PBS: A Virtual Grid Architecture for Gradient-based Active Querying in Sensor Networks

Jabed Faruque
Department of Electrical Engineering
University of Southern California
Los Angeles, CA 90089
faruque@usc.edu

Ahmed Helmy
Department of CISE
University of Florida
Gainesville, FL 32611
helmy@ufl.edu

ABSTRACT

Every physical event diffuses its effect geographically, which results in perceivable information gradient within the proximity of the phenomenon. In this paper, we propose a novel framework for on-demand query processing that exploits this diffusion property to form a virtual grid-based querying architecture, Probe-before-Spray (PBS), for wireless sensor networks. PBS effectively divides the sensor field on-demand based on the query type and parameters in addition to the gradient spread. Also, it combines gradient routing and in-network processing for efficient and scalable querying in sensor networks. Based on PBS, we design new algorithms to process basic aggregate queries - count, sum, average, max and min, and combined queries. Through analysis, we analyze the worst-case overhead to process these queries using PBS. Also, using extensive simulations, we demonstrate that PBS helps to reduce search overhead significantly (over 30%) to process such queries while attaining accuracy over 99%.

KEYWORDS: Sensor networks, query routing, query processing, gradient-based.

1. INTRODUCTION

The fine grain environmental monitoring capability of wireless sensor networks associates the physical world with computing platforms. With the advances in sensor technology, it is possible to detect and/or measure a wide variety of physical phenomena like temperature, light, sound, radiation, humidity, chemical contamination, nitrate level in the water, etc. In real life, within the proximity of the phenomenon, every physical event leaves some fingerprints, i.e., information gradient in terms of the event's effect; e.g., fire increases temperature, chemical spill increases contamination, nuclear leakage increases radiation, so on. Moreover, most of the physical phenomena follows known diffu-

sion laws[16][17] with distance. That is, $f(d) \propto \frac{1}{d^\alpha}$, where d is the distance from the point having the maximum effect of the event, $f(d)$ is the magnitude of the event's effect, and α is the exponent of the diffusion function that depends on the type of effect and medium; e.g., for light $\alpha = 2$, and for heat $\alpha = 1$.

Using the diffusion property of an event's effect, we can determine the magnitude of the event from apart if the distance from the event is known. Conversely, for a given magnitude of an event source, we can estimate the spread (e.g., area) of the effect through known diffusion laws, empirical data or the local collaboration of sensor nodes. This property of information gradient can be utilized for query processing, especially for on-demand query processing about the event(s). In this document, we refer the area around an event source from which an event's effect can be perceived as "*the geometry of event's effect*". One of the challenges of using this diffusion property is that the gradient is not perfect in reality and may be distorted by various environmental effects. We carefully consider this fact in our design and analysis to exploit this property.

In recent years, sensor network has been viewed as a distributed database that collects the measurements of the physical world[1]. Users specify the named data they want to collect or the event of interest through application specific or declarative queries, and the infrastructure efficiently collects and processes the data within the sensor network. Typical queries can be on-demand simple one-shot, aggregate or combined queries or long-lived continuous queries.

Existing declarative query processing systems, including TinyDB[9] and Cougar[12] resolve query via a routing tree, which need to be established by initial flooding throughout the network. Also, the application specific query mechanisms, like Directed Diffusion[2], disseminate the interest hop-by-hop throughout the network similar to flooding; however, they optimize the interest forward based on query parameters, e.g., location. Random walk based mechanisms, like ACQUIRE[3], process simple one-shot and combined queries leveraging replicated data. Nevertheless,

9

all such approaches ignore the physical properties of the events of interest that can reduce significant search overhead.

Information gradient-based query mechanisms [4][5][8] exploit the diffusion patterns of events for directionality towards source(s) or node(s) that satisfy given query parameters. However, in multiple events scenario, sources can be sparsely located and create non-overlapping information gradient regions. Thus, existing information gradient-based approaches unable to explore the gradients due to all sources. Therefore, the query processing may produce only partial results.

In this study, we propose a novel framework that exploits the diffusion property to form virtual grid-based architecture, Probe-before-Spray(PBS) to process information gradient-based queries. Leveraging geographical information and the geometry of event's effect, the querier (i.e., sink) establishes a virtual grid structure in a sensor field and initiates the query in each grid cell. The grid structure of PBS uses the geometry of event's effect to introduce search scope and reduces search overhead. Here, the cell size can vary with query type. Also, PBS uses probing before actual search to identify the occurrence or the existence of event(s) and saves search overhead, especially for a region without event(s). Further, PBS overcomes the limitation of existing information gradient-based query processing approaches and explores the information gradients due to sparsely located sources.

Compared to existing grid-based architectures, the grid cells of PBS are resizable and can be variable. Also, the querier establishes the grid on-demand in the network. Further, based on the proposed PBS architecture, we have designed algorithms to compute basic *aggregate* queries - *count*, *sum*, *average*, *max* and *min*, and *combined* query, which combines multiple sub-queries by conjunction operator.

In this study, we focus our attention on the set of events where the event's effect diffuses after its occurrence. Here, we assume that the sensor nodes are able to detect the changes due to event(s). Several recent work ([18, 19]) justifies this assumption. Also, initially we assume that the surrounding region of an event source is obstacle free to diffuse the information gradient of event's effect. However, this assumption can be relaxed using local collaboration to detect such obstacles. Finally, throughout this study, we consider the approximate geometry of event's effect based on empirical results and known diffusion laws. This geometry may change with environment condition. Precise computation of the geometry of event's effect is beyond the scope of this work. However, for performance evaluation, we consider the distortion of diffusion that captures this effect in some extend. The main contributions of this paper include:

- A novel architecture, Probe-before-Spray(PBS) that forms resizable virtual grid cells to process information gradient-based queries. This reduces search overhead significantly by exploiting the geometry of event's effect and the geographical information.

- New query processing algorithms based on PBS to process aggregate queries - *count, sum, average, max* and *min*, and combined query.

- The worst-case performance analysis of the query processing algorithms as well as PBS architecture using simple analytical models.

2. RELATED WORK

In sensor networks perspective, query processing is an effort to co-design both query processing and networking subsystems to enable efficient and scalable self-organized data retrieval and in-network processing in a reliable, energy efficient and timely manner.

Among several in-network query processing systems, Directed-Diffusion[2] is pioneer work for both query dissemination mechanisms and flexible in-network processing. Protocols based on this approach describe a query by interest message. A sink node originates and disseminates the interest throughout the network by flooding. Here, query parameters (e.g., location) may control interest forwarding. However, using this approach, a region without event also receives the query.

Declarative query processing systems, like TinyDB[9] and Cougar[12], use flooding to disseminate queries in the network and collect the replies via a routing tree, where the root node usually is the user's physical location. Here, queries are parsed and optimized at user's PC and then injected into the tree-based sensor network for processing. Like Directed-Diffusion, here in-network processing can be done at leaf nodes or intermediate nodes to reduce the amount of data flow to the root.

Leveraging geographical information and the diffusion spread, i.e., the geometry of event's effect, our proposed query-processing architecture, PBS forms virtual grid on-demand in the sensor field. PBS determines the grid cell size based on query type, parameters and the diffusion pattern of the event of interest. Also, it selects one node of each cell as a virtual querier that probes the cell to check the existence of event source(s) and initiates query forwarding in the cell. Thus, PBS eliminates search overhead in cell(s) where required sources are absent as well as performs in-network query processing without flooding the whole network. It is important to mention that PBS is appropriate for a set of events, where the event's effect can be perceived within the surrounding region of an event source.

Several query systems define policies to avoid flooding for query dissemination and forward the query only to nodes that produces relevant results for a particular query. For

10

example, [10] uses semantic routing tree (SRT) to limit the query dissemination only to nodes whose readings are within a particular range. Here, each node needs to collect information about its children or subtrees. The SRT concept is analogous to the indexing of a conventional database system and suitable for less dynamic environment. However, PBS perceives the presence of event(s) through probing and avoids search where the probing fails. Also, [6] discovers querying paths for target tracking that uses an objective function to choose a node that optimizes the usefulness of sensor data and corresponding communication costs along the paths.

Model-based data acquisition scheme proposed in [11] has some similarity with our approach of using diffusion model concept. The proposed architecture in [11] combines model-based approximate query answering to optimize the data gathering. However, we use known diffusion models just to estimate the geometry of event's effect.

The use of virtual grids in PBS has certain similarity with TTDD[13] that provides scalable and efficient data delivery to multiple mobile sinks. Compared to PBS, in TTDD, the grid cell size is fixed and independent of the geometry of event's effect. Also, TTDD is not suitable for in-network query processing.

In [7], three information-driven algorithms, DAM, EBAM and EMLAM have been proposed for constructing and maintaining sensor aggregates that collectively monitor target activity in the environment. All three algorithms are used for leader election without addressing query processing and associated routing issues.

In this work, in addition to proposing the querying architecture, Probe-before-Spray (PBS), we also develop new algorithms for basic aggregate queries (*count*, *sum*, *average*, *max* and *min*) and combined query. Also, we analyze the performance of the algorithms using simple analytical models and extensive simulations.

3. OVERVIEW OF PBS ARCHITECTURE

The proposed virtual grid-based active querying architecture, PBS relies upon two foundations - (1) the geometry of event's effect, and (2) an underlying geographic routing scheme.

An event source having specific magnitude (say, X) diffuses its effect in a sensor field. Depending on the sensitivity of embedded sensors, this diffusion spreads up to a certain area. At the periphery, the recorded magnitude of the event's effect is much lower than X. However, this small magnitude of the event's effect can be regarded as an indication that the required source(s) may exist within that area, which is called "*the geometry of event's effect*" in this paper. Leveraging this key idea, a sensor node, S, having minimum sensitivity m (where, $m < X$) can establish a virtual circular contour, C, within which it can detect the pres-

ence of a source having magnitude X. In addition to known diffusion laws, this C can be determined by empirical data or local collaboration also. Now, virtual grid formation is described through Figure 1.

Consider the distance between a source (of event or having required sensor reading about event's effect) and a node, S is d. Now, according to "*the geometry of events effect*", the radius of contour C is d. For conservative estimation of d and to avoid gap or overlapping area due to circular region, here we consider the inner square (say, C_{is}) of C. Thus, the node S is able to detect the presence of a source having magnitude X within C_{is}. Here, the length of each side of C_{is} is $d\sqrt{2}$. In a two-dimensional sensor field, using C_{is} as the area of grid cell, PBS divides the specified region into grid cells as shown in Figure 1. Depending on query types, the cells size can be equal (e.g., *count, sum, average*) or variable (e.g., *max, min, combined*). For each cell, the node closest to the center of corresponding cell is considered as virtual querier, Q_v. For variable sized grid cell, the center of a cell may be already visited for a given query. So, the first unvisited node along the diagonal of the cell is selected as Q_v. These virtual queriers initiate query in the corresponding grid cells on behalf of the querier.

To initiate query in a cell, the corresponding virtual querier, Q_v, performs following two tasks.

1. *Information probing*: Q_v uses a probing phase to identify the existence of information gradient in a cell. To improve the quality of probing, Q_v also collects data about the required information gradient from its one-hop way neighbors.

2. *Query spray i.e., dissemination*: If information probing finds the required information gradient, Q_v disseminates the query either by scoped flooding or information gradient-based query dissemination mechanisms, like RUGGED[8]. The routing protocol, RUGGED uses braided multiple-path exploration and controls the instantiation of paths using a probabilistic
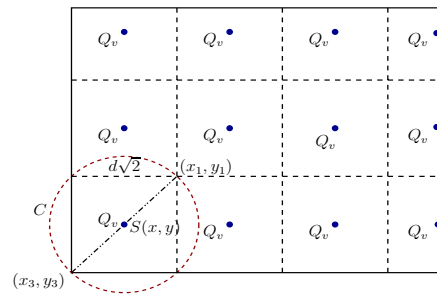

Figure 1: Virtual grid of PBS with virtual queriers, Q_v. If S is located at (x, y), then (x_1, y_1) and (x_3, y_3) are $(x + \frac{d}{\sqrt{2}}, y + \frac{d}{\sqrt{2}})$ and $(x - \frac{d}{\sqrt{2}}, y - \frac{d}{\sqrt{2}})$ respectively.

11

function. In the information gradient region, a node forwards the query greedily towards the region having the required level (according to query parameters) of information gradient, where nodes use scoped flooding to find all nodes that satisfy the given query. Here, both query parameter(s) and the boundary of the grid cell limit the scope of flooding. On the other hand, when a probing is unable to identify the required information gradient in the cell, Q_v simply forwards the query to the Q_v of the next cell.

In addition to query dissemination within grid cells, the proposed querying mechanism uses a geographical routing protocol, GPSR[14] to route the query among Q_vs and to get a reply. GPSR[14] is previously developed in literature to enable packet/query delivery to a node at a specified location. This routing mechanism has of two modes - (1) greedy-mode forwarding, and (2) perimeter-mode traversal. In greedy mode traversal, when a node receives a packet destined to a node at location (x, y), it forwards the packet to the neighbor closest to (x, y). In the absence of any such neighbor or due to existence of void in the network, the node forwards the packet using perimeter mode traversal that uses right-hand rule to get around the voids.

According to above description, PBS architecture is based on three assumptions. First, all nodes have the knowledge about "*the geometry of event's effect*". This depends on the type of application and sensing modality. As previously mentioned, in this study, we consider the approximate geometry of event's effect based on known diffusion laws (e.g., light, temperature etc.), empirical results or local collaborations. Approximate estimation of d is detailed in Section 4. Second, all nodes know the approximate geographical perimeter of the network, which may be configured at the time of deployment or using simple discovery protocol. Finally, nodes location can be determined using existing localization protocols.

Although the basic idea of PBS is simple, the main challenging part is to design energy efficient query processing algorithms for various query types. Here, we develop algorithms for aggregate queries - *count, sum, average, max, min*, and combined query using PBS architecture. Following sections detailed the approximate estimation of 'd' and the query processing algorithms.

4. APPROXIMATE ESTIMATION OF '*d*'

Approximate value of 'd' can be estimated using known diffusion laws, empirical data set or local collaboration. This section details the estimation of 'd' from known diffusion laws. [20] describes empirical data collection to identify diffusion equation and then estimate 'd'.

Consider the minimum change detection sensitivity of sensor node is m for the event of interest. Also, assume that the event's effect follows a diffusion law having diffusion parameter α. Now, consider a query to find node(s) having magnitude X. If a sensor node can measure an effect having magnitude m from d distance away from the source of interest, then the distance, d, can be expressed as

$$d = \sqrt[\alpha]{\frac{X}{m}}. \tag{1}$$

Here, the value of α may change with the change of environmental condition and the elasticity of medium. Thus, the above equation computes only the approximate geometry of event's effect.

5. QUERY PROCESSING ALGORITHM

In this section, we describe the details of new query processing algorithms that uses PBS architecture.

5.1. Aggregate query - Count, Sum and Avg

The aggregate query *Count* counts the total number of sensor nodes in the network that satisfies the given query parameters, for example, find the number of nodes in a sensor field having temperature sensor reading 80^oF or radiation $20Ci$ or more. In addition to geographical scope, here the query parameters also specify the magnitude(s) of the source(s) or event's effect (e.g., temperature, radiation, etc.) of interest. Existing querying approaches start with disseminating the query by flooding within the specified geographic scope and then use in-network aggregation for counting. When each node and/or its descendents satisfy the query, the node reports the accumulated count to its parent node. Here, initial flooding causes significant energy dissipation. The developed *Count* algorithm using PBS leverages "*the geometry of event's effect*" that corresponds to the query parameter to reduce the search overhead where the required level of sensor reading (e.g., temperature or radiation) is not available.

Sum and *Average* queries are similar to *Count* query. In these cases, in addition to counting the nodes they also aggregate sensor readings where the reading satisfies given query parameter(s). In this paper, we only describe the *Count* algorithm.

Consider a query about an event of type E having magnitude X or more. Estimate the approximate geometry of event having magnitude X using Equation (1). Here, we consider an obstacle free environment for the diffusion of event's effect, which will be relaxed later. Assume that the event's effect diffuses up to d_x and beyond that the magnitude of the events effect drops below the minimum sensitivity of sensor node, i.e., m. Now, the steps of *Count* algorithm are as follows:

1. Establish a virtual grid of cells where the area of each cell is $d_x\sqrt{2} \times d_x\sqrt{2}$, except the edge cells.
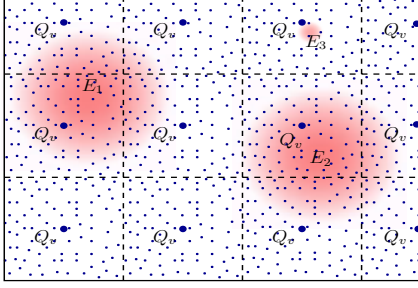
12

Figure 2: E_1,E_2 and E_3 are three events of same type. The magnitude of E_1 and E_2 are X or more, while E_3 is much smaller. Here, small dots represent sensor nodes.

2. Virtual querier, Q_v, uses information probing as described in Section 3 to find the existence of information gradient within its corresponding cell.

   (a) If the probing finds information gradient in the cell, Q_v disseminates the query in the cell as described in Section 3 to find source(s) having magnitude X or more.

   (b) Otherwise, Q_v skips the query dissemination in that cell.

   PBS continues this step for all remaining cells in a given sensor field.

In step (2a), we consider only sources having magnitude X or more. However, sources having magnitude less than X, but close to Q_v may result sufficient information gradient. So, information probing may identify information gradient and Q_v triggers query dissemination in that cell. In such a scenario, the query dissemination using scoped-flooding approach causes some extra overhead, while the information gradient-based query dissemination approach stops query forwarding after few steps. Since, information gradient-based query dissemination approach is unable to improve information level to X.

In the presence of obstacles within a cell, information gradient pattern for the diffusion may be different among nearby nodes. Through local collaboration, nodes can divide the corresponding grid cell(s) to obtain proper diffusion pattern within each portion of the cell.

5.2. Aggregate query - Max and Min

In a sensor field, aggregate query *Max* finds a node that records the maximum magnitude of the event's effect. Existing approaches collect data from all nodes and the maximum is identified at root node i.e., sink. To reduce the amount of data flow, intermediate nodes suppress non-promising responses. However, flooding-based query dissemination and collecting reply through tree (based on

child-parent relationship) causes significant transmission overhead. Using PBS architecture, we develop a new *Max* algorithm that reduces energy overhead significantly in most cases.

Consider a query to find the maximum magnitude, say $\mathcal{M}$, of an event of type E. Assume, M_x is the maximum sensing limit of sensor for the event of type E. Now, the steps to determine $\mathcal{M}$ are as follows:

1. Determine the initial value of $\mathcal{M}$ using scoped-flooding within the virtual grid cell that corresponds to M_x. Assume, $d_x\sqrt{2} \times d_x\sqrt{2}$ is the area of the cell according to Equation (1). Say, M_1 is the maximum information gradient within the cell and $M_1 \leq M_x$. Thus, the initial value of $\mathcal{M}$ is M_1. Now, assume that the area of a virtual grid cell that corresponds to the current value of $\mathcal{M}$ (i.e., M_1) is $d_1\sqrt{2} \times d_1\sqrt{2}$ according to Equation (1), where $d_1 \leq d_x$.

   This initialization step continues until some information gradient is perceived due to an event source.

2. This step is similar to the step (2) of the *Count* algorithm, except the current cell area is determined by the current value of $\mathcal{M}$. Here, $\mathcal{M}$ is non-decreasing as well as the area of the cell corresponding to $\mathcal{M}$. Now, depending on the result of information probing, Q_v has following two choices:

   (a) If Q_v perceives information gradient and according to Equation (1) the information gradient is higher than the current value of $\mathcal{M}$, Q_v disseminates the query. Using information gradient-based protocols, like RUGGED[8], Q_v finds the maximum value, say M_c, within the current cell.
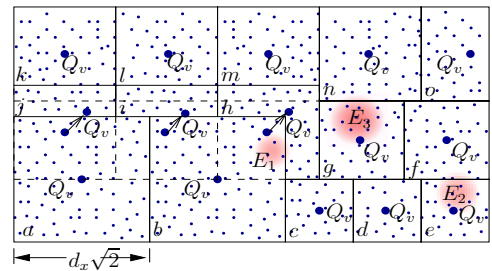

Figure 3: E_1, E_2 and E_3 are three events of same type, where $E_3 > E_2 > E_1$. Cell's number, $a, b, c, \ldots, o$ indicates the order of visit. Scoped flooding is used in cell a and then b. Information gradient is perceived in cell b that determines the size of cell c. Again, cell e and g have more powerful event sources, so cell size is increased. For h, i, and j cells, centers are already visited, so Q_v is moved diagonally to first unvisited node for those cells. Here, the result of *Max* query is the magnitude of E_3.

13

Thus, $\mathcal{M}$ can be updated as

$$\mathcal{M} = \max(\mathcal{M}, M_c).$$

This updates $\mathcal{M}$ to M_c, if $\mathcal{M} < M_c$.

(b) Otherwise, Q_v skips the query dissemination within the current cell.

Continue this step to cover the whole sensor field.

In this algorithm, the query routing between Q_vs is not simple due to the variable size of cells. Using cells of different area, the algorithm scans the sensor field horizontally from left to right and right to left and so on. To avoid any gap between the cells of two consecutive horizontal scans, the starting position of new horizontal scan is determined by the smallest cell of the most recently completed horizontal scan as shown in Figure 3. This causes some overlapping cells and also the center node of a cell, the potential virtual querier, may be visited during the previous horizontal scan. In such a scenario, the query is forwarded diagonally further from the center within the cell until an unvisited node is found, which is the virtual querier, Q_v of the cell. Finally, if no source exists in a sensor field, the overall algorithm becomes equivalent to multiple scoped-flooding at different parts of the sensor field that cover all nodes.

Using the similar steps of the above algorithm, it is also possible to design an algorithm to find an event source having minimum magnitude.

5.3. Combined Query

Combined query consists of several sub-queries that are combined by conjunction operator. In a multi-modal sensors field, sub-queries can be on different type of events having different magnitudes. Also, the corresponding diffusion patterns may follow different diffusion laws.

Consider a combined query consists of n sub-queries about n different type of events, say $E_1, E_2, \ldots, E_n$ having magnitude $X_1, X_2, \ldots, X_n$. Assume that the area of virtual cells corresponds to $X_1, X_2, \ldots, X_n$ are $A_1, A_2, \ldots, A_n$ respectively, where $A_i = 2d_{x_i}^2$, for $i = 1, 2, \ldots, n$, according to Equation (1). Thus, possible cell area set $\mathcal{A} = \{A_1, A_2, \ldots, A_n\}$. Now, using PBS architecture, the steps of combined query processing algorithm are as follows:

1. Set current cell area to $\min(\mathcal{A})$ and initiate information probing. This cell area allows Q_v to perceive the presence of information gradient due to any remaining events of interest.

2. Depending on the result of probing, Q_v chooses one of the following steps:

   (a) If probing finds information gradient, Q_v disseminates the query within the cell to find node(s) that solves some unsolved sub-queries.

(b) Otherwise, Q_v skips the query dissemination in that cell.

3. Rebuild the set $\mathcal{A}$ based on remaining sub-queries. Now, if $\mathcal{A} = \phi$, the query is successful and send the reply to the querier. On the other hand, if $\mathcal{A} \neq \phi$ and the sensor field is fully visited, the query is unsuccessful. Finally, if $\mathcal{A} \neq \phi$ and the sensor field is not fully visited, then continue from step(1).

Here the area of cells are also variable. Thus, the query dissemination between Q_vs is similar to *Max* algorithm described in Section 5.2.

6. ANALYSIS OF THE ALGORITHM

In this section, we present simple analysis to highlight the energy efficiency of PBS architecture to process the query processing algorithms mentioned in Section 5.

6.1. Assumptions

Consider a rectangular 2-D sensor field with uniformly distributed N nodes. Also, consider average neighborhood size is n_b. Thus, the energy overhead of *information probing* per virtual querier, Q_v, equals

$$C_p = n_b + 1.$$

Here, the neighbors reply the broadcast of Q_v. The collection of information about required event from neighbors in addition to Q_v reduces the effect of environmental noise as well as the distortion of diffusion.

6.2. Count Query

Consider a *Count* query about an event of type E having magnitude X or more. Now, use Equation (1) to find the area of cell and assume that there are x cells within the sensor field. For simplicity of analysis, assume that the area of all cells are equal including edge cells. Thus, each cell has $n_c = \frac{N}{x}$ nodes and there are $\sqrt{n_c}$ nodes on each side of a cell.

Let m be the number of sources in the sensor field having magnitude X or more. For simplicity, assume that the sensor nodes with reading X or more for an event are located within a same cell. Now, if $m = 1$, the probability that a cell does not have any source is $(1 - \frac{1}{x})$. Thus, for m sources, the probability of finding at least one source in a cell equals

$$P_e = 1 - \left(1 - \frac{1}{x}\right)^m \approx \left(1 - e^{-\frac{m}{x}}\right).$$

Here, $\left(1 - \frac{1}{x}\right)^m$ is the probability that the cell does not have any source.

14

To assess the worst-case energy overhead, assume that each Q_v uses scoped flooding to find the required node(s) in a cell where sensor reading is X or more. Thus, the query spray i.e., dissemination overhead per cell equals

$$C_s = n_c = \frac{N}{x},$$

since, nodes are uniformly distributed. Therefore, the energy overhead per cell equals

$$T_c = P_e \left(C_p + C_s\right) + \left(1 - P_e\right) C_p.$$

Here, the first part computes the overhead in the presence of source(s), while the other part determines the overhead if no source is available in the cell.

Q_vs use geographic routing protocol (a multi-hop routing protocol) to route the query between them. Since, all cells are square and identical, so the distance between two consecutive nodes is approximately equal to the length of cell's side. Thus, the number of transmissions require to route the query among Q_vs equals

$$T_{gr} = (x - 1)\sqrt{n_c} = (x - 1)\sqrt{\frac{N}{x}}.$$

Thus, the total energy overhead to process the *Count* query equals

$$\begin{aligned} T &= xT_c + T_{gr}, \\ &= x(n_b + 1) + N\left(1 - e^{-\frac{m}{x}}\right) + (x - 1)\sqrt{\frac{N}{x}}. \quad (2) \end{aligned}$$

This equation captures the impacts of both the number of sources in a given sensor field and the number of cells on query processing's overhead. In PBS, query parameters determines the number of cells.

Consider a sensor field of $N = 1000$ sensor nodes where the average neighborhood size, $n_b = 6$. For this sensor field, Figure 4(a) shows the query processing's overhead where the number of cells and sources vary between 1 to 100 and 1 to 20 respectively. Figure 4(a) shows that for a fixed number of sources, initially the query-processing overhead reduces with the increase of the number of cells i.e., querying for smaller values. Since, query spray overhead is higher in larger cells. Further, with the increase of number cells in a sensor field, information-probing overhead increases, but at a slower rate. The minimum overhead and corresponding optimal number cells are shown in Figure 4(b) for different number of sources in a sensor field.

6.3 Max Query

In the absence of events in a sensor field, the *Max* query algorithm performs multiple scoped flooding as discussed in
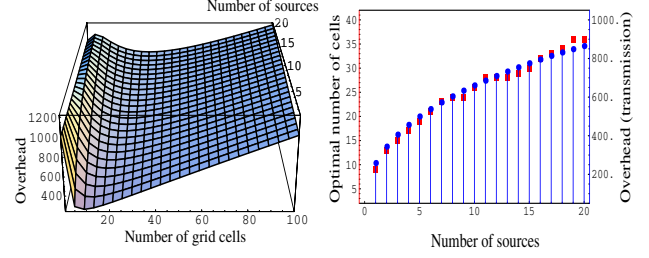
(a) Overhead for various number of sources and cells.

(b) Optimal number of cells (shown by boxes) and corresponding overhead (shown by dots) for number of sources.

Figure 4: Count query processing overhead for various number of sources and cells in the sensor field

Section 5.2. Now, considering M_x is the maximum sensing limit and using Equation (1) to find the size of grid cell, assume that there are x cells within the sensor field. Therefore, the overhead of the *Max* query-processing algorithm equals

$$T_{noevent} = N + (x - 1)\sqrt{\frac{N}{x}}.$$

Here, the first term is combined flooding overhead and the second term is the query routing overhead between Q_vs. This overhead is larger than usual flooding based approach. The overhead of the algorithm increases further if information gradient is found during initialization and later current maximum, $\mathcal{M}$, increases at every step of query processing. Since, this causes probing overhead in smaller cells in addition to query dissemination overhead. However, due to horizontal scans of the algorithm, this scenario is very unlikely to occur.

6.4 Combined Query

In the worst case scenario, events (say, n events) will be located in different cells. Thus, using scoped-flooding for query spray, the worst-case overhead of query spray equals

$$T_{s_w} = n\left(\frac{N}{x}\right).$$

Therefore, in the worst case, PBS architecture will be energy efficient over flooding-based approach for combined query processing, if

$$(n_b + 1)x + n\left(\frac{N}{x}\right) + (x - 1)\sqrt{\frac{N}{x}} \leq N.$$

Here, N is the overhead of flooding-based approach. The average overhead of this query type is detailed in [20].

15

7. SIMULATION AND PERFORMANCE

We evaluate the performance of PBS architecture for proposed query processing algorithms through extensive simulations and consider following performance metrics:

1. *Overhead* in terms of energy dissipation is the average number of transmissions required to process a query.

2. *Success ratio* is the ratio of obtained value through query algorithm over actual value. This metric is used for *Count* and *Max* queries.

3. *Absolute success probability* is the fraction of total queries when obtained value equals actual value.

7.1. Simulation Model

In our simulations, we use a $100ft \times 100ft$ uniform random grid with 10^4 sensor nodes placed at distance $1ft$ from each other. Except for the border nodes, each node is able to communicate with eight neighbors. For the simulations of *Count* and *Max* queries, we use empirical data set of [20] to emulate event source(s), where the exponent of the diffusion function i.e., α equals 2.0. For combined query, we simulate five different types of events having α equals 2.0, 1.9, 1.8, 1.7 and 1.6. For the distortion of information diffusion, we use Degree of Irregularity (DOI) and Weibull distribution with shape parameter 1.13 and scale parameter 0.28 similar to [15].

Both actual event(s) and small noisy events are uniformly distributed in sensor field. Here, small events are unable to solve queries but may cause false positive results at probing phase. Also, consider lossy wireless links and ARQ is used only for information-probing. For query spray, both scope-flooding and information gradient-based routing are used. Information gradient-based routing as specified in [8] uses a probabilistic diffusion function with exponent β for probabilistic forwarding, i.e., $p_j = f(j) = \frac{1}{j^\beta}$, where j is

the hop count in the gradient region.

The performance of query processing using PBS depends on information probing and query spray. For query spray, scoped flooding is more robust as well as causes more energy overhead than information gradient-based routing. Thus, the robustness and energy efficiency achieved using scoped flooding mainly represents the effectiveness of probing.

7.2. Count Query

In our simulations, the *success ratio* of *Count* query is over 99% using scoped flooding as shown in Fig.5(a). For small query value, occasionally probing may fail due to noise (i.e., higher DOI). The probing quality can be improved further by collecting information from neighbors more than one hop away from virtual querier. Using information gradient-based routing, the success ratio drops for large query values and in the presence of more events. Cell area is large for large query values; so gradient-based routing may unable to find all nodes that satisfy the query in the presence of noise and lossy wireless links. However, using smaller value of β as shown in [8], the success ratio can be improved further. Similarly, Fig.5(b) shows that absolute success probability is high when scoped flooding is used for spray.

In the presence of no events, the overhead of PBS is only 20% as shown in Fig.6(a). However, with the increase of number of events, the overhead increases as more nodes can satisfy the query and require more transmissions to find them. Scoped flooding is unable to stop query forwarding if probing result is false positive and causes more overhead.

7.3. Max Query

Fig.7(a) shows the success ratio of *Max* query is over 99% i.e., obtained maximum is very close to actual maximum in our simulations even in the presence of lossy wireless links and distortion.
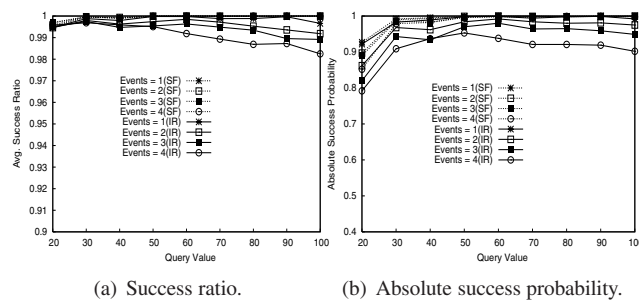

(a) Success ratio. (b) Absolute success probability.

Figure 5: *Count* query using Scoped-flooding (SF) and Information gradient-based routing (IR) with $\beta = 0.7$. Here, $DOI = 0.05$ and *Pr(link loss)* $= 0.1$.

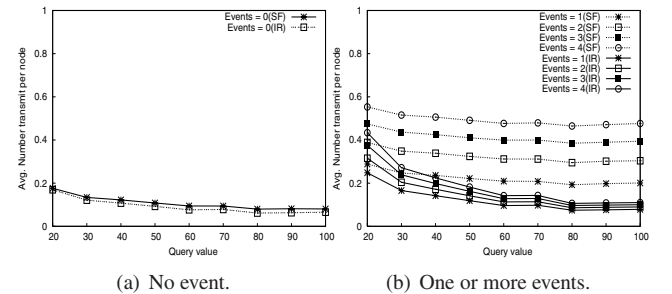(a) No event. (b) One or more events.

Figure 6: (Normalized) Overhead of *Count* query using scoped-flooding (SF) and Information gradient-based routing (IR) with $\beta = 0.7$. Here, $DOI = 0.05$ and *Pr(link loss)* $= 0.1$.

16

We notice that the overhead of query processing decreases with the increase of number of events as shown in Fig.7(b).

(a) Success ratio.
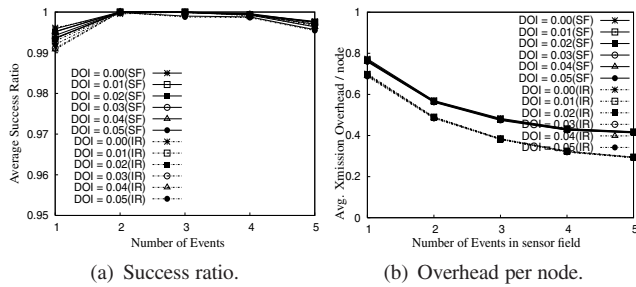
(b) Overhead per node.

Figure 7: *Max* query using Scoped-flooding (SF) and Information gradient-based routing (IR) with $\beta = 0.7$. Here, *Pr(link loss)* $= 0.1$.

Because, less number of scoped flooding is required to obtain the initial *Max* value. Also, at the early stages of query processing, *Max* value becomes high, so probing helps to avoid query spray that improves the overhead further.

7.4. Combined Query

We consider three sets of combined queries, where the area of cells corresponds to sub-queries are (1) equal (i.e., $1 : 1 : 1 : \dots$), (2) linearly increasing (i.e., $1 : 2 : 3 : \dots$) and (3) exponentially increasing ($1 : 2 : 4 : \dots$). The success probabilities in all cases are over 99% as shown in Fig.8(a) even in the presence of diffusion distortion and lossy wireless links. Also, the query processing overhead using gradient-based routing is below 50% as shown in Fig.8(b). However,
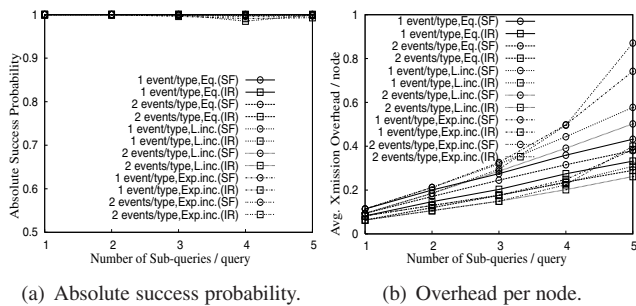

(a) Absolute success probability.

(b) Overhead per node.

Figure 8: *Combined* query using Scoped-flooding (SF) and Information gradient-based routing (IR) with $\beta = 0.7$. Here, *DOI* $= 0.05$ and *Pr(link loss)* $= 0.1$.

as the area corresponds to sub-queries increases exponentially, the overhead of query spray (i.e., dissemination) using scoped flooding increases sharply due to large cell area.

8. CONCLUSION

In this paper, we have presented a novel architecture, *Probe-before-Spray* (PBS) for information gradient-based active query processing. This reduces search overhead by exploiting geographical information and the diffusion spread to form resizable virtual cells within a query specified region. Based on PBS, we develop query-processing algorithms for aggregate queries and combined query.

We analyze the performance of PBS using simple analytical models. Also, through simulations, we found that *Count*, *Max* and *Combined* query algorithms based on PBS reduces search overhead over 40%, 30% and 50% respectively over usual flooding based approach while attaining accuracy over 99%.

In addition, the proposed architecture can be easily augmented with both Directed-Diffusion[2] and TinyDB[9] or Cougar[12] to reduce flooding overhead for energy efficient in-network query processing. Further, considering each virtual querier as a cluster head, PBS can also be used for hierarchical sensor networks.

REFERENCES

[1] A. Woo, S. Madden and R. Govindan, "Networking Support for Query Processing in Sensor Networks", Communications of the ACM, Vol. 47, No 6, June 2004.

[2] C. Intanagonwiwat, R. Govindan and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks", MobiCom 2000.

[3] N. Sadagopan, B. Krishnamachari, and A. Helmy, "Active Query Forwarding in Sensor Networks (ACQUIRE)", Journal of Ad Hoc Networks, Vol 3, Issue 1, pp. 91-113, January 2005.

[4] 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, Fall 2002.

[5] J. Liu, F. Zhao, and D. Petrovic, "Information-Directed Routing in Ad Hoc Sensor Networks", WSNA 2003.

[6] F. Zhao, J. Liu, L. Guibas and J. Reich, "Collaborative Signal and Information Processing: An Information Directed Approach", Proceeding of the IEEE , 91(8), 2003.

[7] Q. Fang, F. Zhao and L. Guibas, "Lightweight Sensing and Communication Protocols for Target Enumeration and Aggregation", MobiHoc 2003.

[8] J. Faruque, A. Helmy, "RUGGED: RoUting on finGerprint Gradients in sEnsor Networks", IEEE ICPS, 2004.

[9] S. Madden, M. Franklin, J. Hellerstein, and W. Hong, "TAG: a Tiny AGgregation Service for Ad-Hoc Sensor Networks". OSDI, December 2002.

[10] S. Madden, M. Franklin, J. Hellerstein, and W. Hong, "The Design of an Acquisitional Query Processing for Sensor Networks", SIGMOD 2003.

[11] A. Deshpande et al.,"Model-Driven Data Acquisition in Sensor Networks", VLDB 2004.

[12] Y. Yao and J. Gehrke, "Query Processing for Sensor Networks", CIDR, Jan. 2003

[13] H. Luo, F. Ye, J. Cheng, S. Lu and L. Zhang, "TTDD: Two-Tier Data Dissemination in Large-Scale Wireless Sensor Networks", Wireless Networks 11, pp. 161-175, 2005.

[14] B. Karp and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks", MobiCom 2000.

[15] G. Zhou et al., "Impact of Radio Irregularity on Wireless Sensor Networks", MobiSYS 2004.

[16] D.R. Askeland, The Science and Engineering of Materials, PWS Publishing Co., 1994.

[17] J.F. Shackelford, Intro to Materials Science For Engineers, 5th Ed., Prentice Hall, 2000.

[18] L. Reznik et al., "Embedding Intelligent Sensor Signal Change Detection into Sensor Network Protocols", IEEE SECON 2005.

[19] L. Gu et al., "Lightweight Detection and Classification for Wireless Sensor Networks in Realistic Environments", ACM SenSys 2005.

[20] J. Faruque, A. Helmy, "PBS: A Virtual Grid Architecture for Information Gradient-based Active Querying in Sensor Networks", USC, CS technical report 07-887.