sources in a certain regions by simply traversing the boundary of the region. Potentials for multiple information types can be aggregated and compressed using a variant of the q-digest data structure. The paper provides both analytic results and detailed simulations supporting these claims.

1. Introduction and Motivation

Recent advances in wireless sensor networks reveal the potential of such embedded networked systems for revolutionizing the way we observe, interact with, and influence the physical world. Early applications on distributed data collection systems have already identified the advantages of inexpensive networked sensors over more traditional centralized sensing systems. As technologies become mature and as sensor networks grow large in size and become inter-connected, we expect that sensor networks will move beyond military deployments and the monitoring of animal or other natural habitats to the places where humans work and live: homes, cars, buildings, roads, cities, etc. Note that in these human spaces a sensor network serves users embedded in the same physical space as the network, not a community of scientists remote from the observation site. Furthermore, there is often the need to deliver relevant information with very low latency, in order to allow users to act in a timely manner, as for example with first responders in disaster recovery scenarios.

In this work we explore the potential of using a network of embedded sensors to aid information discovery and navigation through a dynamic environment. This includes the navigation of packets (answering user queries from any node), as well as the navigation of physical objects (people or vehicles) moving in the same space — such as users with hand-held devices communicating with nearby sensor nodes to get *real-time* navigation information. For example, road-side sensors can monitor local traffic congestion; empty parking lots in downtown areas can be detected and tracked by sensors deployed at each parking spot. A real-time navigation system in such a dynamic environment is quite useful — for finding an empty parking spot, for guiding vehicles to road exits in an emergency, for diverting cars to alleviate and avoid traffic jams, etc. The embedded sensors serve two purposes: discovering/detecting the events of interest (e.g., a parking spot is left empty); and forming a supporting infrastructure for users to navigate towards or around and act on the detected events. In this setting, the events of interest or the destinations to which the users want to navigate to are modeled as *sources* and the users (or the nodes in which the query is generated) are modeled as *sinks*.

These emerging application scenarios have a few characteristics that differentiate them from traditional scientific monitoring applications. First, the environment can be dynamic: parking spaces are freed up or occupied over time; road conditions are changing at different periods of the day. Thus the navigation system needs to accommodate these environmental changes. Second, an event of interest may emerge anywhere in the network and a node typically does not have prior knowledge of when and where the event may appear. Third, a data source is often of the most interest to the users in its immediate neighborhood. For example, cars near a traffic jam may look for navigation suggestions to avoid the jammed area; or an empty parking space is of the most value to cars within a few blocks. Fourth, multiple queries may be arise at once seeking the same source, as in disaster recovery. Fifth, unlike scientific monitoring applications in which data is gathered to the base station for post processing at a later time, in these scenarios low latency in answering queries is a major quality-of-service requirement.

These application characteristics and new QoS requirements demand a radically different system design for information discovery and routing. Existing work has focused on infrequent queries of long duration (i.e., for streaming data). Thus information discovery phase takes a reactive approach and allows the query node to flood its interests in the network searching for relevant data [11]. Data aggregation can also be performed on the way back to the sink [19]. Little preprocessing is done; as a result information discovery may require high delay. To avoid flooding, a logical brokerage structure can be imposed in the network, enabling queries to rendez-vous with data in the network. For example, geographical hash tables [22] use a content-based hash function that maps the event type to a geographical location so that sensors near the geographical location store the data and serve as render-vous for later queries. But the separation of the logical structure from the physical structure introduces awkward triangular routing — a user may need to visit a distant rendez-vous first to learn the way to the data source, even if the latter is very close. This further exacerbates traffic bottlenecks at rendez-vous nodes holding popular data.

1.1. Overview

In this paper we explore an information diffusion scheme that maintains a potential field and establishes *information gradients* in the entire network, or appropriate neighborhoods of it, depending on the application. Hints left on sensor nodes on the existence of data sources will smoothly guide queries or mobile users towards desired sources. The construction and maintenance costs of these information potentials are justified by and amortized over the expected high frequency of queries about the data sources. As long as environmental changes occur at a slower rate than the time it takes to establish or repair these information potentials in relevant source neighborhoods, our mechanism will successfully guide queries to their destination.

Information-guided routing has been explored before as a scalable approach for settings with high query frequency [3,6,7,18,27]. Most of these gradient-based approaches [3,6,7,18] use the natural gradients of physical phenomena, since the spatial distribution of many physical quantities, e.g., temperature measurements for heat, follows a natural diffusion law. However, gradients imposed by natural laws can be far from perfect guides, as witnessed by the existence of local extrema or large plateau regions, forcing information-guided routing to deteriorate to a random walk.

The novelty of our construction is to create an artificial information potential field that is guaranteed to be free of local maxima and minima. Specifically, we mimic an information diffusion process by using *harmonic functions* [16]. A harmonic function $\Phi(x)$ defined in a domain Ω satisfies the Laplace's equation $\nabla^2\Phi(x) = 0$, familiar from the heat equation. With boundary values specified, a harmonic function is uniquely determined. In a discrete sensor network, we can specify the potential of a source node as the maximum value[1] and construct the potential field for the rest of the nodes by solving for the harmonic function. This construction is possible by a simple local iteration on the nodes, akin to gossiping with one's neighbors. Harmonic functions bring us a number of benefits, due to their nice algebraic properties, as shown in the following.

Support for local greedy routing. Most importantly, the potential field induced by the harmonic function has *no local maxima*. On each non-source node u, our discrete harmonic function Φ satisfies a condition analogous to the mean value property of continuous harmonic functions: $\Phi(u)$ is the average of the Φ values of its neighbors. From this it immediately follows that we cannot have a node with higher information strength than all of its neighbors, unless it is a source node. Thus the information gradients support an efficient local routing algorithm by simply ascending the potential field. The query messages, or the physical objects navigating with the information gradients, will in each case eventually reach the data source/destination of interest. The set of all links from each non-source node to its neighbor with the highest information strength implicitly defines a routing tree towards the source.

Aggregating coherent gradients. The rich algebraic properties of harmonic function support an efficient way to aggregate gradients for different sources. For two data types PAIDPARKINGLOT, FREEPARKINGLOT with information strength fields, Φ_P and Φ_F, respectively, we can use the summed value $\Phi_P + \Phi_F$ to guide queries that search for any PARKINGLOT — either a PAIDPARKINGLOT or a FREEPARKINGLOT. By the definition of harmonic functions, any node that does not detect PAIDPARKINGLOT or

[1] We also fix some other nodes, e.g., a few on the network boundary, as having potential 0, to enforce an information gradient throughout the network. The nodes with preassigned potentials form the Dirichlet boundary conditions for the harmonic function.

FREEPARKINGLOT cannot be a local maximum of the function $\Phi_P + \Phi_F$. Thus queries for a range of data types can be guided simply by the sum of the individual potential fields and will eventually reach a source node within the specified range. More generally, gradient aggregation can be exploited to compress the potentials and save storage space. We show how the q-digest data structure of [24] can be used to do so for any tree-structured set of information potential types.

Routing diversity and traffic balancing. Both in the case of navigation in the presence of traffic jams, as well as in the case of finding empty parking spaces, multiple queries may simultaneously ask for navigation information towards sources of the same type (freeway exits or empty parking lots). Thus it is extremely important to distribute evenly the traffic among the multiple destinations and along the paths to these data sources. If multiple queries follow the same potential field for the same source, the routing paths are likely to converge as they come near the source. This subsequently introduces load accumulation for packet routing, and traffic congestion for navigation of physical objects. But with harmonic potentials, the query from each user can choose a set of random linear coefficients λ_i and ascend the potential field $\sum_i \lambda_i \Phi(s_i)$, where $\Phi(s_i)$ is the potential field for source i. The linear combinations of harmonic functions are still harmonic, thus each query follows its 'personalized' potential field towards one of the sources. We show that this will uniformly distribute the users among different destinations, and furthermore spread out the routing paths that the users take to these destinations. Such routing diversity and traffic balancing can be appealing features for emergency evacuation.

Answering counting range queries. A counting range query asks for the number of sources inside a given (arbitrary) geographical region, such as the number of empty parting spots within a given set of blocks. With the potential field, a counting range query can be answered by simply touring the boundary of the range and summing up the difference of the potential values on the edges across the region boundary. This summed difference is precisely the number of sources in the interior of the range by the divergence-free property of harmonic gradients and Faraday's law of induction.

The information potentials are particularly suitable for sensor networks due to their inherent robustness to both environmental changes as well as wireless link dynamics and quality fluctuations. This robustness comes from a simple, gossip-style local algorithm for the potential construction, as well as from the global properties of the harmonic potentials themselves.

Distributed gradient construction The gradient construction is accomplished by the classical Jacobi iteration. The data sources fix their values at the global maximum and the rest of the nodes iterate setting their value to the average of those of their neighbors. The process stops when certain local convergence criteria are met. We remark that this construction and maintenance algorithm is completely distributed and 'blind'. A node does not need to know about environmental changes or the emergence/disappearance of data sources, thus enabling the algorithm to automatically adapt to environmental and topological changes of the network — the same reason why gossip-style algorithms are favored in dynamic networks. The construction and maintenance of the gradient field is often within a local neighborhood of the events of interest (thus reducing the total communication cost) and aims to support answering a large number of simultaneous queries in a certain region surrounding the source. Thus the construction and maintenance costs are amortized over the subsequent queries.

Robustness to low-level link variations A standard way to guide queries towards a specific node in a network is to build a shortest path tree rooted at that node — that guarantees greedy routing towards the root from any node. Trees, however, are fragile structures. A single failed link can disconnect the tree and make the root inaccessible from a large subset of nodes. As we have discussed, our potentials also define implicitly a routing tree to the source. However, each node is not committed to a single parent — rather, the node's parent is only determined when the query arrives at the node. Of course a classical shortest-path tree can also be implemented in this fashion, by giving each node a 'potential' which is its hop count distance to the source. But the real benefit of the harmonic potentials is that they can be thought of as normalized hop counts which have been smoothed via the global effects of the Jacobi iteration. Due to the discreteness in the hop count definition, link variations and node failures create many more irregularities and disturbances in hop count values than those in the harmonic strength fields. Effectively the harmonic potential creates a smooth 'mountain' with a single peak at the source; almost all nodes on this mountain side are likely to have several ascending neighbors, and thus greater capacity to reach the source. The robustness of the harmonic potentials over hop counts is supported by simulations we present later on, as well as by a theoretical analysis on link asymmetry. This trick of smoothing out the discrete hop counts by a harmonic function can also be applied in other settings where smooth vector fields of information flow need to be maintained [25].

Lastly, we note that others have also used protocols motivated by the solution to partial differential equations in sensor networks. For example, [12–14, 26] use routing based on an electrostatic potential field; but in those papers the emphasis is on network capacity and not on dynamic and efficient information discovery, the topic explored in this paper.

We introduce harmonic information potentials in Section 2, and present their main applications in Section 3. In Section 4 we describe a simple local method for computing harmonic potentials and updating them after small changes in network connectivity or source positions. Section 5 contains experimental evaluation by extensive simulation, aimed at better understanding the suitability and performance of these techniques.

2. Harmonic Information Potentials

Before the formal description of information potentials, we introduce the following terminology. The raw sensor readings are processed into high-level events, which are categorized into a set of *data types*. These data types might be chosen from a fixed universe, such as the parking spots or road exits. The nodes holding data with a particular type are called *sources*. The nodes that search for data of this type are called *sinks*. We explore in this section *information diffusion schemes* for pushing information about data sources into the network, so as to later facilitate information discovery. We establish an *information potential field*, that indicates the intensity of the diffused strength at any node, for an existing data type.

2.1. Harmonic functions

The key to our information gradient scheme is the notion of *harmonic functions*. On a domain $\Omega \subseteq R^2$, a harmonic function Φ is a real function whose continuous second partial derivative satisfies *Laplace's equation* [16]: $\nabla^2 \Phi(x, y) = 0$. If the value of the function is specified on all boundaries, referred to as *Dirichlet boundary conditions*, the solution to the Laplace's equation is unique.

A dense sensor network can be viewed as a discrete approximation of the underlying continuous geometric domain. Given certain boundary conditions, Laplace's equation in the discrete form becomes

$$\Phi(u) = \frac{1}{d(u)} \Sigma_{v \in N(u)} \Phi(v)\,,$$

where u is a node in the discrete network, $N(u)$ is the set of u's neighbors, and $d(u)$ denotes the degree of u. This naturally leads to a relaxation method for computing the discrete harmonic function Φ, namely, the *Jacobi iteration method* (also called the rubber band algorithm). Each non-boundary node performs the iteration

$$\Phi^{k+1}(u) \leftarrow \frac{1}{d(u)} \Sigma_{v \in N(u)} \Phi^k(v)\,,$$

where $\Phi^k(u)$ is the value of node u in the k-th iteration. The sources are fixed at a maximum potential value say 1. We also fix some other nodes, typically nodes on the network boundary, with information strength 0 to enforce a gradient throughout the network. The rest of the nodes perform Jacobi iterations to compute the information strength field. The Jacobi iteration method converges to the harmonic function with the pre-specified boundary values. Figure 1 gives some examples of the strength fields with different boundary conditions.

The algorithm can be intuitively understood by imagining that all the edges in the network are rubber bands. Sources or boundary nodes are pinned at their fixed values. The algorithm converges to the minimum energy state where each node is placed at the center of mass of all its neighbors. Notice that in this iterative algorithm only local neighborhood information is needed, so that the algorithm can be easily realized in a distributed sensor network.

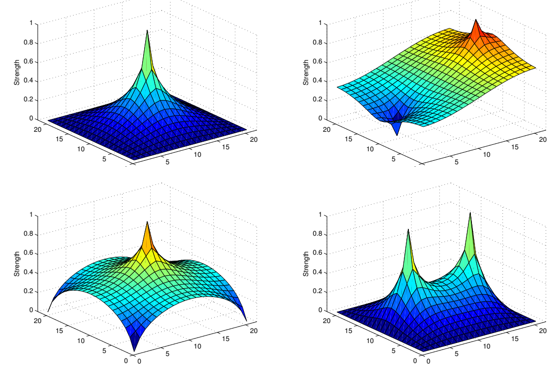

Figure 1. Examples of the potential field. The boundary conditions are specified as follows. Top left: the center node is maximum and all perimeter nodes are minima; Top right: maximum and minimum are fixed at two internal nodes, respectively; Bottom left: the center node is maximum and 4 corner nodes on perimetry are minima; Bottom right: two internal nodes are maxima and all perimeter nodes are minima.

2.2. Information potential & greedy routing

A solution to Laplace's equation has the property that the average value over a spherical surface is equal to the value at the center of the sphere (Gauss's harmonic function theorem). In other words, harmonic functions are guaranteed to be free of local minima or maxima within the solution region, also referred to as the *min-max principle*. Because of this prominent property, harmonic functions have been applied to many fields such as robot path-planning [4, 15], virtual coordinate construction in sensor networks [21] and many others. For a potential function where the goal is to find the source by local greedy routing, the min-max principle ensures that no matter where the source and the minima are located, greedy routing will succeed: starting from any node, by repeatedly ascending to the node in the neighborhood with the greatest information strength we are guaranteed to eventually hit the source.

Theoretically, the information gradient may encounter a plateau region, where all the neighbors have the same information strength. This may be due to saddle points in the harmonic function or to narrow necks such as bridges in the connectivity graph. Flat regions caused by saddle points can be easily dealt with by local discovery. By searching the local neighborhood through either a random walk or flooding, we can reach a nearby non-stationary point and continue greedy routing. Plateaus created by irregular network topology such as narrow bridges or cuts can be avoided by the placement of additional boundary nodes with minimum information strength inside the plateau regions. See also the discussion of this issue in [25].

This scheme can be easily extended to multiple sources of the same type by simply fixing the maximum information strength for all the source nodes and running the same iterative algorithm at all other nodes. Since all the maxima in the harmonic function are realized on the boundaries, a gradient path always leads to one of the sources.

2.3. Linearity of information gradients

The rich algebraic properties of harmonic functions enable a number of possibilities for aggregation and compression of coherent data types, as well as navigation in the potential field. For two data types e_1 and e_2 with information strength fields, Φ_{e_1} and Φ_{e_2}, respectively, we can use the summation $\Phi = \lambda_1\Phi_{e_1} + \lambda_2\Phi_{e_2}$ to guide queries that search for either e_1 or e_2, where λ_1, λ_2 are positive constant coefficients. It is easy to check, using the definition of harmonic function, that Φ is the harmonic function under the boundary condition $\Phi(w) = \lambda_1\Phi_{e_1}(w) + \lambda_2\Phi_{e_2}(w)$, where w is a source for e_1 or e_2. Hence Φ *cannot* have local extrema except at the source nodes for data type e_1 or e_2. In the next section we will exploit this feature to achieve routing diversity and gradient aggregation.

2.4. Robustness to low-level link variations

In practice, wireless links can be asymmetric. Thus we can model the network by a directed communication graph. Following the same distributed protocol as in the symmetric case results in a potential function whose value at any node u is the average of the values which u can *receive* — these correspond to u's *incoming* edges.

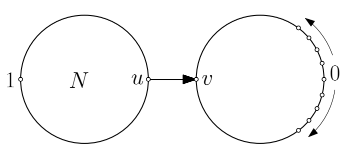

Figure 2. Trivial potential function in a large part of the network as a result of poor connectivity.

Of course, when links are directed, not every node is reachable from any other, in general. It is easy to see that this can result in potential functions which are not meaningful. For example, in Figure 2, the source cannot be reached from the boundary. As a consequence, an arbitrarily large subnetwork N may have the same potential as the source, which is trivial and useless. In other words, even though there might be an abundance of paths leading from some node in N to the source, gradient paths are not helpful in finding any of them. Notice that this would not happen if the link (u, v) were symmetric. However, we will show that our method does not require full symmetry in the links, but only a much weaker condition: bi-directional *reachability* between any two nodes, possibly along a multi-hop path.

In the ideal case of fully symmetric links, for any two nodes the shortest paths in both directions have equal lengths. Intuitively, the difference between the two shortest path lengths (the ratio of the longer one to the shorter one) is a measure of link asymmetry. The performance of our gradient-based method *degrades gracefully* with respect to this measure.

Theorem 2.1. *If the network is strongly connected, the potential function described above is unique and can be computed using the standard iterative method. Furthermore, if for any two nodes u and v the shortest path lengths from u to v and from v to u differ by at most a factor of r, then any non-source node can find a node with a potential value no higher/no lower than its value in its $\lfloor r \rfloor$-hop neighborhood.*

Proof: We define the system matrix M as follows: row i corresponds to the interior node (i.e., a non-Dirichlet boundary node) i, with the diagonal entry (i, i) equal to the in-degree of i, and the entry (i, j) equal to -1 if there is an edge from another interior node j to i, zero otherwise. This is the Laplacian matrix of the directed network, with rows and columns corresponding to the interior nodes.

The matrix-tree theorem for directed graphs (Chapter 9 of [2]) states that the determinant of M is the number of directed spanning forests (arborescences) all of whose directed trees are rooted at (i.e. their edges pointing to) the boundary nodes. Equivalently, this is the number of directed spanning trees pointing to the node obtained by contracting the boundary nodes. Because of our connectivity assumption, there exists at least one such tree. Thus M has full rank, which means the harmonic function is unique, as a solution to the linear system $Mv = 0$, with v as the vector for all interior nodes.

The convergence of the iterative method can be proved in the same manner as in the undirected case (spectral radius argument).

Finally, note that any node u has a neighbor v with a potential value no higher/no lower, such that (v, u) is an edge. As u can be reached from v in one hop, v can be reached from u in at most $\lfloor r \rfloor$ hops. This completes the proof. $\square$

3. Applications of Information Potentials

3.1. Routing diversity

Consider an emergency evacuation scenario in which many users are guided by the information gradients to building or road exits (each exit is modeled as a data source). It is important to spread out uniformly the users along the paths to these exits, to avoid traffic congestion. We abstract this scenario as multiple queries or navigation requests for the same set of sources s_i, and we would like to use local routing guidance to achieve global routing diversity and traffic balancing.

Assuming that an information potential field Φ_i has been constructed for each source (e.g., exit) s_i, we could simply let each user choose uniformly at random among the set of possible sources, and then use the potential for that source to guide the way. However, traffic tends to accumulate on the paths to the same source, as they are directed by the same gradient function. Once two navigation requests converge at one node, they are going to follow the same path from this point on.

Instead, each query j can choose some random coefficients λ_{ij} to form a 'personalized' potential field $\sum_i \lambda_{ij}\Phi_i$, where Φ_i is the potential for source i. By the linearity of information gradients, this linear combination of harmonic functions is still harmonic. Thus routing will not get stuck until it reaches a source node. However, each query is guided by a *different* potential function, thus the query routes exhibit spatial diversity and traffic load is more evenly spread out on the routes to these sources.

To better understand this feature, we consider the following scenario, in which there are k sources s_i and source i fixes its potential as $\Phi_i(s_i) = 1$, and $\Phi_i(s_j) = 0$, for $j \neq i$. All the other nodes have a potential $0 < \Phi_i < 1$. Now we form a configuration space as a k-dimensional vector space, $c(u) = (\Phi_1(u), \Phi_2(u), \cdots, \Phi_k(u))$, for each node u. The vector of coefficients for query j is $\theta_j = (\lambda_{1j}, \lambda_{2j}, \cdots, \lambda_{kj})$. The potential function Φ_j simply guides the query in the direction θ_j; the neighbor selected at each step is the node u whose potential vector maximizes $\theta_j \cdot c(u)$ It is easy to see that, by linearity of potentials, all points $c(u)$ are inside the convex hull of the source points $c(s_i)$, i.e. in the simplex spanned by $c(s_i)$, $\forall i$. In addition, the sources s_i are located in uniformly spread directions around the simplex, and thus a random direction θ_j will have equal probability to lead to any one of the sources. To summarize, as each query chooses its coefficients randomly, it will arrive at a source node with uniform probability, but the routing paths for different queries will follow their respective individual potentials. We present simulation results later to demonstrate the effectiveness of this approach in load balancing.

3.2. Potential aggregation and compression

The linearity of harmonic functions immediately enables an efficient implementation of queries for aggregated data types. As illustrated in the introduction, queries for a range of sources can simply ascend at each step to a neighbor with higher summed information potential and they will eventually reach a source node within the specified range.

In many real world scenarios an event is only of interest to the users within close proximity, i.e., the 'strength' or 'importance' of a detection is in many cases proportional to proximity of the node to the event, or the 'scale' of the event. For example, in the disaster relief scenario, an ambulance vehicle moving through the network is more likely to respond to a building collapse if it happens nearby, or if the building in question is a highrise. As the total number of events in the network may be large, nodes can simply 'forget' about the less important ones, thus saving storage for new, more important detections that may occur in the near future.

In this section we show that our approach naturally supports this notion of event importance. The idea is to have the nodes estimate the importance of an event using their local value of its potential. A potential is lumped with other small potentials, if it is smaller than some threshold. Large scale events generate a lot of sources, which can then combine their potentials. As a result, more significant events will be detected at larger distances, because the combined potential will be above the threshold further away from the sources. This kind of compression and potential aggregation save on-board storage without losing much of the navigation capability.

We use the idea of q-digest, developed by Shrivastava *et al.* [24] for answering approximate quantile queries with fixed memory requirement. In particular, suppose there are n different types of data sources that form a logical hierarchy, i.e. correspond to leaf nodes in a (balanced) binary tree. The hierarchy can be arbitrary, but many applications have a natural classification of types (e.g. big vs. small animals, dogs vs. cats, etc.). Instead of storing the n potential values separately, each node stores these potentials in a *local* q-digest data structure, constructed as follows.

Start with a binary tree describing the hierarchy of types. At each leaf node i record the potential Φ_i for source of type i. Small values will be lumped together into internal nodes of the tree so that for each node in the tree two properties will hold: (i) the value of a node is at most ε; (ii) the sum of the values of a node, its parent and sibling is at least ε. To achieve this, examine the nodes bottom up. If a node does not satisfy (ii), lump its value and the value of its sibling into the parent. The storage needed is the number of non-empty values at the nodes in this tree.

Suppose that at a sensor node u the sum of the potentials is $M(u) = \sum_{i=1}^{n} \Phi_i(u)$. If the leaf node i has a non-empty value, this value is precisely $\Phi_i(u)$, i.e., no compression is done for source i. Otherwise, its value is lumped into the value of the lowest ancestor with non-empty value and is at most ε. Denote the value of a node x in the tree as $V_u(x)$, its parent as $p(x)$, its sibling as $s(x)$, and $S_u(x)$ as the set of sources that contribute to the potential at x. To count the number of non-empty values of the digest, denoted by m, we calculate the following sum: $M = \sum_x [V_u(x) + V_u(p(x)) + V_u(s(x))] \leq 3 \sum_x V_u(x) = 3M(u)$. At the same time, $M \geq \sum_x \varepsilon = m\varepsilon$. Thus we have $m \leq 3M(u)/\varepsilon$. That is, the storage requirement at a node is only dependent on the *total* potential value, but not the number of sources present in the network.

Since the potentials are 'compressed' when they are smaller than ε, gradient routing from a node with a small potential value may need to do a local flooding until it encounters a node with a visible potential value (i.e., higher than ε), from where the standard gradient routing is adopted. Alternatively to avoid the local flooding, we can also 'decompress' the aggregated potentials by distributing uniformly the non-empty value of an internal node x to all the leaf nodes that have contributed to x, i.e., those in $S_u(x)$. Routing for a particular source can be guided by this lossily decompressed potential (and may get into a local minimum in the worse case).

This compression scheme with the q-digest can be integrated nicely with the gradient update scheme, in particular, with the Jacobi iteration for computing the harmonic values. Notice that each internal node will have its value as the sum of the potential of a subset of sources. By the linearity of harmonic functions, one can directly perform the Jacobi iteration on this bucket at a node u if at node u and all its neighbors the bucket x contains the potential values of the same subset of sources — in other words, if $S_u(x) = S_v(x)$ for all neighbors v of u. In this case, we simply take the average of the values at bucket x of u's neighbors as the new value at $V_u(x)$. In a more complicated scenario, the bucket x at node u is non-empty but the bucket x at its neighbor v is lumped into an ancestor y of x — we will let node v also keep the value at bucket x. Then the Jacobi update can be performed at node u. Symmetrically, when we update the value at bucket y of node v, we will take the sum of the values of sources in $S_v(y)$ at sensor node u, by taking the sum of appropriate buckets at node u.

To summarize, each node maintains the non-empty buckets of its own q-digest, as well as the values at the buckets corresponding to the non-empty buckets at the q-digest of its neighbors. The storage requirement is at most a constant factor more, as a non-empty bucket at a node u in the worst case causes its neighbors to also keep the value at this bucket beyond what they have already maintained in the q-digest. In a network with bounded node degree, this modification at most increases the storage requirement by a constant factor.

When we perform a Jacobi iteration, we also check the properties of q-digest to make sure they still hold. Two possibilities can happen:

- If the sum of the values $V_u(x)$ with $V_u(p(x))$ and $V_u(s(x))$ is smaller than ε, then the value of x and its sibling is lumped to the parent $p(x)$. The value at $V_u(x)$ is still maintained unless all the neighbors of u do not have non-empty bucket x.
- If the value $V_u(x)$ is larger than ε, we will need to 'demote' this bucket. The insight here is that for at least one of u's neighbors v, $\sum_{i \in V_u(x)} \Phi_i(v) > \varepsilon$ — if otherwise the new value $V_u(x)$ is the average of its neighbors' values and can not be larger than ε. Thus u has already been maintaining the buckets (lower than x) corresponding to the non-empty buckets in the q-digest at v. Thus we will demote bucket x to those non-empty buckets in its subtree.

One more advantage of this setup is that it supports ambiguous event detections. Sometimes a node cannot determine the basic type, but only a higher-level class of its detection (cannot tell a cat from a dog, but knows it is a small animal). Such a detection can then be associated with an internal node in the ontology tree of types, just like in the case of a composition of several basic potentials.

3.3. Counting range queries

A range query asks for the value of certain attributes inside a given geographical range. Previous approaches for range queries either choose to flood the region for computing the attributes of interest, or preprocess the sensor data into partial aggregates that are later assembled properly for the correct answer [8–10,23]. For the later approach, the shape of the geographical range affects the assembly cost: the more complicated the geographical range is, the more partial aggregates are to be used.

The algebraic property of harmonic function allows an efficient algorithm for counting query in an *arbitrary* range. In particular, suppose we would like to count the number of sources inside a simple closed curve (a Jordan curve). We simply tour along the curve and sum up the difference in the potential values on the edges across the region boundary. In particular, say an edge uv is crossing the region boundary with u inside the range. Then the signed difference $\Phi(u) - \Phi(v)$ is added to the sum. The summed difference will give precisely the number of sources in the interior of the range, assuming that the same signed difference, evaluated for the set of edges adjacent to each source, is equal to 1. However, the latter can be guaranteed by a simple modification to the basic Jacobi iteration described in Section 2.1 – the sources previously had their values fixed at 1, whereas now they perform the following iteration

$$\Phi^{k+1}(u) \leftarrow \frac{1}{d(u)} \left[\Sigma_{v \in N(u)} \Phi^k(v) + 1 \right] .$$

To better understand this, one can imagine that each source is equipped with an 'external' constant inflow of value 1 with one node fixed to the ground with voltage 0[2]. The signed difference $\Phi(u) - \Phi(v)$ is the electrical flow on the edge uv from interior to the exterior. If the range contains no sources, then the amount of flow entering the range must be the same as the amount of flow leaving it, by Kirchoff's current law.

Beyond its simplicity, this approach also compares favorably with other approaches for answering counting range queries, in terms of the communication cost. For example, the quad-tree/fractional cascading approach [9] incurs a query cost proportional to $O(h \log h)$ for a rectangular range with perimeter h. Our query only incurs a communication cost of h and it works for ranges of arbitrary shape.

4. Construction & Maintenance

The establishment of information strength field is achieved by on-demand Jacobi iterations. The source nodes and some boundary nodes always fix their information strength as the maximum and minimum value, respectively. A non-source node u, upon the receipt of strength values from its neighbors, takes the average of the neighbors' values, i.e., $\Phi'(u) = \sum_{v \in N(u)} \Phi(v)$. If the new strength value $\Phi'(u)$ is sufficiently different from the old strength value $\Phi(u)$, the new value is updated and broadcast to its neighbors. Otherwise, nothing is changed at node u. The update criterion can be selected to provide a tradeoff between update cost and gradient quality. We provide two basic update criteria as follows.

- **Relative difference threshold:** The update stops if the relative difference is below a threshold δ. In other words, $|\Phi'(u) - \Phi(u)| \leq \delta \cdot \max\{\Phi(u), \Phi'(u)\}$.
- **Stable relative ordering:** The update stops if the relative ordering of the strength values between u and

[2]Technically, the equation being solved in this case is the discrete *Poisson* ('nonhomogeneous Laplace') equation, where the sources are no longer boundary nodes, but nonhomogeneous terms. The theory of Jacobi method is essentially the same, however.

all neighbors of u stabilizes. In other words, for all $w \in N(u)$, $\Phi'(u) < \Phi(w)$ if and only if $\Phi(u) < \Phi(w)$.

In the relative difference threshold condition, the threshold δ bounds the relative difference of the current strength field from the harmonic function. The smaller δ we choose, the better strength field approximates the harmonic function and the higher construction cost we pay.

The stable relative ordering criterion is a more relaxed condition. In fact, the stable relative ordering is obviously sufficient to guarantee that non-source nodes do not form a local maximum or minimum of the strength values. Thus greedy routing never gets stuck at a non-source node.

The update condition can also be a combination of the stable relative ordering and the relative difference threshold conditions, so that the orderings stabilize and the relative error is below the specified threshold. The convergence condition controls the quality of the information gradients, which consequently affects the query quality and query path lengths. The convergence condition is a system parameter that can be tuned in an application specific fashion to trade preprocessing for query time.

We remark that the gradient construction and maintenance is performed in an on-demand and asynchronous way. Upon the appearance of data sources, information diffuses to the network. The closer a node is to the data sources, the less delay a node experiences in encountering a visible gradient. The delay it takes for this information diffusion usually depends on the network diameter. However, with on-demand computation, the amount of iterations each node performs, and thus the energy consumption at each node, depends mainly on the convergence condition and are relatively independent of the network size. This scalability is verified by simulations (Figure 3(iii)).

Gradient maintenance can also be triggered by user queries. Before the information gradients stabilize or when the convergence condition is too loose (e.g., δ is large in the relative difference threshold case), a user query may reach a local maximum and get stuck at a non-source node u. This may trigger further improvements of the gradients, by initiating Jacobi iterations at u (possibly with a tighter convergence condition).

5. Evaluation by Simulations

We evaluate information gradients by simulation in the following aspects: the construction and maintenance costs of the information strength field, robustness to network dynamics, the tradeoff of query qualities versus gradient maintenance cost, as well as the applications of the potential fields in Section 3.

5.1. Simulation setup

We use two sets of network topologies. One is a grid network with radio range of 1 unit and exact node degree of 4.

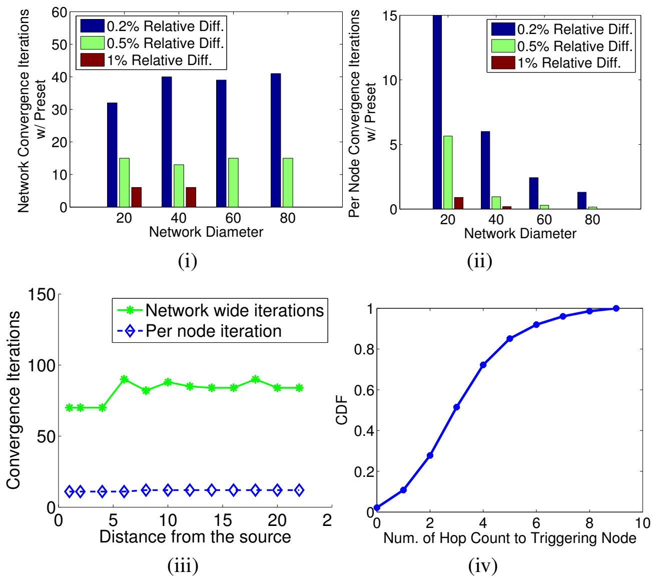

Figure 3. The boot-up phase with preset approximations: (i) The convergence rates with relative difference threshold model, $\varepsilon = 0.2\%, 0.5\%, 1\%$. (ii) The average number of iterations each node experiences. The maintenance phase: (iii) The network and per node convergence iterations for link toggles at different positions in a 50×50 grid network. (iv) The cumulative distribution function of the number of updated nodes at different distances.

The other is sensor nodes deployed uniformly randomly in a rectangular region.

We model wireless transmission using the same radio models as in TOSSIM [1, 17]: simple mode and lossy mode. In the simple mode, all nodes within the transmission range can communicate perfectly without data corruption. In the lossy mode, each link has a bit error rate reflecting the probability that a bit is flipped, according to the distance between the two communicating parties. We feed in our node locations in the TOSSIM radio model and obtain connectivity and link quality for each pair of nodes. To model link failure, at any particular time slot we also set a percentage of randomly selected links to be not available, throughout all the experiments.

The maintenance of the information strength field is on-demand. We build the gradient scheme upon a neighbor discovery protocol, which notifies the gradient maintenance component about the appearance and disappearance of neighbors. Gradient maintenance and routing are in the networking layer, and can be integrated with existing protocols that maintain a neighbor list for each node [5, 20].

5.2. Gradient construction

A critical system parameter is how fast the information strength converges. In the boot-up phase, gradients are initially set up for newly introduced data sources. In the maintenance phase afterwards, gradients are updated at link changes and repaired upon source motion.

We first study the boot-up phase behavior. In the grid topology with increasing size from 20×20 to 100×100, we fix a source at the center of the network and set the maximum value as 1000. We use a simple radio model with communication range 1. We evaluate the number of relaxation iterations (i.e., the delay) it takes for the whole network to stabilize under different convergence criteria, as shown in Figure 3 (i). The total number of iteration steps is proportional to the network diameter. We also observed that the convergence threshold ε affects the coverage scope of the potential field. Correspondingly, the convergence rates are missing in case of incomplete coverage. We also evaluate the number of Jacobi iterations each node performs. The average number of actual iterations each node experiences is much smaller, as shown in Figure 3 (ii). In these experiments we adopted a simplified pre-set scheme for the information strength field. We pre-set the strength field as a field that linearly decays from the source in the boot-up phase. The decaying amount at each step is set by as $S = \mathrm{MAX}/D$, where MAX is the maximal strength value at sources and D is the network diameter. In practice, the estimated decaying step S can be preloaded on all sensor nodes. When a node first receives a positive strength value from a neighbor, it directly sets its strength value to be the neighbor's value minus S. Standard Jacobi iterations are performed afterwards. This linear approximation is shown by simulation to be very effective. Most of the numbers are below 10 iterations. For $\varepsilon = 0.5\%, 1\%$, the average number of iterations is even less than 1. Another interesting behavior observed is that the approximation favors large networks, in the sense that the number of iterations per node decreases with network diameter. This is because both the gradient and the preset approximation functions are less steep. Thus only a few adjustments are needed.

5.3. Robustness to link dynamics

After the establishment of information gradients, the strength field is maintained to accommodate various types of dynamics. When one link toggles (appears or disappears), the gradient maintenance component will be notified with the lost (or new) neighbor and perform a Jacobi iteration. If the convergence condition is violated, gradient update will be triggered and new values are broadcast to the neighbors. We evaluate the update convergence rates and update scopes (how far the updates span). We first study whether the proximity of the link dynamics to source matters. In a 50×50 grid network, the potential field is constructed with 0.1% convergence threshold and maintained by 1% threshold. We sample nodes at different distances from the source and randomly fail one of the attached links. Figure 3 (iii) shows the number of iterations for the network to stabilize. The number of iterations per node is about 10. Figure 3 (iv) describes the cumulative distribution function about the percentage of updated nodes within a certain distance threshold. The nodes updated are all within vicinity. Nearly 90% of them are within 6 hops from the triggering node.

We also test the algorithm for scalability. By varying the diameter of the grid network from 20 to 80, we observe that both the average convergence rates (the total and per node number of iterations), as shown in Figure 4(b), and the up-

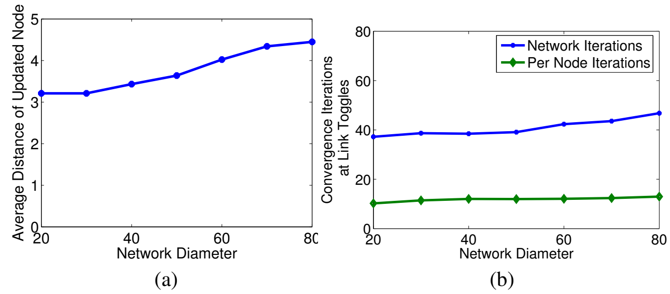

Figure 4. Gradient maintenance cost under link dynamics and source mobility. (a) the average number of hop counts of updated nodes to the triggering node with a link failure, (b) the average convergence iterations (both total and per node) for link failure reparation, All the simulations are conducted on a grid network with the simple radio

date scope, as shown in Figure 4(a) with 1% convergence threshold, do not change much when the network size increases. Thus information gradients updates for link dynamics are completely local.

5.4. Packet loss and query quality

In these experiments we incorporate the lossy wireless communication model described in section 5.1 and study the query success rate and path quality in this scenario. Under the lossy radio model, message may get corrupted, which is detected by the CRC checksum. For a corrupted gradient update message, the receiver drops this message and skip the Jacobi iteration that may be triggered. A missing Jacobi iteration can be compensated in later iterations. If a query message gets corrupted, the sender will retry. In the simulation query messages are acknowledged implicitly by overhearing retransmission from the receiver. If a sender does not hear the transmission of the query message by its neighbor after a certain period of time, it retransmits the query. In case of a second-time failure, the sender then chooses the neighbor with the second largest information strength (still higher than its own strength value though). The process repeats in case of consistent transmission failure and finally claims a query failure when there is no qualified neighbor to proceed.

In Figure 5(a), we show the query success rates under the lossy communication model initiating from increasing distance ranges. We compare it with the query success rate with the DAG formed by hop count distance to source, by using the same query routing algorithm as explained above (Figure 5(b)). In the experiments, we place a source randomly in a uniformly distributed 4000 nodes network. A potential field using 0.1% convergence threshold or a DAG according to hop count distance is constructed next. We then collect the query statistics for every non-source node. The results show that greedy routing using the gradient field is much more robust than shortest path trees. The horizontal axis in Figure 5(a) and 5(b), 'lossy model scaling factor', is a parameter used in the TOSSIM radio model that controls the loss rate. The higher this factor is, the more lossy the radio links are.

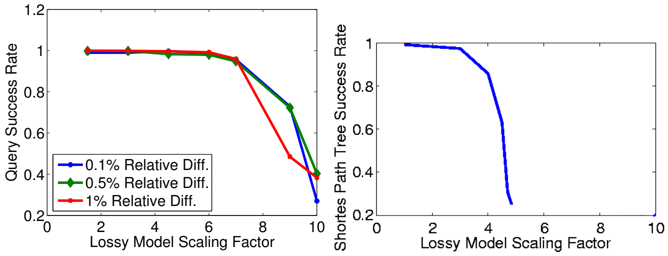

Figure 5. We place 4000 nodes uniformly randomly in a square of size 100×100. The communication range is 3. The link quality is generated with the TOSSIM lossy radio model. (a) and (b) show the query success rates of greedy routing with information gradients and a shortest path tree.

5.5. Improving routing diversity

To demonstrate the use of information gradients to achieve routing diversity, we consider a perturbed grid network of 625 nodes, with two sources in the upper-right and lower-right corners of the network. We generate 300 queries, each of them looking for any one of the 2 sources. Each query originates from one of the three nodes along the left boundary of the network (indicated by dark bars) chosen at random among the three. To guide the query message, we follow the ascending path of a function which is guaranteed to have all its local maxima at the sources. We compare three choices of such function, namely

- the potential which happens to be the strongest at the point where the query originates,
- a potential chosen uniformly at random,
- a linear combination of the potentials, with positive coefficients[3] λ and $1 - \lambda$, where λ is chosen uniformly at random from $[0, 1]$.

Figure 6 shows the results for the communication load. We see that in the third case there is a significant improvement in the nodes' communication load distribution, as long as they are not very close to the sources and query points (in the latter case high load is unavoidable).

Because we diversify our paths, we may expect to pay some penalty in terms of path stretch. Clearly, our approach provides a way to trade off path diversity for path stretch by restricting the domain from which random coefficients are drawn. However, our results show that this penalty is not too large even with the entire interval $[0, 1]$. In the above experiment we obtained the stretch value of 1.22 for the third approach, versus 1.14 and 1.15 for the first two approaches.

5.6. Potential influence zones

We tested the influence zone of aggregated gradients. In a perturbed grid network of 400 nodes, we choose two sources close to each other near the left boundary of the network. We compute their individual potentials, but store the gradients only if they are larger than ε. We want to see how many

[3] If we allowed negative coefficients, there might be local maxima at the network boundary.

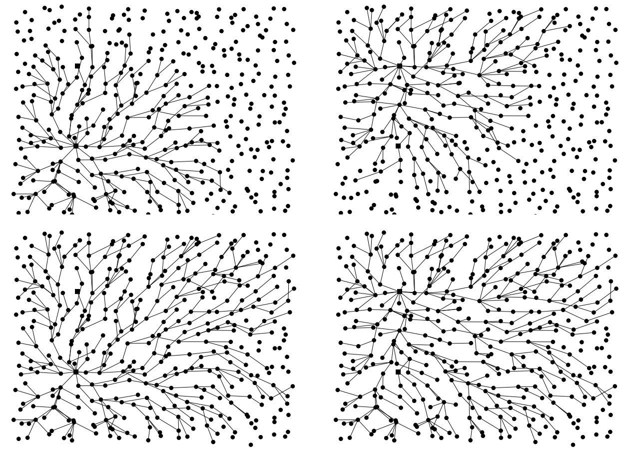

Figure 7. Nodes can adapt their potentials' zone of influence based on the importance and scale of detected events. In this example the significance threshold is $\varepsilon = 1/64$. Top row: Influence of the individual potentials. Bottom row: Influence of the combined potential.

more nodes will learn about the two events if the potentials are combined.

Figure 7 shows the result for $\varepsilon = 1/64$. In the case of individual potentials (Figure 7 top), the number of nodes unaware of the event is 182 and 183 for the lower and upper source, respectively, and in the case of the combined potential the number is only 43 (Figure 7 bottom; as expected, the number is the same in both cases).

5.7. Saving storage space using q-digests

In this section we test the idea of reducing the number of stored potential values, but in such a way to be able to reconstruct the original gradient with a guarantee on the additive error. The q-digest data structure provides a way to do this, as described in Section 3.2.

We consider the following simple example. Suppose we have a perturbed grid network of size 20×20 nodes, with two sources s_1 and s_2 near the left (resp. right) boundary, and we want to have as many nodes as possible store only one value instead of two.

If we try to compress using $\varepsilon = 0.2$, our approximate potentials (after decompression) will have some local extrema. In particular, the approximate potential of s_1 will have local maxima around s_2 and vice versa (Figure 8 top left). This is because in the region around s_2 the potential of s_2 is much stronger than that of s_1, so equal splitting results in underestimating s_2 and overestimating s_1. This effect is a lot less pronounced if we use less compression (smaller ε). Figure 8 (middle) shows the results for $\varepsilon = 0.05$. Most nodes have ascending paths to both sources even with approximate potentials, but the storage savings are also smaller. The number of nodes that decide to aggregate their potential is 380 (out of 400) for $\varepsilon = 0.2$, and 185 for $\varepsilon = 0.05$.

Finally, notice that the aggregated potential (stored in the internal node of the q-digest tree) is also a potential function whose domain is a subset of the nodes. Thus, if it is not possible to route using approximate potentials, the ag-

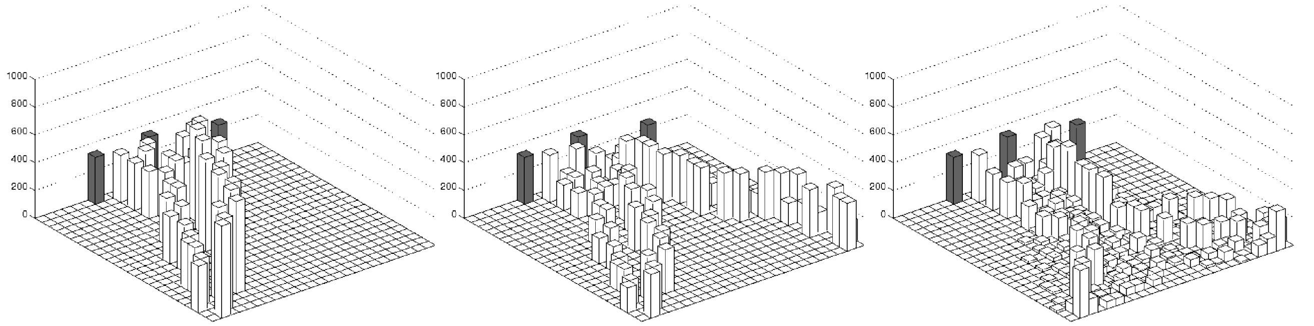

Figure 6. Path diversity results. Left: using the strongest potential at the query origin. Middle: using a randomly chosen potential. Right: using a random linear combination of potentials. Dark bars represent loads at query origins.

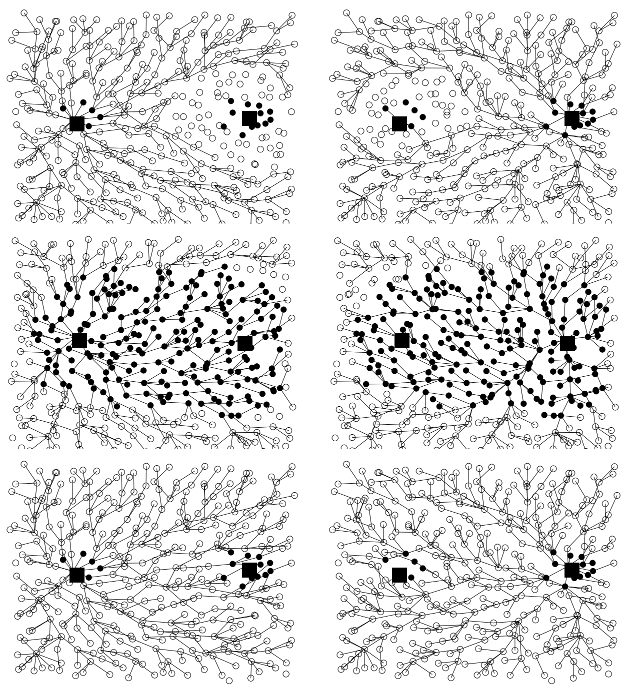

Figure 8. Compressing potential functions using q-digests. Sources are represented by black squares, empty circles represent the nodes that compress their local potentials. Top: for $\varepsilon = 0.2$, some nodes in the vicinity of s_1 have no ascending paths to s_2, and vice versa. Middle: with less compression ($\varepsilon = 0.05$) the effect is less pronounced. Bottom: using the aggregate potential (from the internal node of the q-digest tree) restores routability for all nodes even for $\varepsilon = 0.2$.

gregate potential might provide a way out of the local minimum. In our case (Figure 8 bottom) it turns out that this strategy (ascending the aggregate potential in the usual way) restores routability for all nodes.

5.8. Counting sources inside a range

In this section we evaluate how many averaging iterations (per node) are needed before the potential becomes accurate enough to be used for counting sources inside a given query region by examining only the boundary of the region (Section 3.3). We place a single source in the middle of a 25×25-node perturbed grid network. The task is to decide whether the source is inside or outside of the query region.

For any given number of iterations per node, we expect the accuracy to depend on the distance of the query region from the source. To measure this dependence accurately, we test with the query regions for which this distance is as uniform

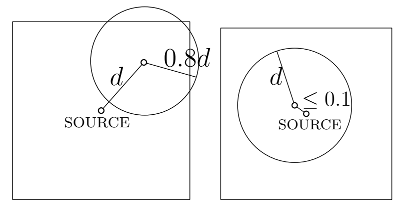

Figure 9. Two types of circular query regions in our source counting simulation, for a given distance parameter d. Regions are chosen to make d roughly equal to the average distance between the region's boundary and the source. When the region does not contain the source (left), the radius of the region is comparable to d. When the region contains the source (right), the region is displaced with respect to the source by a small distance (compared to d).

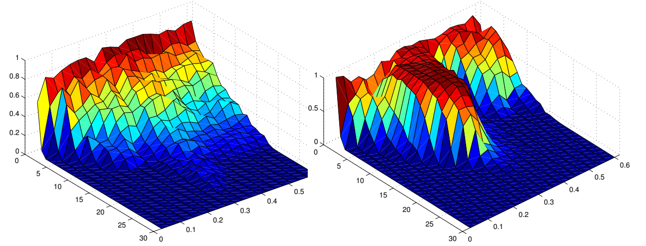

Figure 10. Source counting empirical error rate as a function of the number of Jacobi iterations and the position of the query region. Left: false positives. Right: false negatives.

as possible over the region boundary. Figure 9 shows the kind of queries that we use.

Starting from a random initial values, we simulate Jacobi iterations in small batches. After each batch of iterations, for all distance values we compute an empirical estimate of the error rate from a few hundred random trials (the random quantity being the coordinates of the center of the query region).

The results are shown in Figure 10. In all cases, the error drops to zero after only 20-30 iterations per node. In the case of false positives, convergence slows down as distance parameter d grows. We believe that this is because in this case the convergence speed is governed by the diffusion process inside the region. As the size of the region grows, it takes more time to eliminate local extrema inside the region through local averaging. In the case of false negatives, the flow has to become relatively *large*, which happens under the influence of the source and boundary. Hence the nodes roughly half way between the source and the boundary exhibit the slowest error decay; the source/boundary information needs to propagate furthest to these nodes, to make the flow in their vicinity relatively large and reliably detectable.

6. Conclusion

In this paper we have shown that harmonic information potentials, a lightweight structure that maintains and diffuses information availability, can be very helpful in guiding information flow and data-centric queries in sensor networks. The rich structure of harmonic functions allows for great flexibility and adaptability in routing algorithm designs and we expect that many future applications of these techniques will be possible.

Acknowledgements: Nikola Milosavljević and Leonidas Guibas were supported by the Army AHPCRC grant W911NF-07-2-0027-1, and NSF grants CNS-0435111, CNS-0626151. Jie Gao would like to acknowledge support though NSF CAREER Award CNS-0643687.

References

[1] TOSSIM: A simulator for TinyOS networks. User's manual in TinyOS documentation.

[2] B. Bollobas. *Modern Graph Theory*. Springer, July 1998.

[3] M. Chu, H. Haussecker, and F. Zhao. Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks. *Int'l J. High Performance Computing Applications*, 16(3):90–110, 2002.

[4] C. I. Connolly and R. A. Grupen. On the application of harmonic functions to robotics. *Journal of Robotic Systems*, 10(7):931–946, October 1993.

[5] D. Culler, P. Dutta, C. T. Ee, R. Fonseca, J. Hui, P. Levis, J. Polastre, S. Shenker, I. Stoica, G. Tolle, and J. Zhao. Towards a sensor network architecture: lowering the waistline. In *HOTOS'05: Proceedings of the 10th conference on Hot Topics in Operating Systems*, pages 24–24, Berkeley, CA, USA, 2005. USENIX Association.

[6] J. Faruque and A. Helmy. RUGGED: Routing on fingerprint gradients in sensor networks. In *IEEE Int'l Conf. on Pervasive Services (ICPS)*, pages 179–188, July 2004.

[7] J. Faruque, K. Psounis, and A. Helmy. Analysis of gradient-based routing protocols in sensor networks. In *IEEE/ACM Int'l Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 258–275, June 2005.

[8] D. Ganesan, D. Estrin, and J. Heidemann. DIMENSIONS: Why do we need a new data handling architecture for sensor networks. In *Proc. ACM SIGCOMM Workshop on Hot Topics in Networks*, pages 143–148, 2002.

[9] J. Gao, L. Guibas, J. Hershberger, and L. Zhang. Fractionally cascaded information in a sensor network. In *Proc. of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN'04)*, pages 311–319, April 2004.

[10] B. Greenstein, D. Estrin, R. Govindan, S. Ratnasamy, and S. Shenker. DIFS: A distributed index for features in sensor networks. In *Proceedings of First IEEE International Workshop on Sensor Network Protocols and Applications*, pages 163–173, Anchorage, Alaska, May 2003.

[11] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 56–67, 2000.

[12] M. Kalantari and M. Shayman. Energy efficient routing in wireless sensor networks. In *Proc. of Conference on Information Sciences and Systems*, 2004.

[13] M. Kalantari and M. Shayman. Routing in wireless ad hoc networks by analogy to electrostatic theory. In *IEEE International Conference on Communications*, 2004.

[14] M. Kalantari and M. Shayman. Design optimization of multi-sink sensor networks by analogy to electrostatic theory. In *IEEE Wireless Communications and Networking Conference*, 2006.

[15] D. Koditschek. Exact robot navigation by means of potential functions: Some topological considerations. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1–6, March 1987.

[16] S. G. Krantz. *Handbook of Complex Variables*. Birkhauser, Boston, MA, 1999.

[17] P. Levis, N. Lee, M. Welsh, and D. Culler. TOSSIM: accurate and scalable simulation of entire tinyos applications. In *SenSys '03: Proc. of the 1st ACM Conference on Embedded Networked Sensor Systems*, pages 126–137, 2003.

[18] J. Liu, F. Zhao, and D. Petrovic. Information-directed routing in ad hoc sensor networks. *IEEE Journal on Selected Areas in Communications*, 23(4):851–861, April 2005.

[19] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. TAG: a tiny aggregation service for ad-hoc sensor networks. *SIGOPS Oper. Syst. Rev.*, 36(SI):131–146, 2002.

[20] J. Polastre, J. Hui, P. Levis, J. Zhao, D. Culler, S. Shenker, and I. Stoica. A unifying link abstraction for wireless sensor networks. In *SenSys '05: Proc. of the 3rd ACM Conference on Embedded Networked Sensor Systems*, pages 76–89, November 2005.

[21] A. Rao, C. Papadimitriou, S. Shenker, and I. Stoica. Geographic routing without location information. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 96–108, 2003.

[22] S. Ratnasamy, L. Yin, F. Yu, D. Estrin, R. Govindan, B. Karp, and S. Shenker. GHT: A geographic hash table for data-centric storage. In *Proc. of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, pages 78–87, September 2002.

[23] R. Sarkar, X. Zhu, and J. Gao. Hierarchical spatial gossip for multi-resolution representations in sensor networks. In *Proc. of the International Conference on Information Processing in Sensor Networks (IPSN'07)*, pages 420–429, April 2007.

[24] N. Shrivastava, C. Buragohain, D. Agrawal, and S. Suri. Medians and beyond: new aggregation techniques for sensor networks. In *SenSys '04: Proc. of the 2nd ACM Conference on Embedded Networked Sensor Systems*, pages 239–249, 2004.

[25] P. Skraba, Q. Fang, A. Nguyen, and L. Guibas. Sweeps over wireless sensor networks. In *IPSN '06: Proceedings of the fifth international conference on Information processing in sensor networks*, pages 143–151, New York, NY, USA, 2006. ACM Press.

[26] S. Toumpis and L. Tassiulas. Packetostatics: Deployment of massively dense sensor networks as an electrostatics problem. In *Proc. 24th Conference of the IEEE Communications Society (InfoCom)*, volume 4, pages 2290–2301, 2005.

[27] F. Ye, G. Zhong, S. Lu, and L. Zhang. GRAdient broadcast: A robust data delivery protocol for large scale sensor networks. *ACM Wireless Networks (WINET)*, 11(3):285–298, 2005.