Exploiting Redundancy in Sensor Networks for Energy Efficient Processing of Spatiotemporal Region Queries*

Alexandru Coman Mario A. Nascimento Jörg Sander
Department of Computing Science
University of Alberta, Canada
{acoman, mn, joerg}@cs.ualberta.ca

ABSTRACT

Sensor networks are made of autonomous devices that are able to collect, store, process and share data with other devices. Spatiotemporal region queries can be used for retrieving information of interest from such networks. Such queries require the answers only from the subset of the network nodes that fall into the query region. If the network is redundant in the sense that the measurements of some nodes can be substituted by those of other nodes with a certain degree of confidence, then a much smaller subset of nodes may be sufficient to answer the query at a lower energy cost. We investigate how to take advantage of such data redundancy and propose two techniques to process spatiotemporal region queries under these conditions. Our techniques reduce up to twenty times the energy cost of query processing compared to the typical network flooding, thus prolonging the lifetime of the sensor network.

Categories and Subject Descriptors

H.2 [**Database Management**]: Query Processing; Distributed Databases

General Terms

Algorithms, Performance

Keywords

Sensor networks, Spatiotemporal region query processing

1. INTRODUCTION

Sensor networks consist of nodes with the ability to measure, store, and process data, as well as to communicate wirelessly with nodes located in their wireless range. There are many application domains where sensor networks are well suited [16], e.g., environmental monitoring, warehouse management, traffic organization and surveillance. Sensor nodes are typically battery operated, which imposes hard constraints on their lifetime since query processing requires energy-wise costly communication among nodes. Therefore, energy-efficient query processing techniques are of utmost importance within a sensor network, and that is the main focus of this paper.

In domains such as GIS [1], a typical query involves building a map of values for a given region, e.g., "find the humidity for each point of lot X12 at 2pm yesterday". Since it is not practical to have a sensor node in each point of the monitored region, one has to settle for approximated values for those points where a sensor node is not present. This leads to the idea of building a map with values for each point in the map, with a confidence level attached to each value. In practice, this leads to two maps, one with the requested values and another one with the confidence levels. In a more general case, the values of interest could form a set of maps, one for each time point at a given time granularity, e.g., "find the hourly humidity values for the past 12 hours for each point of lot X12 as long as the confidence in the value is above 40%". We call this new type of query a SpatioTemporal Data Map (STDMap) query. An important fact to note is that in the same way some points of the map are "covered" by the approximations of one or more sensor nodes, the nodes themselves can also be covered by other sensor nodes. In effect this means that some nodes could take advantage of such coverage redundancy and not participate in the query processing, thus saving energy, without loss in the quality of the answer. As sensor nodes spend most of their energy during communication [14], we aim at minimizing the amount of data exchanged during query processing, further extending the lifetime of the network.

We study this problem over historical sensor data in a wireless sensor network, where all sensor nodes have similar capabilities, sensed measurements are stored locally, each node is only aware of the existence of the other nodes located within its communication range, and the query can be initiated at any node. The advantages of this environment are network robustness, a balanced use of sensors' energy resources, and a wide range of application scenarios where the presented solutions can be used. An application domain where such a query and sensor network fit well is environmental monitoring. The sensor nodes could be deployed from a plane over a region of interest. Upon activation, each node starts observing periodically various phenomena, such as the humidity of the soil. Park rangers patrolling through the region can access the network through any node in their proximity using a laptop. When certain events such as vegetation diseases or small fires are observed, the ranger could

* This work is supported in part by NSERC Canada. We would like to thank I. Nikolaidis for valuable discussions.

CIKM'05, October 31–November 5, 2005, Bremen, Germany.

query the network about historical measurements, which may lead to understanding what has caused such events.

Our main contribution in this paper is the proposal of two techniques to address the problem of approximate query processing in sensor networks, namely: EFM, which is an energy-aware parallel flooding, where a node decides whether it should participate or not in the query answering based on sound criteria including the amount of energy it has; and MSM, which is a technique that uses the completion of the query answer itself as a guide to traverse the region's nodes. Note that both techniques aim at taking advantage of the redundancy mentioned earlier. Through extensive experimentation, we show that the proposed in-network processing of STDMap queries reduces by up to twenty times the energy use compared to the typical network flooding that retrieves all the relevant sensor measurements and assembles the STDMap answer off-line.

The remainder of the paper is organized as follows. Section 2 describes research related to ours. Section 3 defines and discusses the STDMap query. Section 4 presents the characteristics of the wireless sensor network environment and introduces two algorithms for processing STDMap queries in this environment. Section 5 presents the experimental evaluation and Section 6 concludes the paper.

2. RELATED WORK

Sensor network technology lays at the confluence of several disciplines, including data management. Among the numerous issues under investigation, an important one is query processing due to its use in the retrieval of the data collected by the sensor network. Several influential works in this direction are [11, 14, 18]. They focus on retrieving real-time sensor measurements, possibly aggregated, and consider that each measurement is independent of one another. In real life this assumption does not generally hold due to the properties of the monitored phenomena (e.g., humidity varies smoothly between close locations). Query processing with approximate answering uses in various ways such properties of the monitored phenomena to substantially reduce query processing costs. Thus, it has recently raised much interest from the research community.

In [6] the query answers are estimated using a statistical model for the sensors' readings, where the model captures the redundancy and correlation in sensor measurements. The sensors are interrogated only when the uncertainty is high, which reduces the query processing costs substantially. To improve the fault tolerance of query processing for aggregations, duplicate insensitive sketches are used in [3] to produce accurate approximations of the aggregate answers. In [4], the authors exploit the correlation and temporal redundancy among the readings of each sensor to compress the short-term historical measurements. Once compressed, the measurements are transmitted to a base station for long-term storage. Spatial correlation in sensor readings is used in [19] to reduce the cost of processing aggregate queries. Caching the sensors' readings is used in [7] to reduce the cost of retrieving the sensor data, with the users specifying their tolerance for stale data in the query. Kotidis [13] introduces the snapshot queries as a different solution for approximate answering of queries, where a snapshot query is a query processed over a representative subset of the query relevant nodes. The correlation in sensors' measurements is also exploited in [5] to generate conditional query plans, where low cost attributes are used to determine the best plan for acquiring the high cost sensor measurements.

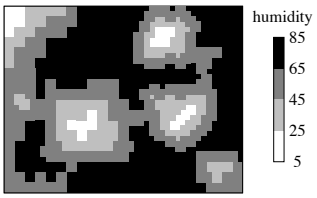

Figure 1: Example of a map

3. THE STDMap QUERY

A common representation of information in environmental remote sensing [1] is in the form of a map capturing the spatial distribution of data (Figure 1), where each map point represents a spatial area and its associated value represents the state of the monitored phenomenon in the area corresponding to the point. Query support for such a representation is important for applications where the spatial distribution of data is more important than individual data values. The map representation for sensor network data can be constructed straightforward by first collecting the sensor measurements from all sensor nodes located in the region of interest, followed by the construction of the map off-line. However, collecting the measurements from all these sensors (called relevant nodes) may be avoided if the answers of some nodes can be approximated by the answers of other nodes. That is, the measurements from only a *subset* of the relevant nodes may be sufficient to construct the map. Indeed, it is not practical to have a sensor node in each point of the monitored region, and therefore the values used for most of the map points must be approximated using the answers of the sensors located nearby. As well, due to the inherent correlation among the states of physical phenomena at close locations, the measurement of any sensor can be approximated with a certain degree of confidence by the measurement of other sensors located nearby. A map representation for sensor data has been first used in [8], but, differently from out work, each pixel in the map represents a sensor in the monitored region and its intensity indicates the sensor measurement.

Let us consider two locations where we are interested in the state of a monitored phenomenon: location s where a sensor node S exists and location l where there is no sensor node. S can provide a measurement for the monitored phenomenon only at location s. For location l, we can approximate a state with the measurement for location s, with some degree of confidence. We model the confidence of a node as a function $C(s, l)$, which represents the confidence of the sensor S that its measurement taken at location s is the same at location l. The choice of the function $C(s, l)$ depends on the monitored phenomenon and the capabilities of the sensing unit, and it is irrelevant with respect to the techniques presented in this paper. Typically $C(s, l)$ decreases with the increase in the distance between s and l. Note that our notion of confidence differs from the one used in [6]. Their confidence values represent the uncertainty with respect to approximated sensor readings, while ours captures the uncertainty in the validity of an actual sensor reading for a different spatial location than where it was acquired.

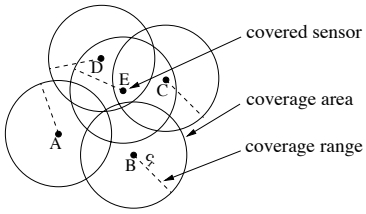

Figure 2: The coverage of sensors

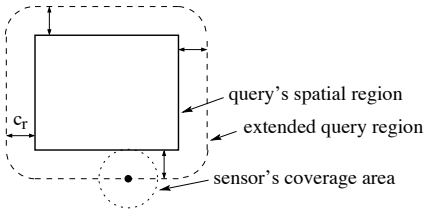

Figure 3: The extended query region

The SpatioTemporal Data Map (STDMap) query supports the map representation of the sensor network data. Assuming a confidence function *C(s,l)* which is dependent of the sensor network setting but is query-independent, we denote the query by STDMap*(qID,sr,tr,ct)*, with the following characteristics:

- It is identified in the sensor network using a unique query identifier *qID*.
- The answer consists of a set of map layers representing the approximations of the sensor measurements with confidence above the minimum confidence *ct* for the region *sr*, with each layer corresponding to one time unit within the (granular) time range *tr*.
- Each point of a map layer corresponds to an area within the query's spatial region, with its value equal to the approximated state of the monitored phenomenon in the corresponding area[1].

Note that each sensor node has a confidence in approximating the state of the monitored phenomenon with its measurement for every possible location. Unfortunately, interpolating the measurements of all sensors for each location using their confidences is not practical. Distributed regression is used in [9] to model spatiotemporal redundancy in sensors' measurements, where the user is responsible for providing the location of kernels and the set of basis functions. This is not feasible for large sensor network deployments. In this paper we associate with a map point the measurement with the highest confidence among all approximations obtained during query processing, reserving the problem of interpolating the sensors' approximations for future work.

Before going further let us introduce the following definitions which are necessary for the remainder of the paper:

DEFINITION 1. *Given a confidence function $C(s,l)$ and a confidence threshold ct, the coverage area of a sensor node is the area around the node in which the confidence of the sensor is above the threshold for every point in the area.*

DEFINITION 2. *Assuming the confidence function is uniform for all directions from a sensor node, the coverage range c_r of a sensor node is the radius of the circle, centered in the node, which forms its coverage area.*

As coverage areas can overlap (depending on the inter-sensor distance, the confidence function and the confidence threshold), the coverage areas of some sensors may be covered by other sensors. In this situation, the STDMap query can be answered using a subset of the relevant sensor nodes, thus saving communication and processing costs. For instance, in Figure 2 sensor nodes A, B, C and D have confidence above *ct* in their approximations for any point of the area covered by E, and thus the measurements of E are not required for answering the query. This is possible as the STDMap query does not require in its answer the measurement with the highest possible confidence, but with a confidence higher than the confidence threshold *ct*.

Since the coverage areas of the sensor nodes located in the proximity of the query's spatial region may intersect the region, finding the query answer for a location inside the query region may require contacting nodes located outside the region, leading to the following:

DEFINITION 3. *The extended query region is the region where the sensor nodes whose coverage areas intersect the query's spatial region can be located.*

Thus, any technique for processing STDMap queries must be able to contact the nodes located in the extended query region. Given a confidence function and a confidence threshold, the extended query region is formed by the extension of the query's spatial region in every direction with the coverage range as shown in Figure 3.

Due to the nature of the environment where the sensors are deployed, it is possible that the approximations of two neighboring sensors for the same location are inconsistent. This could be due, for instance, to possible obstacles affecting the expected variation of the monitored phenomenon. We leave the problem of inconsistent approximations for future work, and we assume in this paper that the nodes covering a certain location provide consistent approximations.

4. STDMap QUERY PROCESSING

We consider a wireless, peer-to-peer, sensor network with fixed nodes that have equal roles in the functionality of the network. A query can be introduced into the network through any of the sensor nodes, with query answers located in some (possibly all) of the nodes. Due to the wireless network characteristics, a sensor node can communicate directly only with the sensors located within its wireless range, which form its neighborhood. We assume that each node knows its location (e.g., through GPS), as well as the location of its neighbors (collected during network activation). Sensors take measurements periodically. The collected values are stored locally for future querying, and have attached the time-stamp corresponding to the time of measurement. A major constraint on sensor nodes is their limited energy supply. Since the energy required by sensing and computation is up to three orders of magnitude less than the energy used for communication [14], we are interested in minimizing the energy cost of communication during query processing.

[1]For areas where values with confidence above *ct* are not available, the map stores either *null* or values with confidence below *ct* from nodes that have answered the query.

We have shown in [2] that for spatiotemporal region queries a two-phase query processing approach is more efficient that the typical network flooding. By forwarding the query to all network nodes, the network flooding contacts many nodes in addition to those that hold the query answers, which increases substantially the energy cost of processing. The main advantages of this two-phase approach are that the query is disseminated to only a small subset of the network nodes and that spatial processing of sensor data can be performed closer to the data sources. Thus, we also use a two-phase approach in this paper. We break each processing algorithm into two phases: one for finding a routing path from the query originator node to the query region *sr*, the other for collecting the query answers from the relevant nodes and returning the answers to the originator node.

While the proposed techniques differ in their processing strategy for the second phase, they use the same routing algorithm for the first. We use a simple greedy approach to discover a routing path from the query originator node to a node located near the center of the query's spatial region, called coordinator node. At each step of the route discovery, the current node forwards the query to its neighbor located closest to the center of the query region. Once a node located in the query's spatial region receives the query and none of its neighbors is closer than itself to the center of the query area, this node assumes coordinator role. Greedy-based routing methods for position based routing have been shown to nearly guarantee delivery for dense network graphs [17], as it is the case for sensor networks. If the sensor network is not dense, more advanced geographic routing techniques such as GPSR [12] could be used to improve the route discovery for the first phase. Note that if the query originator node is located inside the query's spatial region, which includes the case where the query window covers the whole network, the first phase is not required. We present in the following the second phase for each query processing technique. When the queries are uniformly distributed over the monitored region, the coordinator role is also uniformly distributed among the sensor nodes. For non-uniform query distributions, the nodes better positioned to have coordinator role would be prone to early energy depletion, i.e., failure. In that case other nodes would be chosen as coordinators, not impairing the proposed techniques.

4.1 Energy-aware Flood (EFM)

The Energy-aware Flood for STDMap (EFM) technique uses a flooding strategy in its second phase, where a node decides to participate in the query answering after considering the state of its neighbors. EFM uses information about the amount of energy nodes have in order to decide which nodes should participate in query answering. Thus, EFM is an energy-aware technique.

Query processing starts when the coordinator node broadcasts the query to its neighbors, regardless of their state. A node receiving a query for the first time checks if its coverage area is covered by its neighbors. If it is not covered, the node decides that its answer is required and sets its state to SEND. If the area is covered, the node does not have yet sufficient information about its neighbors to safely decide to skip answering, and therefore sets its own state to OPEN. Next, the node broadcasts the query and its current state.

Once a node has received the query broadcast from all its neighbors, it checks if its neighbors that have decided to answer are covering its area, i.e., if its area is covered by nodes in SEND state. If its area is covered, the node can safely skip answering and it sets its state to SKIP. After this check, each node broadcasts a state update message to its neighbors and waits for the state updates from its neighbors. If a neighbor B of a node A has changed its state from OPEN to SKIP, it means that B is fully covered by its neighbors in SEND state, and thus any overlap it has with A is also covered. Consequently, a node can safely skip answering if its area is fully covered by its neighbors in SEND or SKIP states. Such a node changes its state to SKIP, but this information is not exchanged. Not exchanging the information on this status update is safe, as any node that is counting on the coverage of its neighbor in SKIP state remains covered (through transitivity) by the nodes covering its neighbor.

At this point, a node can be in any of the three possible states (SEND, SKIP or OPEN). If a node's state is SEND, the node returns its answer to the neighbor it first received the query from. If its state is SKIP, nothing has to be done as the node's coverage area is covered by other nodes that will answer. If its state is OPEN, the node must decide based on the information about its neighbors whether to send or to skip. To determine correctly whether it has to send its answer, however, it needs to know if information about its own area that is covered by neighbors that are also in OPEN state will be sent (by the neighbors directly or by other nodes covering the neighbors). This is a potential problem as the covering relation is symmetric[2]. Two nodes in OPEN state which partially cover each others areas have to independently make a consistent decision. In particular we must avoid that two nodes decide to skip and no other node will send information about their overlapping area. The information about the current state of its neighbors is not sufficient to take a consistent decision. Among the possible solutions for taking a consistent decision (e.g., node ID, turns, tournaments), we choose to use the amount of energy left in nodes. This choice leads to a more balanced energy usage at the nodes, as we show in Section 5.

In EFM, we use the amount of energy left in nodes to determine which one of a pair of nodes in OPEN state with overlapping coverage areas will ensure the coverage of the overlap (the energy information can be exchanged during the broadcasting of the state update message). For each pair, the node with more energy will be responsible for the coverage of the overlapping area. Thus, a node A in OPEN state with more energy will not count on a neighbor B in OPEN state with less energy to cover its area, and vice versa, B will count on the coverage by A, i.e., B will consider A to be in SEND state. This policy guarantees that for each pair of neighboring nodes in OPEN state with overlapping areas only one node will assume that the overlap is covered by the other node. If two nodes have the same amount of energy left, they use their unique node IDs as a tie-breaker. Any other tie-braking policy would work as long as both nodes make a consistent decision. After each node updates its local representation of the state of the OPEN neighbors according to the above policy, it checks again if its area is covered by neighbors in SEND or SKIP states. If its area is not covered, the node will assume its answer is required and sends it to the coordinator node. Otherwise, if its area is covered, it skips answering. Note that if a node A skips answering, this

[2]If the coverage areas of two neighbors overlap, each node may consider the other node covering the overlapping area.

Algorithm 1: EFM Technique - Phase 2

```
   Input    : Node N, NeighborList NB
 1 Receive query Q, PNB.state from ParentNeighbor PNB
 2 if N.location not in Q.extRegion then STOP
 3 NBE ← ∅ /* list of neighbors in Q's extended region */
 4 foreach node N_i in NB do
 5 |   if N_i in Q.extRegion then
 6 |   |   Add N_i to NBE
 7 Initialize status of all NBE nodes to OPEN state
 8 Update status of node PNB in NBE to PNB.state
 9 Construct coverage N.cov using N.location, Q.extRegion, Q.ct
10 Check if all NBE nodes cover N.cov
11 if N.cov is covered then
12 |   N.state ← OPEN
   else
13 |   N.state ← SEND
14 Broadcast Q, N.state /* SEND|OPEN state */
15 Wait for broadcasts of Q, status from all NBE nodes
16 Update status for all NBE nodes based on broadcasts
17 Check if NBE nodes with SEND state cover N.cov
18 if N.cov is covered then
19 |   N.state ← SKIP
20 Broadcast N.state, N.energy /* SEND|OPEN|SKIP state */
21 Wait for broadcasts of status, energy from all NBE nodes
22 Update status, energy for all NBE nodes
23 foreach node N_i in NBE do
24 |   if N_i.state = OPEN & N_i.energy > N.energy then
25 |   |   Change status of N_i in NBE to SEND state
26 Check if NBE nodes in SEND or SKIP state cover N.cov
27 if N.cov is not covered then
28 |   Return (N.data in Q.tr) to PNB
```

does not affect a neighbor B with less energy that counts on A for covering their overlap, since other nodes will answer for A's area. This discussion is summarized in Algorithm 1 and the correctness is stated in the following lemma:

LEMMA 1. *Any area from the query's spatial region covered by sensor nodes will be covered in the final answer using the EFM technique.*

PROOF. We assume the network within the extended query region is connected and every area within the query's spatial region it covered by the coverage areas of one or more sensor nodes. Let us assume there is an area A from the query's spatial region that is covered by exactly the set of nodes $\{N_1, ..., N_k\}$, and A is not covered in the answer, i.e., none of the nodes $N_1, ..., N_k$ is sending its answer to the coordinator node. Thus nodes $N_1, ..., N_k$ are in SKIP state. However, there is $j \in 1...k$ such that N_j has maximal energy among $N_1, ..., N_k$. The node N_j cannot skip. There is no neighbor N^* of N_j covering A having higher energy so that node N_j could count on N^* to cover A (if such a node would exist, it would be among $N_1, ..., N_k$, and N_j would not be the node with the highest energy in that set). Therefore, node N_j must be in SEND state, and area A is covered in the final answer. □

4.2 Map-guided Search (MSM)

The second processing technique for STDMap queries is MSM (Map-guided Search for STDMap). Differently from the EFM technique, MSM uses a partial query answer to guide the processing. MSM finds the STDMap answer by forwarding the query and the current partial STDMap answer. At each step, the current sensor node forwards the query to its neighbor that can provide answers with confidence above the confidence threshold ct to most map points

Algorithm 2: MSM Technique - Phase 2

```
   Input    : Node N, NeighborList NB
 1 Receive query Q, map M from ParentNeighbor PNB
 2 Update M using N. location, N.data and Q
 3 CNB = ∅ /* candidate neighbors for forwarding */
 4 foreach node N_i in NB do
 5 |   if N_i.location in Q.extRegion then
 6 |   |   Add N_i to CNB
 7 while CNB not empty & M not full do
 8 |   BN = ∅ /* best neighbor */
 9 |   foreach node N_i in CNB do
10 |   |   if gain(N_i, M) = 0 then
11 |   |   |   Remove N_i from CNB
12 |   |   if gain(N_i, M) > gain(BN,M) then
13 |   |   |   BN ← N_i
14 |   if BN ≠ ∅ then
15 |   |   Send Q, M to BN
16 |   |   Wait for M from BN
17 |   |   Remove BN from CNB
18 Return map M to PNB
```

that do not hold an answer yet. If a neighbor of the current sensor node has a coverage area where every location has already been covered with confidence higher than ct, then the neighbor is not contacted at all during query processing.

The query message received by a node contains both the query and the partial STDMap answer as obtained so far. The node first answers the query and adds its answers to the STDMap answer. Before forwarding the query to any neighbor, the neighbors that are not located within the extended query area are discarded from the list of candidate neighbors. The algorithm goes iteratively through the candidate neighbors and forwards the new partial STDMap answer to the best of them until either the query area is fully covered or there are no more candidate neighbors. The query answering is considered complete when STDMap's map layers have associated approximated values for every location. This procedure is formalized in Algorithm 2, where the *gain* function counts the number of map points covered by a node that does not have any approximated measurement associated yet. The correctness is stated in the following lemma:

LEMMA 2. *Any area from the query's spatial region covered by sensor nodes will be covered in the final answer using the MSM technique.*

PROOF. By construction of the MSM technique. We assume the network within the extended query region is connected and every area within the query region is covered by the coverage areas of one or more sensor nodes. The MSM technique uses sequential depth-first forwarding to contact nodes and therefore all nodes within the extended query region are contacted until the query region is fully covered. □

Since the query is forwarded in a sequential depth-first manner, only one node is processing the query at each point in time. This process ensures that only one copy of the partial STDMap answer is available in the network, and each node processing the query is aware of the contribution to the STDMap answer of the nodes previously involved in the query answering. While this strategy is likely to result in a longer query processing time for each individual query than the EFM technique, it facilitates several queries being processed simultaneously by the same group of relevant sensors.

Table 1: Parameters of query and sensor network

Parameter	Default Value
Size of monitored region	1000x1000 m
Wireless range	50 m
Average number of neighbors	15
Size of a measurement tuple	64 bits
Spatial region (sr)	4% (of region)
Temporal range (tr)	60 measurements
Confidence threshold (ct)	0.40, 0.80
Size of query message	256 bits
Energy used to transmit a bit	$\alpha + \gamma d^n$ nJ/bit [15]
Energy used to receive a bit	β nJ/bit [15]

5. EXPERIMENTAL EVALUATION

We implemented a sensor network simulator to study the performance of the proposed techniques. The placement of the sensor nodes follows a uniform distribution over a two dimensional region. The STDMap query requires the coordinates of a spatial region (sr), a temporal range (tr) and a confidence threshold (ct). The query's spatial region covers 4% of the monitored region, unless otherwise noted. The query's temporal range covers 60 measurements, which corresponds to one hour of measurements for a rate of one measurements per minute. For the answer of a STDMap query, each point of a map layer corresponds to 1 m^2 square area in the query's region. If a different map granularity is required, the granularity can be added as a parameter to the STDMap query. The query originator and the center of the query's region are uniformly distributed over the monitored region. The measurements are averaged over 10 randomly generated sensor networks with 10 random queries over each network. A summary of query and sensor network parameters and their default values used in our evaluation is presented in Table 1.

While our techniques are general with respect to the confidence function $C(s,l)$ used by the sensor nodes, we need an explicit confidence function in the evaluation[3]. We use the Gaussian distribution function $C(s,l) = e^{\frac{-d(s,l)}{2*\sigma^2}}$ to model the confidence function, where $d(s,l)$ represents the Euclidean distance between the sensor node located at s and location l and σ^2 is the variance of the function. Gaussian functions have been used before (e.g., [6]) to capture the behavior of physical phenomena and the correlation in sensor measurements. Using this function, the confidence of a sensor is high in its proximity and decreases rapidly with increasing distance. Given a confidence function $C(s,l)$ and a confidence threshold ct, the coverage area of a sensor node is determined by the confidence range c_r (see Section 3). When c_r is smaller than half of the wireless range, the nodes that cover a sensor's area are among its neighbors. When c_r is larger than half of the wireless range, the sensors nodes that cover a sensor's area may be located outside its wireless range. We use $\sigma = 50$ and show results for two confidence thresholds $ct \in \{0.4, 0.8\}$, which allows us to evaluate the algorithms for both situations.

We compare the performance of our techniques using two metrics. The first metric is the number of relevant sensor nodes used for answering the STDMap query. Using a smaller number of nodes in answering the query may increase the energy cost of processing due to an increase in the communication cost for the control messages. Thus, we use the total energy used during query processing as our second metric. Since we are interested in energy-efficient query processing, we are mainly interested in the energy metric, but the first metric allows us to better understand the behavior and trade-offs of each technique. We use the formulas in [15] (see Table 1) with the following parameter values for the energy costs [10]: $\alpha = \beta = 50\ nJ/bit$, $n = 2$, and $\gamma = 10\ pJ/bit/m^2$. Similar to other sensor network evaluations, our simulator considers that the message delivery is instantaneous and error-free between nodes communicating directly. In our experiments we only measure the energy used to transmit and receive messages during query processing, which includes the messages used for query forwarding, for returning the answers and for status updates. We focus on the energy-efficiency of the query processing and therefore the measurements are independent of the characteristics of the MAC layer (for instance 802.11 radios consume as much energy in idle mode as for receive mode, while other radios may switch to a low-energy state when idle).

We also compare our algorithms against a basic query processing solutions consisting of a network flooding with spatiotemporal constraints, called STF. In STF, the query originator node sends the query to all the sensor nodes, but only those located in the extended query region answer the query, returning the raw sensor measurements collected within the query's temporal range to the originator. Since this solution does not use a coordinator node, all query answers are returned to the originator node over the shortest path. For consistency with the format of data collected by the STF algorithm, the proposed techniques store the map layers as the raw measurements from which the layers can be constructed. This is reasonable as most values forming the map layers are approximations of a few measurements. In reality, the map layers could be stored as compressed images, which would substantially reduce their size.

5.1 Varying the Node Density

We use the average number of neighbors per sensor node to represent the sensor network density. This parameter combines the size of the monitored region, the number of sensor nodes and the wireless range. Studying the effect of the number of neighbors helps us understand the effect of these three parameters on the investigated techniques.

Figures 4(a) and 4(b) show the percentage of relevant nodes that answered the query for each technique. Note that for the STDMap query the relevant nodes are the nodes located within the extended query region. The STF method retrieves the measurements of all the relevant nodes (i.e., 100%), and therefore it is not shown. The EFM method forwards the query to all relevant nodes, with only some of these nodes answering the query. The MSM algorithm contacts only the nodes that are used in answering the query. As the number of neighbors increases, both EFM and MSM select a smaller percentage of the relevant nodes to cover the query region. In the case of MSM, each node has a larger set of neighbors to choose from for the next step of the algorithm, which leads to a better selection of the nodes used for covering the query region. For the EFM algorithm, a larger set of neighbors increases the probability that a node's confidence area is covered, which reflects on the lower percentage of nodes that answer the query. The increase in the

[3]Identifying the right confidence function to model the characteristics of physical phenomena is beyond the scope of this paper.

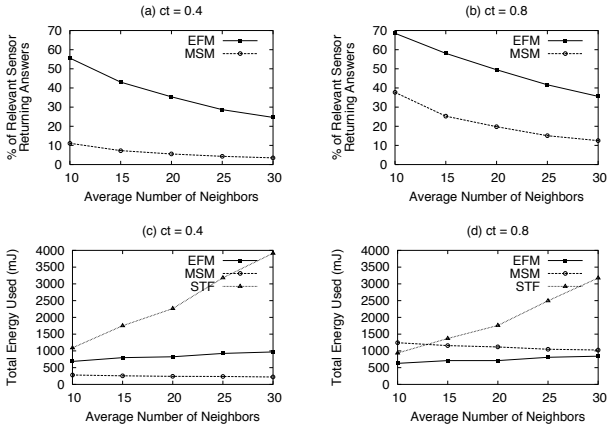

Figure 4: The effect of the number of neighbors

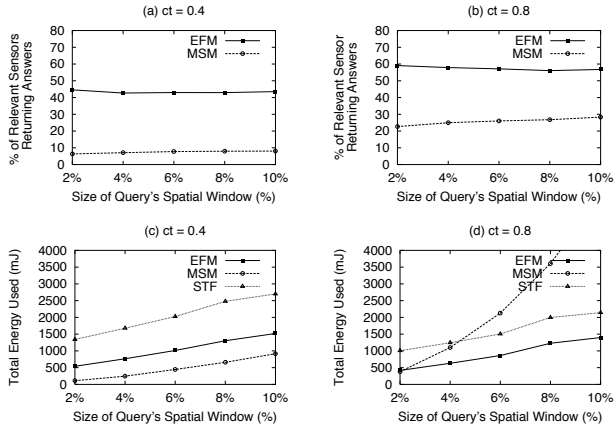

Figure 5: The effect of the query's spatial window

confidence threshold reduces the coverage range, which has a double effect on the query processing performance: both the extended query region and each node's confidence area are smaller. A smaller extended query region contains a smaller number of relevant nodes, but, on the other hand, the smaller coverage areas force more nodes to answer the query in order to cover the query region. Overall, the increase in confidence threshold forces a larger portion of the relevant sensor nodes to answer the query in both techniques. This effect is amplified in the MSM because of the reduced overlap among the sensors that MSM uses for query processing. For EFM, the overlap of nodes' coverage areas is high for $ct = 0.4$, which allows EFM to have a smaller increase than MSM in the number of relevant sensors nodes used for answering the query when $ct = 0.8$.

Figures 4(c) and 4(d) show the variation of the total energy used for processing a query when the network density increases. The STF algorithm is the most affected due to the linear increase in the number of relevant nodes that return answers. Our techniques use only a few more relevant nodes to answer the STDMap query for denser networks, which reflects in their relatively stable energy use. As MSM uses less nodes than EFM to answer the query, it has a lower energy cost for $ct = 0.4$. When nodes' coverage areas are small ($ct = 0.8$) and the network density is low, MSM contacts a larger percentage of the relevant nodes. This leads to a high energy cost for MSM. There are two reasons: the cost of transferring the STDMap partial answer to every contacted node in order to keep track of the already covered area is not negligible; and MSM uses a depth-first strategy that leads to most nodes contacted to be on the same path. Thus the STDMap answer is returned to the coordinator node over a long and costly path. Overall, the EFM technique is the least affected by the increase in network density. While MSM has the lowest energy use for low confidence thresholds (up to 20 times less than STF) and uses only a few of the relevant nodes to answer the STDMap query, it uses more energy than STF for a combination of low density and high confidence threshold.

5.2 Varying the Size of the Query's Region

In our second set of experiments, we varied the size of the query's spatial region between 2 and 10 percent of the size of the monitored region. Figures 5(a) and 5(b) show the effect of the query size on the percentage of relevant sensors that answer the STDMap query. The EFM algorithm uses each node's neighborhood to decide which nodes should answer, and therefore it is not affected by the size of the query region, using about the same percentage of relevant nodes to answer the query for all region sizes. In the case of MSM, larger query regions are more difficult to cover efficiently, and thus the percentage of relevant nodes that answer the query increases with the size of the query region. When the confidence threshold is high, both EFM and MSM use a higher percent of the relevant nodes to cover the query region since the area that each sensor covers is smaller. Consistent with our observation when investigating the effect of network density, higher confidence thresholds lead to a higher increase in the percentage of relevant nodes used by the MSM compared to the EFM technique.

Figures 5(c) and 5(d) show the total energy used during query processing when the size of the query's spatial region is varied. The MSM algorithm is affected the most due to the the larger number of nodes that it contacts and the increased length of the path over which these nodes are contacted and the answers are returned. The increase in the size of the query region also leads to an increase in the energy used by EFM as more nodes must answer the query to cover the larger query region. In addition, EFM uses two floods over the relevant sensor nodes for updating the nodes' status. Overall, the MSM uses the least energy when the number of nodes answering the query is small, but its cost increases sharply with the increase in the number of these nodes (Figure 5(d)). The EFM technique behaves better than MSM, the increase in the size of the query region causing a smaller increase in its energy costs.

5.3 EFM for Non-uniform Queries

To test out intuition that the EFM algorithm produces a more balanced energy use at the nodes within the extended query region, we compared it to a similar technique, called RFM, that uses a random tie-braking instead of the energy level used in EFM. We compared the effect of the EFM and RFM on the nodes' energy levels over several executions of the same query. We fixed the position and size of the query's spatial region while we allowed the originator node to be randomly selected among the network's nodes. We only measured the energy used for collecting the answers

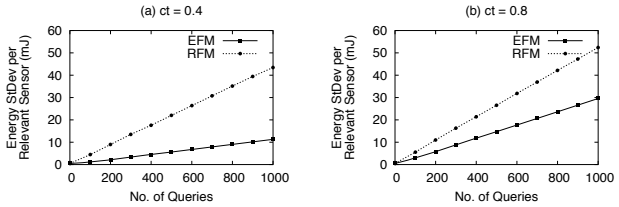

Figure 6: The energy balance at nodes

in the second phase of the EFM and RFM techniques. Before initiating the query processing, we charged all nodes with similar energy levels. After each set of 100 executions of the query, we calculated the average energy left in the nodes within the query region, and the standard deviation of energy from the average for the same nodes. We use the standard deviation to evaluate the energy balance among the sensor nodes.

Figure 6 shows the standard deviation of energy for the nodes within the query's spatial region. As more queries are processed, EFM produces a more balanced energy use compared to RFM. As RFM uses a non-energy aware tie-braking, it forces some nodes to use more energy than their neighbors, which may leads to their early failure. With the increase in the confidence threshold, the difference in the energy balance of the two technique increases. This benefit of EFM is likely to extend the quality of the query processing in the long-term since more nodes will be available, as well as possibly leading to an increased network lifetime.

6. CONCLUSIONS

In this paper we proposed the STDMap query which exploits the redundancy of sensor measurements on the spatial dimension. We showed that this type of query can be effectively answered by only a subset of the relevant nodes using the MSM and EFM techniques in a wireless sensor network environment with fixed nodes.

We studied the performance of the proposed techniques under several conditions. The MSM algorithm is best suited for queries where the coverage range c_r is larger than half the wireless range. In this case, MSM answers the STDMap query using a smaller subset of nodes than EFM and has a lower processing cost. The EFM algorithm is the most energy-efficient for most scenarios, showing consistent performance with respect to various network and query parameters. EFM also provides a balanced energy use at the sensor nodes, an important advantage in applications where the queries are not uniformly distributed. Therefore, we recommend EFM for processing STDMap queries as well as other queries with similar characteristics to the STDMap query.

Our investigations have revealed several issues that require further analysis. The trade-off between the robustness and energy-efficiency of query processing in the case of sensor failures requires further attention. The natural medium may also cause measurement inconsistencies among sensors and solutions for handling them are in our focus. In this paper we considered as the most reliable approximation the one with the highest confidence. Combining the approximations of several sensors based on their confidence is an open problem that has potential for improving the quality of approximations.

7. REFERENCES

[1] E.C. Barret and L.F. Curtis. *Introduction to Environmental Remote Sensing*. Stanley Thornes, 1999.

[2] A. Coman, M.A. Nascimento, and J. Sander. A framework for spatio-temporal query processing over wireless sensor networks. In *Proc. of DMSN Workshop*, pages 104–110, 2004.

[3] J. Considine, F. Li, G. Kollios, and J. Byers. Approximate aggregation techniques for sensor databases. In *Proc. of ICDE*, pages 449–460, 2004.

[4] A. Deligiannakis, Y. Kotidis, and N. Roussopoulos. Compressing historical information in sensor networks. In *Proc. of SIGMOD*, pages 527–538, 2004.

[5] A. Deshpande, C. Guestrin, W. Hong, and S. Madden. Exploiting correlated attributes in acquisitional query processing. In *Proc. of ICDE*, pages 143–154, 2005.

[6] A. Deshpande, C. Guestrin, S. Madden, J. Hellerstein, and W. Hong. Model-driven data acquisition in sensor networks. In *Proc. of VLDB*, pages 588–599, 2004.

[7] A. Deshpande, S. Nath, P.B. Gibbons, and S. Seshan. Cache-and-query for wide area sensor databases. In *Proc. of SIGMOD*, pages 503–514, 2003.

[8] S. Goel, and T. Imielinski. Prediction-based monitoring in sensor networks: taking lessons from MPEG. In *Proc. of CCR*, pages 82–98, 2001.

[9] C. Guestrin, P. Bodik, R. Thibaux, M. Paskin, and S. Madden. Distributed regression: an efficient framework for modeling sensor network data. In *Proc. of IPSN*, pages 1–10, 2004.

[10] W. Heinzelman. *Application-Specific Protocol Architectures for Wireless Networks*. PhD thesis, MIT, 2000.

[11] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva. Directed diffusion for wireless sensor networking. *IEEE Trans. on Networking*, 11(1):2–16, 2003.

[12] B. Karp and H.T. Kung. Greedy perimeter stateless routing for wireless networks. In *Proc. of MobiCom*, pages 243–254, 2000.

[13] Y. Kotidis. Snapshot queries: towards data-centric sensor networks. In *Proc. of ICDE*, pages 131–142, 2005.

[14] S. Madden, M.J. Franklin, J.M. Hellerstein, and W. Hong. The design of an acquisitional query processor for sensor networks. In *Proc. of SIGMOD*, pages 491–502, 2003.

[15] T. Rappaport. *Wireless Communications: Principles and Practice*. Prentice-Hall Inc., 1996.

[16] A. Ricadela. Sensors everywhere. *Information Week*, Jan. 24, 2005.

[17] I. Stojmenovic. Position based routing in ad hoc networks. *IEEE Communications Magazine*, 40(7):128–134, 2002.

[18] Y. Yao and J. Gehrke. Query processing in sensor networks. In *Proc. of CIDR*, 2003.

[19] S. Yoon, and C. Shahabi. Exploiting spatial correlation towards an energy efficient clustered aggregation technique (CAG). In *Proc. of ICC*, pages 82–98, 2005.