The d-Hop k-Data Coverage Query Problem in Wireless Sensor Networks*

Maria Kontaki
Aristotle University
Department of Informatics
Thessaloniki, 54124, GREECE
kontaki@delab.csd.auth.gr

Dimitrios Katsaros†
University of Thessaly
Computer & Communication Engineering
Volos, 38221, GREECE
dkatsar@inf.uth.gr

Yannis Manolopoulos
Aristotle University
Department of Informatics
Thessaloniki, 54124, GREECE
manolopo@delab.csd.auth.gr

ABSTRACT

The task of querying wireless sensor network (WSN) to retrieve data of interest is very significant and a wealth of query types have been proposed in the context of WSNs. This article describes the *d-hop k-data coverage query problem*, which is a novel query type, aiming at extracting "feature(s) distribution maps" from WSNs. This problem generalizes earlier research problems, like top-k, skyband, and *d*-hop dominating sets. For this problem, we provide a fully distributed solution, the DaCoN protocol, that avoids constructing a "network spanner", since such a structure requires an expensive initialization procedure, misses the notion of neighborhoods, and most importantly, it creates hotspots of communication, that shorten the network lifetime (i.e., some nodes deplete their energy very fast). We have developed a simulator to study the performance and behavior of the DaCoN protocol for various sensornet topologies and data distributions, and the obtained results attested the energy-efficiency and effectiveness of the protocol.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication*; C.2.4 [**Computer-Communication Networks**]: Distributed Systems—*Distributed applications*; E.1. [**Data**]: Data Structures—*Distributed data structures*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

Keywords

Data Coverage, Energy efficiency, Distributed Algorithms, Wireless Sensor Networks

*Research supported by a *ΓΓΕΤ* grant in the context of the project "Data Stream Management Systems" funded by *ΠΕΝΕΔ* national research program.

†Research supported by a *ΓΓΕΤ* grant in the context of the project "Data Management in Mobile Ad Hoc Networks" funded by *ΠΥΘΑΓΟΡΑΣ ΙΙ* national research program.

DMSN'08, August 24, 2008, Auckland, New Zealand.

1. INTRODUCTION

The constant improvements in computing and storage technologies as envisioned by Moore's Law, along with the progress in battery technology and micro-electro-mechanical systems, have revolutionized a new distributed embedded computing, where tiny low-power devices, i.e., sensors, equipped with a processor, memory, sensing and communication units are networked to support various applications in the scientific, medical, commercial, and military domains [11]. These wireless sensor networks (WSNs) have found interesting applications in environmental monitoring, smart homes and offices, intelligent transportation systems and many others.

A WSN consists of hundreds or even thousands of sensor nodes that are deployed inside or close to the 'phenomenon' being monitored. The positions of sensor nodes need not be engineered or predetermined; the sensors self-organize into an ad hoc network, where the communication among them is performed in a hop-by-hop fashion using multihop paths. WSNs represent a paradigm shift in computing in that they must support energy-efficient operations (since battery recharging is not possible/easy in hostile or harsh environments) and also must act in a cooperative, distributed manner, either because it is the only alternative to fulfil their goal (e.g., in collaborative target tracking) or because collaboration increases their capabilities (e.g., storage/computing capacity).

The ability to query a WSN and retrieve the data of interest is of paramount importance and a wealth of query types and evaluation methods have been proposed in the context of sensor networks (cf. Section 5). The majority of the proposed queries so far demand from the querer to have some knowledge of the network topology and/or to provide substantial specialization of the query. For instance, a query like "Report the k smallest values of humidity across the whole sensornet along with the sensors that sensed them" confine the querer in that the sensornet returns very few values that sometimes result due to sensor failures. On the other hand, a query like "Report the k smallest values of humidity within region X along with the sensors that sensed them" is sensitive to small shifts in the region boundaries, since the reported values may not be the k smallest anymore if the region shrinks or grows or moves even at a small amount. It would be much more informative if the sensornet could answer queries like "Report the sensors which sensed the k smallest humidity values in their neighborhood along with these humidity values". Such queries are very useful when we try to obtain a "feature distribution map" of the sensornet (e.g., regions of a monitored area with high pollutants distribution) by instructing the sensors in these areas to transmit a special beacon message. For instance, consider a environmental monitoring application using a wireless sensor network aiming to protect a forest from fire. Suppose that a number of fires with various intensities and extents have broken out in this forest. Then, a query which reports the maximum temperature across the whole network would be able to indicate the location of only one fire whereas we interested to identify locations with higher temperature with respect to the surroundings as possible fire nests.

How can a sensornet respond to such a query that does not involve global (sensor-wide) predicates (maxima/minima) and that underspecifies the boundaries of regions? We will present in the next section, that such a query, and much more generic ones, can be formulated with a query type, called the *d-hop k-data coverage query* introduced for the first time in the literature by the present article. This work makes the following contributions:

- Formulates a novel problem, i.e., the d-hop k-data coverage (query) problem, which is an ad hoc query run over static wireless sensor networks.
- This problem generalizes some well-known problems:
  - *top-k*, when the sensors produce one-dimensional data and the coverage predicate involves very restricted similarity functions (e.g., max/min, identity),
  - *skyband*, when the sensors must export all their results in an external warehouse, and the coverage predicate involves network-wide conditions,
  - *d-hop dominating set formation*, which is used for clustering wireless ad hoc networks.
- Provides a distributed, energy-efficient solution for responding to such queries, without the need to pre-establish any network "spanners", like broadcast trees, aggregation trees, which make the sensors higher up in the hierarchy to deplete their energy very fast.
- Evaluates the effectiveness and efficiency of the solution across a range of sensor network topologies and data distributions.
- Proposes routes for extending the query, i.e., continuous version, with generalized coverage predicates, etc.

The rest of the article is organized as follows: Section 2 presents the new query type via examples. Section 3 contributes a fully distributed protocol for evaluating such query types; Section 4 presents the performance measurement of the protocol. Section 5 surveys the relevant works, and finally, Section 6 concludes the article with a summary of the results and discusses our current work towards extending the article's ideas.

2. THE D-HOP K-DATA COVERAGE PROBLEM

Consider a small, sample WSN consisting of a few static sensors, where each node senses humidity and keeps the most recent values (depicted in the graph beside each sensor), and we take a snapshot of this network with the data values sensed by the sensors. Let this network snapshot be illustrated in Figure 1.

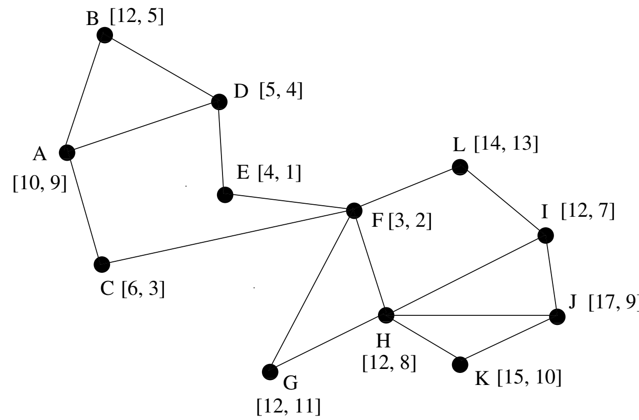


Figure 1: The concept of the d-hop k-data coverage query in a WSN.

There are quite a lot of approaches to easily answer a query like the following "Report the sensor(s) whose humidity value is not *covered* by any other humidity value across the whole sensornet". What this query seeks for is the sensor(s) whose humidity value is the maximum humidity across the whole sensornet. This is value 17, and the associated sensor is J. A generalized version of this query is the following "Report the sensor(s) whose humidity value is covered by at most k other humidity values across the whole sensornet", i.e., this query looks for the sensors whose sensed values are within the $k+1$ largest humidity values of the whole sensornet". For instance, for $k=2$, these are the values $17, 15, 14$, and the associated sensors are J, K, L. The "activated" (identified) sensors may comprise "islands" (e.g., sensor L) or larger "fields" (e.g., the field defined by the sensing area of sensors J, K). These examples demonstrated the notion of *k-data coverage query*.

In these queries we considered only humidity values. Apparently, the notion of k-data coverage extends straightforwardly in the multidimensional scenario with humidity and temperature, e.g., "Report the sensor(s) whose pair of humidity and temperature values is not *covered* by any other pair across the whole sensornet" and "Report the sensor(s) whose pair of humidity and temperature values is *covered by at most k other pairs* across the whole sensornet". It is the case that the k-data coverage query in the one-dimensional case, can be considered as a top-k query [21], but this does not hold in the case of multidimensional sensor readings, neither it holds under generic coverage conditions; thus the k-data coverage query is not identical to top-k queries in WSNs.

The main problems with all the aforementioned types of queries are the following: a) they are vulnerable to sensor failures, e.g., sensors reporting large values due to malfunction, and b) they are "network-wide" ("global"), in the sense that they can not address needs like the following "Depict the points (i.e., sensors) with the largest, relative to their neighboring sensors, humidities". Such queries are useful for constructing "feature(s) distribution maps" of the monitored area. Therefore, we need a "neighborhood-wide" ("localized") definition of the k-data coverage query. If we tried to address such a need by pre-specifying the neighborhood (i.e., define a region's geographical coordinates) and get the answers by using geocasting, then we would make the query results extremely sensitive even to small scales and translations of the specified region, which could result in missing the information being sought.

Instead, we must define the neighborhoods in abstract terms, using the notion of the d-hop neighborhood and come up with a solution that will not involve geocasting, convergecasting and so on. This formulation has some useful features: if we define d to be the sensornet diameter, then we get the *network-wide data coverage query*. For various values of d, we have different sizes of the neighborhoods. For instance, examining humidity values, for $d = 3$ and $k = 2$, the 3-hop 2-data coverage query, will identify the sensors A, B in one neighborhood whose values are not covered by more than two other values in their neighborhood, and also will identify the sensors J, K and L in a distant neighborhood, whose values are covered by at most two other values in their neighborhood. Notice here, that the values of sensors A, B are not among the results returned by the network-wide 2-data coverage query.

In the sequel, we propose the protocol DaCoN (Data Coverage in Neighborhoods), an efficient scheme for processing d-hop k-data coverage queries. The problem is stated as follows:

Given a sensor network and the size of the desired neighborhood d (in terms of hops) **determine** the sensor data that are covered by k other sensor data at most inside their neighborhood.

We will explain later (cf. Sections 3 and 5) how the d-hop k-data coverage problem, for particular values of the parameters d, and/or k, generalizes the ad hoc network clustering problem defined in [2], specific instances of the top-k problem [21], and the skyband query problem defined in [12].

3. THE DACON PROTOCOL

This section describes the DaCoN (Data Coverage in Neighborhoods) distributed protocol for processing d-hop k-data coverage queries in WSNs. Table 1 summarizes the basic symbols used throughout the study. This protocol is fully distributed, and runs localized in neighborhoods. Any solution that would rely on creating "network spanners", like aggregation or spanning trees [17, 21] suffers a) from the fact that it requires a demanding initialization phase to construct the spanner, and also b) from the fact that it creates hotspots in communication, i.e., the sensors higher up in the hierarchy. Such hierarchical structures force some sensors to deplete their energy quickly, and even when they minimize the average energy consumed, they contradict the optimality principle [4] which states that the energy consumption should be done as uniformly as possible across all nodes in order to prolong the network lifetime.

Assume there exists a WSN with gn sensor nodes. Each sensor S_i has a number of data values $S_i.vn$, e.g., temperature values of the last 5 minutes. For the sake of simplicity, we assume that all the sensors have vn data values,; though the proposed protocol apply without modifications with arbitrary number of data in each sensor.

The storage overhead per sensor S_i of the protocol is limited to maintaining two buffers (frb, srb) of d messages size each one. Each message contains the ID of the sensor that sends the message (source ID), the data values and the sensor ID of which sensor sensed each value, i.e., the data of a message are the values that a sensor has received (including its own values) and they are not covered by more than k other values.

Symbol	Description
D,**D'** and D,D'	dimensions sets and the number of dimensions
gn	number of sensor nodes of the network
S, S_i	a sensor
vn	the number of data values of the sensors
v_j, $S_i.v_j$	the j-th value of the i-th sensor
d	the number of hops for the neighborhood
k	the maximum number of values of which a value v_j can be covered.
$netDiam$	the diameter of the network
frb	buffer of messages for the first round
srb	buffer of messages for the second round

Table 1: Basic symbols used throughout the study.

In a multidimensional space, depending on the semantics of each dimension, in other cases the user may ask for maximization/minimization of the dimensions, or any other combination (minimization in some dimensions and maximization in the others). A data value *covers* another value, if the desired relationships among the values hold in every dimension.

For example, in a 2-dimensional space, assume that we wish the maximization of the first dimension and the minimization of the second one. Let $v_i.d_x$ denotes the x-th dimension of value v_i. A value v_i covers a value v_j, if it holds $v_i.d_1 > v_j.d_1$ and $v_i.d_2 < v_j.d_2$. In this work, we examine the case of 1-dimensional data, but the proposed protocol can be applied for the generic scenario of multidimensional data. In the sequel, without loss of generality and for ease of exposition, we restrict the coverage predicate to coincide with data value maximization.

The DaCoN protocol has three phases. In the first phase, the values that belong to the answer of the query, are identified. The second phase propagates smaller values that have survived from the first phase. This favors the selection of smaller values since the sensors can determine local maxima in the next phase. In the third phase, each sensor computes its answer. The next three paragraphs present the proposed protocol's operations.

PHASE 1. First d-rounds: In the first round, each node sends its k-th larger values to all 1-hop neighbors. Then, it finds the k-th larger values taking account its own values and the values that has received from its neighbors. Moreover, it forms a message with these values and it stores the message into frb. In the next d-1 rounds, the above procedure is repeated with the difference that now each sensor considers as its k-th larger values, the values of the last message of the frb.

PHASE 2. Next d-rounds: For the next d rounds the procedure is slightly different. Each sensor finds its k-th values by taking account the previous message and the messages that has received from its neighbors as follows: each v_i value ($1 \leq i \leq k$) is selected by keeping the smaller i-th value of these messages. These values form a message that is stored into srb.

PHASE 3. Answer of query: Overall, the DaCoN protocol runs in $2 \cdot d$ rounds. After these rounds, each sensor can decide the answer of the d-hop k-data coverage query. Each value v_i ($1 \leq i \leq k$) of the answer is selected as fol-

	Sensor nodes																							
	A		B		C		D		E		F		G		H		I		J		K		L	
	10	9	12	5	6	3	5	4	4	1	3	2	12	11	12	8	12	7	17	9	15	10	14	13
1.	12	10	12	10	10	9	12	10	5	4	14	13	12	11	17	15	17	14	17	15	17	15	14	13
2.	12	10	12	10	14	13	12	10	14	13	17	15	17	15	17	15	17	15	17	15	17	15	17	14
3.	14	13	12	10	17	15	14	13	17	15	17	15	17	15	17	15	17	15	17	15	17	15	17	15
4.	12	10	12	10	14	13	12	10	14	13	17	15	17	15	17	15	17	15	17	15	17	15	17	15
5.	12	10	12	10	12	10	12	10	12	10	14	13	17	15	17	15	17	15	17	15	17	15	17	15
6.	12	10	12	10	12	10	12	10	12	10	12	10	14	13	14	13	17	15	17	15	17	15	14	13
	12	10	12	10	14	10	12	10	14	13	14	13	17	15	17	15	17	15	17	15	17	15	14	13

Table 2: An example execution of the DaCon protocol for $d = 3$ and $k = 2$.

lows: the sensor compares the messages of frb and srb and tries to find pairs of values in the first i-th values of each message. After the identification of all pairs of values, the sensor selects the minimum pair as the i-th value of its answer. If a pair of values does not exist, the sensor selects the maximum of the first i-th values of the messages of frb.

During the rounds of message exchange, we can avoid message collisions, and contention for the wireless channel, by creating a TDMA schedule following the algorithm in [7]. Thus, it is safe to assume that there are not message losses.

Example execution of DaCoN: Table 2 shows the execution of DaCoN protocol for the network of Figure 1 with $d = 3$ and $k = 2$. Initially, each node sends its two larger values. Lines 1, 2 and 3 show the message that is stored into frb at the end of round 1, 2 and 3, respectively of the first phase, and the lines $4, 5$ and 6 the respective messages of the second phase. For example, sensor node C sends values 6 and 3. At the end of round 1, has received values 10 and 9 from sensor A and values 3 and 2 from F. Sensor C finds that values 10 and 9 are the larger ones. In Table 2, we give only the values and omit other information of the messages for clarity reasons. In the same way, at the first round of the second phase sensor node C sends values 17 and 15. At the end of round 4, it has received values 14 and 13 from sensor A and the values 17 and 15 from F. Sensor C finds that the values 14 and 13 are the smaller ones.

After $2 \cdot d$ rounds, sensor C will compute its answer. First, it tries to find the largest value. Thus, it searches only the first value of each message and tries to find pairs between the messages of frb and srb. Only value 14 has a pair both in the two buffers. Then, sensor C tries to find the second value. Now, it searches in the first two values of each message and ignores values equal or larger than the answer of the previous step (i.e., 14). There are two pairs (values 10 and 13). In this case, sensor C selects the minimum pair. $\dashv$

The DaCoN protocol can also handle cases where the minimization of attributes is required by choosing smaller values in the first d rounds and larger values in the next d ones. Moreover, DaCoN can handle multidimensional data after some slight modifications, but these considerations are beyond the scope of the present article.

The d-hop k-data coverage problem as a generalization of existing problems. The d-hop k-data coverage query is a generalization of three existing problems. In the case where $k = 1$, the d-hop k-data coverage query is similar to finding a d-hop dominating set in wireless ad hoc networks [2]; however, there are substantial differences among the two problems. [2] proposed a heuristic that can handle only the maximization of one dimension, i.e., of the sensor ID. In contrast, DaCoN handles data of the sensor and therefore it is able to handle multidimensional values of different semantics (maximization, minimization or combination of them, etc). Moreover, DaCoN can answer in $2 \cdot d$ rounds k values instead of only one value.

In the case where $d = netDiam$, the query is transformed to the k-skyband query [12]. The algorithm proposed in [12] does not take into account the notion of neighborhood and it is not distributed, therefore it is not appropriate for wireless sensor networks, which require localized algorithms.

Finally, the d-hop k-data coverage problem can be considered as generalization of top-k queries [21]. In the case of 1-dimensional space and when $d = netDiam$, if the preference function of the dimension is MAX or MIN, the d-hop k-data coverage query is transformed in a top-k query. Though, as stated, the algorithms that have been proposed for top-k queries, can be applied only if $d = netDiam$. Thus, these algorithms can not be used for the d-hop k-data coverage problem.

4. PERFORMANCE EVALUATION

In this section we present our experimental setup which consists of a protocol simulator and two generators, sensor network topology and sensor data generator. Since the query type is novel, there exist no other methods that could be implemented as competitors. One could consider the case of continuously outsourcing all sensor data in an external warehouse and running the query there, but clearly such solution is not energy efficient, and moreover it is not appropriate for ad hoc queries, like the present one.

4.1 Network topology generator

We created network topologies, modelling features such as the existence and "strength" of clusters of sensors, density of sensor nodes, and so on. We observed that the topologies generated with procedures like that in [16], or with procedures that distribute nodes randomly in the plane, are alike the Random Graph Model of Erdös-Rényi. Although this model is quite useful, we argue that it is not suitable for modelling ad hoc network graphs, because these graphs are not formed uniformly at random, but present a *group/cluster-based behavior*. Thus, we had to resort to richer graph models that model the existence of clusters, like that of Pennock [13]. The parameters of such a network topology generator that we developed are the following:

- gn: the number of network nodes (default value: 500).

- gc: the number of network clusters (default value: 20).
- gd (*density*): a float depicting the fraction of edges relative to the edges of a complete graph with gn nodes; small values of gd simulate a small transmission radius. (default value: 0.20).
- $ga \in [0.5 \ldots 0.99]$ (*assortativity*): a float depicting the fraction of edges which exist inside the clusters, relative to the total number of edges present in the graph (default value: 85%). Large values ($> 85\%$) simulate clusters with very dense linkage inside them and only a few links toward other clusters, whereas values around 0.50 completely "blur" the existence of clusters.

4.2 Sensor data generator

The sensor data generator has the following parameters:

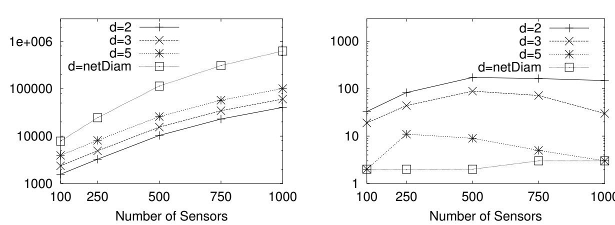
- inn: number of initial network nodes (default value: 10).
- dn: number of data values for each network node (default value: 100).
- $minv, maxv$: the range of data values.

Initially, we randomly peak inn sensors and assign to them dn values in the range $[minv,\ maxv]$. Next, for each initial sensor, the sensors in their 1-hop neighborhood are chosen. We skip sensors that have already data assigned to them. We generate smaller values for them than the data values of the initial nodes. We add noise in the data values to avoid a monotonous reduction. The procedure is repeated until we have assigned data to all the sensor nodes.

4.3 Experimental evaluation

Competitors of our proposed protocol could be well established methods in the database community for top-k and skyband queries. However, the majority of these methods assume a centralized scheme or use some form of aggregation tree that create "hotspots", which reduce significantly the network lifetime. Thus, these methods are inappropriate in wireless sensor networks.

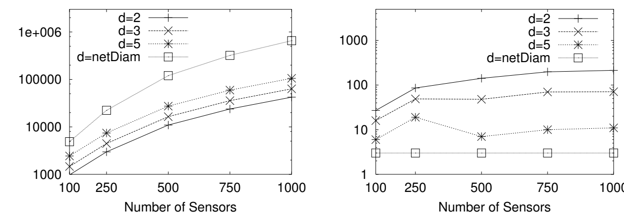
We examine the performance of the protocol by varying several of the most important parameters such as the number of sensor nodes, the parameter k (default value: 3), the parameter d, the number of clusters and the network assortativity. We measure the number of messages, the *activated sensor nodes* (i.e., the sensor nodes belonging to the answer) and the hotspots of the sensornet.



(a) (b)

Figure 2: Impact of sensornet size (number of clusters is fixed) on: (a) number of messages, (b) number of activated sensors.

First, we study the performance of the DaCoN protocol w.r.t. the number of sensors in the network. Figure 2 depicts the results for varying values of parameter d. As expected, the number of messages increases with increasing d and also for larger sensornets. The number of activated nodes increases with decreasing d since, implicitly we define more neighborhoods, and thus the protocol runs more localized. But when the sensornet size increases, the number of activated sensors decreases, because, since the number of clusters remains the same, the readings of a few sensors in each cluster cover many more sensors, i.e., the neighborhoods are merged.

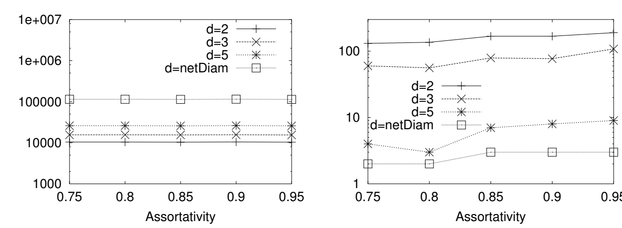


(a) (b)

Figure 3: Impact of sensornet size (number of clusters increases with sensornet size) on: (a) number of messages, (b) number of activated sensors.

To better comprehend this result, Figure 3 illustrates the results of the same experiment with the difference now being that the number of clusters and the number of the initial sensor nodes varies analogously to the number of sensor nodes. Recall that the number of neighborhoods that the proposed protocol discovers, depends on both the sensornet topology and the data distribution. In this experiment, the ratio of the number of clusters to the number of sensors, and the ratio of the number of "initial sensor nodes" to the number of sensors remain fixed, thus the number of activated sensors increases as the size of the sensornet increases.

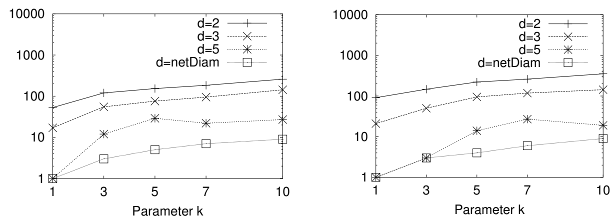
Next, we examine the performance of the DaCoN protocol w.r.t. the network assortativity. The results are depicted in Figure 4. The size of the network and the number of cluster are both fixed. It is evident that the assortativity scarcely affects the number of messages, though the number of activated sensors increases, since large values for assortativity induce well separated neighborhoods.



(a) (b)

Figure 4: Impact of network assortativity on: (a) number of messages, (b) and number of activated sensors.

The plots in Figure 5 show the number of activated nodes w.r.t. parameter k. As expected, the number of activated



(a) (b)

Figure 5: Impact of parameter k on the number of activated nodes in: (a) small sensornets $gn = 500$, (b) larger sensornets $gn = 1000$.

sensors increases with increasing k. The results are similar for the two networks of 500 and 1000 sensors. Since the messages are not affected by the parameter k, this result is omitted.

Sensornet size	Number of messages sent per sensor			
100	≤ 24 74.0	25 - 50 14.0	51 - 75 8.0	76 - 100 4.0
250	≤ 19 69.6	20 - 50 23.2	51 - 80 5.6	81 - 110 1.6
500	≤ 31 70.0	32 - 100 26.8	101 - 155 3.0	156 - 210 0.2
750	≤ 45 65.3	46 - 120 29.8	121 - 210 4.5	211 - 300 0.4
1000	≤ 60 68.9	61 - 150 26.2	151 - 250 4.2	251 - 350 0.7

Table 3: Percentages of sensor nodes w.r.t. the number of messages that they transmit.

Finally, we identify the "communication hotspot" sensors for the experiment reported in Figure 2. Table 3 depicts the results. The top number of the second column in each row gives the average number of messages per sensor node. For instance, for $gn = 500$, the average number of messages per sensor is 31. The bottom number in each column gives the percentage of sensors that sent that many messages as the top number (or range) indicates. For instance, for $gn = 100$, 74% of the sensor nodes sent less than 24 messages, 14% sent 25 up to 50 messages, 8% sent between 51 and 75 messages and so on. Therefore, the second and third columns show the sensor nodes that sent only a small number of messages. These sensor nodes are in all cases more than 85% of the whole network sensors. The fourth and fifth columns contain the percentages of sensors which can be considered as hotspots since the number of messages that they sent is far above the average. We observe that the number of hotspots reduce as the network size increases, proving that the protocol can achieve a uniform energy consumption.

5. RELEVANT WORK

Although there is no prior work on the d-hop k-data coverage problem in WSNs, the following areas are somehow related to this problem:

Coverage problems in WSNs. Since the sensors can be spread in an arbitrary manner, one of the fundamental issues in a WSN is the coverage problem, i.e., to determine how well the sensing field (i.e., each point or target) is monitored or tracked by sensors. Moreover, the issue of coverage and connectivity arises in redundant sensor deployments in order to conserve energy via sleep scheduling. In the literature, the coverage problem has been formulated in various ways; for instance, k-sensor coverage [10], coverage with directional antennas, coverage problems with irregular sensors, partial coverage, probabilistic coverage [9], coverage in 3D networks [1], and so on. For a survey of the area consult [20]. Nevertheless, none of these formulations and solutions is appropriate for the problem considered here, since they do not deal with the data generated by the sensors, but rather with the positions and networking of the sensor nodes.

Queries in WSNs. Queries posed to a WSN can be classified as "one-time" queries, i.e., those injected at random times to obtain a snapshot view of the data attributes, and "periodic or standing or continuous" queries, i.e., those retrieving data from the sensors after regular/continuous time intervals. Depending on the nature of the data processing demanded by the application, queries can be classified as:

- Simple queries [14], e.g., "Report the value of the humidity".
- Aggregate queries [5, 15], e.g., "Report the average humidity of all sensors in region X".
- Approximate queries [8], requiring data summarization to perform holistic data aggregation in the form of histograms, contour maps, e.g., "Report the contour of toxic chemical gas in region X".
- Complex queries [11], which, if expressed in SQL, would involve joins nested or conditioned-based sub-queries (like the aforementioned). A sample complex query could be the following one "Among regions X and Y, report the average humidity of the region with the highest temperature".
- Advanced queries, such as top-k [17, 21] and k-NN [6], which retrieve more sophisticated knowledge from the data. For example, a top-k query with MAX as the preference function could be "Report the k data objects with the highest temperature".

Despite their wealth, none of the above queries is alike the d-hop k-data coverage query; moreover, most of them are solved using some form of an aggregation tree that inevitably creates "energy holes" (i.e., the sensors higher up in the hierarchy become hotspots in communication), which contradicts the optimality criterion for achieving the maximum network lifetime as described in [4]. Even the algorithms for top-k [3, 17], which are characterized as distributed, in fact they are using some sort of "base station" or "coordinator". On the other hand, our presented query type employs a generic, localized description and admits a fully distributed solution.

Skyline and skyband. Skyline queries have received considerable attention recently, due to their aid in selecting the most preferred items, especially when the selection criteria are contradictory. Although the problem has been attacked by taking a database perspective [12], only recently it has been investigated in sensor networks [18, 19]. However, the proposed data coverage query is a generalization of skyline query and therefore these approaches can not be applied in the problem considered here.

In [12], an interesting variation of skyline query, skyband query, has been proposed. A k-skyband query reports the set of points which are dominated by at most k points. Thus, skyband query can be considered as a data coverage query. However, skyband assumed to be a centralized query while our proposed query involves the notion of neighborhood and it is processed in a fully distributed manner.

6. CONCLUSIONS

We described a novel problem, the d-hop k-data coverage problem in WSNs. Solutions to this problem can aid in the creation of "features distribution maps" in WSNs. For this problem we presented a fully distributed solution, the DaCoN protocol, that refrains from creating any network spanners, like aggregation or spanning trees, since these structures require an expensive initialization phase and also create hotspots in communication. The described distributed protocol poses very light storage, computational and communication requirements in each sensor. To study the behavior of the protocol we performed a simulation-driven performance evaluation against a series of network and data distributions. Currently, we are working towards extending the study of the protocol along several directions: we consider multi-dimensional sensor readings, more complex coverage predicates, and continuous d-hop k-data coverage queries over WSNs.

7. REFERENCES

[1] S. M. N. Alam and Z. J. Haas. Coverage and connectivity in three-dimensional networks. In *Proceedings of ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, pages 346–357, 2006.

[2] A. D. Amis, R. Prakash, T. H. P. Vuong, and D. T. Huynh. Max-min d-cluster formation in wireless ad hoc networks. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, pages 32–41, 2000.

[3] B. Babcock and C. Olston. Distributed top-k monitoring. In *Proceedings of ACM International Conference on Management of Data (SIGMOD)*, pages 28–39, 2003.

[4] J. H. Chang and L. Tassiulas. Energy conserving routing in wireless ad-hoc networks. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, pages 22–31, 2000.

[5] A. Deligiannakis, Y. Kotidis, and N. Roussopoulos. Hierarchical in-network data aggregation with quality guarantees. In *Proceedings of International Conference on Extending Data Base Technology (EDBT)*, pages 658–675, 2004.

[6] T. Y. Fu, W. C. Peng, and W. C. Lee. Optimizing parallel itineraries for kNN query processing in wireless sensor networks. In *Proceedings of ACM International Conference on Information and Knowledge Management (CIKM)*, pages 391–400, 2007.

[7] R. Gandhi and S. Parthasarathy. Fast distributed well connected dominating sets for ad hoc networks. Technical Report CS-TR-4559, Computer Science Department, University of Maryland at College Park, 2004.

[8] D. Ganesan, D. Estrin, and J. S. Heidemann. Dimensions: Why do we need a new data handling architecture for sensor networks? *ACM SIGCOMM Computer Communication Review*, 33(1):143–148, 2003.

[9] M. Hefeeda and H. Ahmadi. A probabilistic coverage protocol for wireless sensor networks. In *Proceedings of IEEE International Conference on Network Protocols (ICNP)*, pages 41–50, 2007.

[10] C.-F. Huang and Y.-C. Tseng. The coverage problem in a wireless sensor network. *ACM Mobile Networks and Applications*, 10(4):519–528, 2005.

[11] H. Karl and A. Willig. *Protocols and Architectures for Wireless Sensor Networks*. John Wiley & Sons, 2006.

[12] D. Papadias, Y. Tao, G. Fu, and B. Seeger. Progressive skyline computation in database systems. *ACM Transactions on Database Systems*, 30(1):41–82, 2005.

[13] D. M. Pennock, G. W. Flake, S. Lawrence, E. J. Glover, and C. L. Giles. Winners don't take all: Characterizing the competition for links on the Web. *Proceedings of the National Academy of Sciences*, 99(8):5207–5211, 2002.

[14] S. Ratnasamy, B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin, and F. Yu. Data-centric storage in sensornets with GHT: A geographic hash table. *ACM Mobile Networks and Applications*, 8(4):427–442, 2003.

[15] M. A. Sharaf, J. Beaver, A. Labrinidis, and P. K. Chrysanthis. Balancing energy efficiency and quality of aggregate data in sensor networks. *The VLDB Journal*, 13(4):384–403, 2004.

[16] I. Stojmenovic, M. Seddigh, and J. Zunic. Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 13(1):14–25, 2002.

[17] M. Wu, J. Xu, X. Tang, and W.-C. Lee. Top-k monitoring in wireless sensor networks. *IEEE Transactions on Knowledge and Data Engineering*, 19(7):962–976, 2007.

[18] J. Xin and G. Wang. Energy-efficient skyline queries over sensor network using mapped skyline filters. In *Proceedings of International Asia-Pacific Web Conference/ Web Age Information Management (APWeb/WAIM)*, pages 144–156, 2007.

[19] J. Xin, G. Wang, L. Chen, X. Zhang, and Z. Wang. Continuously maintaining sliding window skylines in a sensor network. In *Proceedings of International Conference on Database Systems for Advanced Applications (DASFAA)*, pages 509–521, 2007.

[20] M. Younis and K. Akkaya. Strategies and techniques for node placement in wireless sensor networks: A survey. *Ad Hoc Networks*, 6:621–655, 2008.

[21] D. Zeinalipour-Yazti, Z. Vagena, D. Gunopulos, V. Kalogeraki, V. J. Tsotras, M. Vlachos, N. Koudas, and D. Srivastava. The threshold join algorithm for top-k queries in distributed sensor networks. In *Proceedings of ACM International Workshop on Data Management in Sensor Networks (DMSN)*, pages 61–66, 2005.