

# Deployment-aware Energy Model for Operator Placement in Sensor Networks

Félix J. Villanueva\*, Michael Daum†, Moritz Strübe†, J.C López\*, Rüdiger Kapitza†, Falko Dressler‡

\*Department of Information Technologies and Systems, University of Castilla-La Mancha, Spain

†Department of Computer Science, University of Erlangen, Germany

‡Institute of Computer Science, University of Innsbruck, Austria

**Abstract**—Query processing has become one of the main paradigms to approach information processing in Wireless Sensor Networks (WSNs). We study and identify relevant factors involved in in-network query deployment in WSNs. As a key outcome, we developed a deployment aware cost estimation model for distributing operators to nodes within the network using the network lifetime as a key metric. In contrast to related approaches, we include the state of each node in the cost model. This provides means for an automatic load balancing mechanism that allows us to insert a higher number of queries compared to other approaches, thus, significantly extending the network lifetime.<sup>1</sup>

## I. INTRODUCTION

The concept of in-network query processing in sensor networks has been emerged as a hot research topic, mainly because of the possible efficiency combined with several challenging issues such as the determination of the optimal operator distribution [1], [2]. A query, expressed in a query language (e.g., SQL), can be split into a set of operators, ones related with each other and susceptible to be executed in the sensor nodes. The possibility of distributing the operators of a query directly into the sensor network appears to be an efficient method avoiding the recollection of all raw data at a sink node, just to execute the query at the sink [3].

Directly associated with in-network query processing is the optimal emplacement of operators. This is considered one of the key problems to be addressed. The assignment of one or several operators to a specific sensor nodes implies that the operator has to be migrated from its current position (e.g., the sink) to its new position. There are, of course, other problems like the operator configuration or the state management.

The first step in solving the operator placement problem is to locate operators that reduce the amount of data transmitted (e.g., aggregation functions) in order to optimize the energy consumption of wireless transmissions. Then, all the other operators have to be distributed optimizing the entire network lifetime [4], [5]. Thus, the main goal is to wisely decide which operator has to be distributed to which node.

With this aim in mind, this paper summarizes proposals and earlier works that help to understand the requirements for designing a cost estimation model and to establish a candidate solution for the operator distribution problem. We

specifically focus on the migration of operators, taking into account the current state of sensor nodes in the operator placement decisions [6].

Our work is part of the RDSP project<sup>2</sup> and it provides relevant metadata for the Data Stream Application Manager (DSAM) [7], the central query manager of RDSP. DSAM supports the deployment of global queries to distributed and heterogeneous sensor nodes and Stream Processing Systems (SPSs). For the operator emplacement, we use a graph-based global query language DSAM-AQL.

In our architecture, we build a *logical view* from the real WSN topology by means of a *Catalog* entity, which includes, for instance, information about memory and energy available in each node and about already deployed operators. Based on this information, the *DSAM-AQL parser* builds a graph with the operator relationships for a submitted query and generates the code for each operator. The cost model carries out a matching process looking for the best operator distribution. The obtained optimal operator placement is used by the sink to generate a set of commands (according to a migration control protocol) to effectively migrate the operators to the corresponding nodes. This can for example be handled using the stateful mobile modules approach [6], [8]. As stated before, the aim of this process is to prolong the overall lifetime of the sensor network [4]. In our case, we consider the network lifetime as the total amount of queries that can be injected before one of the queries fails due to energy starvation of the sensor nodes. If there is no alternative operator distribution for successful query execution, we consider the WSN dead from a functional point of view.

The main contributions of this paper can be summarized as follows. Primarily, we show the influence of parameters such as operator migration cost, distance between nodes and topology on the operator placement problem. Based on the analysis of previous cost model proposed, we define a new cost model for operator placement including all the studied parameters. Our cost model minimizes network transmissions and includes a per-hop state consideration in order to exclude nodes with low-level of energy in the operator placement decisions. Including this per-hop state energy parameter enables a load balancing mechanism, which significantly extends the

<sup>1</sup>This work is supported by Spanish Ministry of Science and Innovation under grants CEN-20091048(Energos) and TEC2008-06553 (DAMA)

<sup>2</sup>Research group on Resource constrained Distributed Stream Processing, see <http://rdsp.informatik.uni-erlangen.de/>

sensor network lifetime.

The rest of the paper is structured as follows. Section II describes some earlier works related with cost models for in-network query processing and associated problems. Section III introduces the proposed per-hop state cost model. This section also outlines different considerations about the operator migration process. Section IV is devoted to analyze individual factors that have a strong influence on the optimal operator location. Section V shows the performance of our cost model under different random topologies and analyzes some problems identified during this test phase. Finally, in Section VI, we draw some conclusions and discuss possible future work.

## II. RESEARCH QUESTION AND RELATED WORK

In a WSN, sending data continuously from any sensor node to one or several sinks can be expensive in terms of power consumption. In order to save energy, in-network query processing has been studied as an efficient way to avoid raw data transmission between the sensor nodes. Several architectures follow this concept like, such as the ones proposed in RDSP [7], SNEE [9] and SSDQP [10].

In all these architectures, a query is expressed according to a query language (e.g., SQL or similar languages). This query is then split into operators (e.g., filter, join, or union), which are inserted into the network. In the stream processing concept, each operator directly returns the processed results of the query. In many cases, a window may define the time required by the query. For example, to calculate an average temperature over ten minutes, this time window indirectly defines the amount of data to be processed.

Having this in mind, one of the key questions that should be answered is where operators should be placed for an optimal WSN performance. Effectively, to combine possible emplacements and different configurations for a query with several operators involved constitutes an NP-hard problem [11]. This problem deserves greater attention from the research community looking for adequate heuristics and algorithms in order to reduce the computation time and to find operator distributions close to the theoretical optimum. All these algorithms strongly depend on a cost model for a correct node emplacement.

To devise a cost model for operator placement, we will demonstrate that the following aspects need to be considered:

- *Operator size* – Operator-specific costs could, depending on the query characteristics, become an important factor.
- *Distance between nodes* – It is well known that routing algorithms mainly use the number of hops as cost function. But, in sensor networks not only the number of transmissions is important but also the physical distance between the nodes.
- *Selectivity of operators* – We define the selectivity as the ratio between incoming and outgoing data streams. This is a key aspect to reducing the amount of data transmitted.
- *Topology* – According to our study, the topology of the network is a crucial aspect for achieving efficient in-network query processing.

In [12], Chatzimilioudis et al. develop an operator placement for snapshot multi-predicative queries in sensor networks. They use the energy of returning the results to the sink as their cost function. However, they do not consider the cost of disseminating the query itself. For energy estimation, they use the cost model proposed in [13], which is representative of works in this area since it provides a global view of the energy cost of a given solution (e.g., the operator distribution in the WSN). Such cost models do not take into account the current state of the network, i.e., the state of each node. The work presented in [5] includes the distance between nodes in the energy cost for operator deployment, but, again, does not consider operator migration costs and reduces the problem to a minimum communication cost placement from a global point of view. A more comprehensive lifetime metric has been presented in [4], which also takes the operational aspects of the network into account.

Despite most of the existing works deal with received data in each node, they usually do not take the total amount of data received by each node into account. Instead, they just consider the data directly addressed to that node [14].

## III. PER-HOP STATE COST MODEL

If we have a query  $Q$  expressed in any query language, we can represent it in form of a set of operators  $O$  connected in a directed graph by a set of edges  $E$ . Each operator represents a vertex in the graph, and each edge represents the data flow between operators. So we can express the query like  $Q = \{E, O\}$  where  $O = \{o_1, o_2, \dots, o_i\}$  with  $o_1, o_2, \dots, o_i$  is the list of operators for query  $Q$ . The set  $E$  can be expressed as:

$$E = \{e_1, e_2, \dots, e_i\} : e_i = \{o_i, o_j\}, o_i, o_j \in O, o_i \neq o_j \quad (1)$$

where an edge  $e_i$  represents that the data flow from  $o_i$  to  $o_j$ .

Each  $o_i$  has associated a selectivity  $s_i$  that is characterized as a ratio between the input stream size and its output stream size. The set of operators can be either traditional database operators (e.g., aggregate, filter, correlate) or user-defined operators. We call  $\Omega$  of a query  $Q$  a valid solution, i.e., a set of emplacements for all operators involved in the query  $Q$  that satisfy all memory and energy requirements. In the same way, assuming a WSN of  $N$  nodes, we can define  $o_i^n$  as the emplacement of operator  $o_i$  in node  $n$  of the sensor network.

A solution has to satisfy several restrictions in order to be considered valid.

The memory available of a node  $n$  ( $n^{mr}$ ) has to be sufficient to store all operators to be placed at node  $n$ :

$$\forall n \in N : n^{mr} \geq \sum_{i \in O} size(o_i^n) \quad (2)$$

Also, if we call  $P$  a route between nodes  $x$  and  $y$ , it is necessary that for all connections between operators in a specific placement, there is a path  $P$  (in the WSN) between these two emplacements:

$$\forall e = o_i^x, o_j^y, e \in E : \exists P = x, y \quad (3)$$

The notation  $o_i^x$  and  $o_j^x$  is possible if two operators are placed on the same node.

We further have to consider energy restrictions. If we can estimate the energy spent per node for a given solution, this amount of energy should be less or equal to the energy available in the node. Between all valid solutions, we need to choose the one that prolongs the overall network lifetime. In our case, we define a simple metric for network lifetime: the number of queries executed in the network before a query fails as one or more nodes involved in the solution cannot provide sufficient remaining energy.

If we call  $C_s$  the estimated cost for a valid solution including the energy necessary for operator migration, we need to add a factor representing the available energy in such way that a single node with energy problems will increase the cost of all solutions including this node. With this factor, we expect to achieve an automatic load balancing mechanism that prolongs the network lifetime.

We call  $x_n^t$  the energy available of node  $n$  at time  $t$ . The capacity factor ( $Cf$ )  $Cf_n^t$  of node  $n$  at time  $t$  is modeled as

$$Cf_n^t = \frac{1}{e^{x_n^t}} \quad (4)$$

Using an exponential function clearly increases the cost of a solution even if just one of the nodes involved in the solution has energy problems. To model the estimated energy consumption, we focus on the transmission and reception of packets, since both processes represent a high percentage of the energy consumed by a sensor node. For example, Klan et al. [15] show that the energy consumed in a single sensor measurement is lower or equal to the sending of a single byte.

For each node  $n$ , the data transmitted  $D_{tx}^n$  for a given query solution is composed of

- the data stream that must be forwarded (without change at node  $n$  and belonging to stream queries or migration of operators),
- the sum of all data transmitted by each operator running at node  $n$ , and
- the data transmitted by operators running at node  $n$ , which had to be migrated.

Similarly, the data received  $D_{rx}^n$  for a given query solution is composed of

- the received data stream that must be forwarded,
- the sum of all data received by each operator running at node  $n$ , and
- the data received as a result of neighbors transmission but not addressed to node  $n$ .

In order to calculate the data transmitted  $D_{tx}^n$ , let us call predecessor subset  $P_i$  of an operator  $o_i$  the set of operators connected to operator  $i$  where this operator is the consumer of the data. Using the edge definition from Equation 1 predecessor subset  $P_i$  is defined as

$$P_i = \{g_1, g_2, \dots, g_j\} \forall g \in O, g \in P_i \Rightarrow \exists e \in E \wedge e = \{g, o_i\} \quad (5)$$

Similarly, the successor subset  $S_i$  of  $o_i$  is defined as

$$S_i = \{c_1, c_2, \dots, c_j\} \forall c \in O, c \in S_i \Rightarrow \exists e \in E \wedge e = \{o_i, c\} \quad (6)$$

Let  $output(P_i)$  be the amount of data generated by  $P_i$  in bytes. Now, we can express  $D_{tx}^n$  as

$$D_{tx}^n = D_f^n + \sum_{o_i^n} (output(P_i) * S_i) * |S_i| + M_{tx} \quad (7)$$

where  $|S_i|$  represents the cardinality of  $S_i$  (we assume that an operator sends the same data to all its destinations).  $M_{tx}$  represents the data transmitted if node  $n$  contains an operator that has to be migrated to another node. Finally,  $D_f^n$  represents the data forwarded by node  $n$  (and which was not processed at  $n$ ).

In the same way, the data received by node  $n$  is  $D_{rx}^n$  and can be expressed as

$$D_{rx}^n = D_f^n + \sum_{o_i^n} (output(P_i)) + M_{rx} \quad (8)$$

Again,  $M_{rx}$  represents the data received if an operator is placed at node  $n$  as a result of the migration process.

Now, the energy consumed by node  $n$  for a given query solution can be expressed as

$$E_n = D_{tx}^n * E_{tx} + D_{rx}^n * E_{rx} \quad (9)$$

where  $E_{tx}$  and  $E_{rx}$  represent the cost for sending and receiving a single byte, respectively. We need to mention that  $E_{tx}$  takes into account the distance of the next hop for each packet transmitted (again, we assume that the energy required for the transmission varies considerably with distance [16]). If we include the capacity factor of each node, the cost function of a valid solution  $\Omega$  can be expressed as

$$C_\Omega = \sum_{n \in N} ((\beta + D_{tx}^n)E_{tx} + (\alpha + D_{rx}^n)E_{rx})Cf_n^t \quad (10)$$

Here, the constants  $\beta$  and  $\alpha$  have been introduced to include the effect of other data not directly related with the operators' functionality, that is, routing data exchanged between nodes, topology control protocol, and others. For the sake of simplicity, both constants have been omitted in the simulations.

#### IV. EVALUATION OF COST MODEL RELEVANT FACTORS

In this section, we discuss the influence of different factors in the cost model estimation. In order to show this influence, we developed a simulator that takes a WSN topology expressed by mean of an XML file and evaluates all candidate positions (according to Equation 10) for a query expressed as an operator tree. The simulator takes the ideal solution following our cost model and applies this solution, updating battery and operator place for each node and operator, respectively. Finally, the simulator calculates the number of queries successfully injected in the network. To shorten the simulation time, all nodes start with charge of 2753.75 J (about 25% of an AA lithium-ion battery). In the simulator, the battery charge is normalized between 10 and 0.1 in steps of 0.1.

TABLE I  
STREAMING CONVERSION FOR EACH TYPE OF OPERATOR

Operator	Out streaming size
Union	$\text{size}(S1) + \text{size}(S2)$
Join	$\sigma * C(S1) * C(S2) * (\text{AIS}(S1) + \text{AIS}(S2))$
Conditional merge	$\sigma * C(S1) * (\text{AIS}(S1) + \text{AIS}(S2))$ , having $C(S1) \approx C(S2)$
Filter	$\sigma * C(S1) * \text{AIS}(S1)$
Map	$C(S1) * \text{new\_AIS}(S1)$
Aggregate	$(\#G / \text{agg\_size}) * C(S1) * \text{AIS}(S1)$

The simulator is implemented in Python using a python-graph library to model the WSN topology and the graph built from the set of operators (and their relationships). In the following sections, we will show how the simulator helps us to evaluate of different aspects of the overall performance of the cost model. We always compare our approach with a global estimation of the energy without taking into account the energy in each node. So, when we talk about a global energy estimation, we refer to Equation 10 without considering the capacity factor  $C_n^t$ .

### A. Stream Manipulating Operators

Different operators used for sensor information fusion tasks have been considered, namely combining operators (union, join, and conditional merge), nomadic operators (filter, map, and aggregate) and separating operators. Each one of them modifies the size of the stream. If we have two raw streams labeled S1 and S2, considering  $C(S1)$  as the cardinality of S1 and  $\text{AIS}(S1)$  as the average item size of S1, we show in the Table I the size of the output stream for each type of operator. We consider  $\sigma$  as the selectivity of the operator (with values between 0 and 1).

Whenever  $\sigma$  gets close to 1, most of the operators increase significantly the size of the stream (i.e., join, conditional merge). In most of these cases, it is more convenient from the in-network processing query point of view to place the operators at the sink node.

For simplification, we will consider a user-defined operator with selectivity  $\sigma$  between 0.2 and 1.6 in the evaluation of the cost model. This represents the set candidates to be inserted in the network. The precise operator type is not relevant for the presented cost model.

### B. Distance Between Nodes

Another consideration broadly ignored in the cost model development is the distance between nodes. Works that deal with topology control show how to regulate the transmission power of the nodes to save energy [17]. This aspect has a strong impact in the overall network lifetime.

The simulator gets the distance between nodes from the topology file and calculates, based on data presented in [16], the TX power level for a CC2420 radio chip [18]. With the TX power level necessary for the transmission, we estimate the energy required per byte according to data presented in [19].

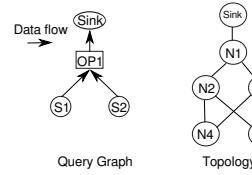


Fig. 1. Query graph and network topology

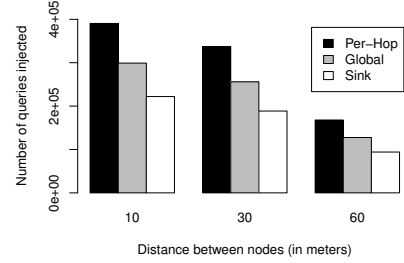


Fig. 2. Successfully injected queries for a join operator of 10 KB

In order to show this influence, we executed a simulation that calculates the optimal position following a greedy approach that tests all candidate solutions. We define a query in which two sources send a stream to a user-defined operator that join both streams and sends the data to a sink as depicted in Figure 1 (left). This query could be expressed in SQL as:

```
CREATE STREAM out AS
SELECT * FROM
conditional_merge (S1, S2)
```

S1 and S2 represent the source streams and the conditional\_merge would be a user-defined operator (OP1 in Figure 1). We assume that the result of the query has to be accessed at the sink node in the WSN.

The internal structure of the user-defined operator is not relevant in this case, but we can assume that it has a selectivity of 0.2. For the sake of simplicity, we insert this query in a simple topology outlined in Figure 1 (right). In our simulator, we repeatedly executed this experiment. Our simulator stops when it is not possible to insert the query in the WSN because some nodes involved are out of battery and there is not an alternative solution.

In Figure 2, we can see a comparison of three possible approaches. The “sink” bars represent a basic approach where the operator is always executed at the sink; “global” bars consider the overall consumed energy to calculate the operator placement; and finally the “per-hop” bars show the behavior of our per-hop state cost model. To show the influence of the node distance, we changed this parameter between 10 m, 30 m, and 60 m. In this simulation, the operator size is 10KB with a selectivity of 0.2. Source node emplacements are N4 and N5, each node has an initial battery of 2753,75 J and Rx power consumed is 1.8912 nJ/byte.

The basic approach just assigns the join operator to the sink

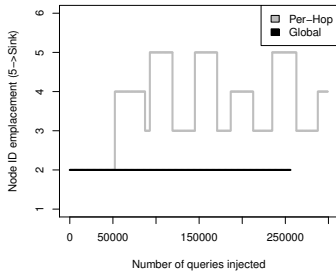


Fig. 3. Emplacement for a user-defined operator of 10 KB

node. Thus, it does not spend energy for operator migration. We use this approach as a reference. Following the traditional approach, we should estimate the energy spent for each solution (including the migration process) from a global point of view. The state of each node is not considered. As we can see in Figure 2, this “global” option improves the lifetime clearly in comparison to the basic approach. With our per-hop state cost model and taking as capability factor the exponential of the available energy, we are more sensible to nodes’ energy starvation. So, we migrate the operator in function of the state of each node. As can be seen, this further improves the expected network lifetime. With our approach, we can execute significantly more queries compared to the “global” approach. This shows that the distance between the nodes has a strong influence on the maximum number of executed queries.

In order to visualize the behavior of the “global” vs. the “per-hop” model, we plot in Figure 3 the emplacement of the user-defined operator over time (actually over the number of executed queries representing time in our experiments). If we use an overall energy estimation, the operator is fixed at a single node (dark line). Using our per-hop state cost model, the operator location changes according to the state of the node (grey line). The operator migration has a cost, but even with this extra cost, we are able to execute more queries and to prolong the network lifetime.

## V. PERFORMANCE EVALUATION

With the aim of studying the overall performance of the proposed per-hop cost model, we have defined a new query with two operators where one of them merges two sources and the result is merged again with another source stream. In SQL, we can express this query as follows:

```
CREATE STREAM out AS
SELECT * FROM
conditional_merge(
conditional_merge(S1, S2, ...), S3, ...);
```

The operator tree for this query is depicted in Figure 4. As can be seen, the three sources S1, S2, and S3 are sending data to operators OP1 and OP2, respectively.

As mentioned before, our cost model, together with the dynamic operator replacement mechanism, enables an automatic load balancing process. To demonstrate this effect, we tested our cost model in a regular square topology of 5x5 nodes

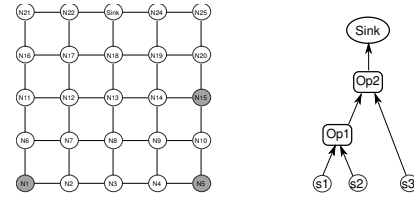


Fig. 4. Grid topology (left) and query graph model (right)

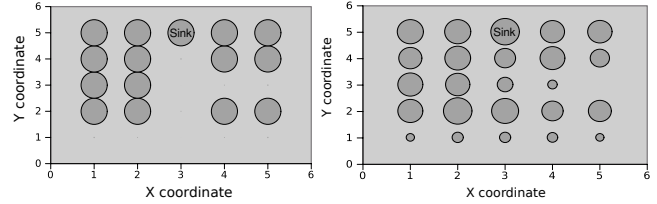


Fig. 5. Energy available in nodes with overall energy estimation (left) and with per-hop state consideration (right)

(Figure 4). The sources are at nodes labeled with N1, N5 and N15, located in coordinates (1,1), (5,1) and (5,3), respectively. The operator size is 3 KB and the distance between nodes is set to 60 m. To simplify the simulation, the selectivity of operators  $\sigma$  is 0.5, source size of 150 KB and the nodes have an initial battery of 2753.75 J. When we talk about source size we refer to the size of the streaming processed according with the window defined in the query.

With this topology and configuration, the overall energy consumption of all nodes is equal as we can see in Figure 5 left. In this figure, the diameter of the circles represents the remaining energy after the simulation terminated. As can be seen, using the global operator distribution approach, the topology becomes partitioned. This approach selects coordinates (3,1) and (3,3) for the two operators

In contrast, our per-hop cost model performs a load balancing mechanism and stops only when one of the sources, in this case sensor node N15 at (5,3), is running out of energy. The rest of the network is still operational (Figure 5 right).

In a final experiment, we compared the total network lifetime for both approaches. Here, we analyzed the number of queries that we can execute with each solution. In this experiment, we generated 10 random topologies of 25 nodes and, again, apply the same query (see Figure 4 right). The sources are located at nodes N1, N5 and N15, independently of the operator distribution. The operator size is 3 KB and the distance between nodes ranges randomly between 1 m to 60 m. To speedup the simulation, the initial value of the battery was set to 110.5 J.

The average of successfully executed queries is shown in Figures 6 and 7 for source sizes of 10 KB and 150 KB. The results are depicted in form of boxlots. For each data set, a box is drawn from the first quartile to the third quartile, and the median is marked with a thick line. Additional whiskers extend from the edges of the box towards the minimum and maximum of the data set, but no further than 1.5 times the interquartile range. Data points outside the range of box and

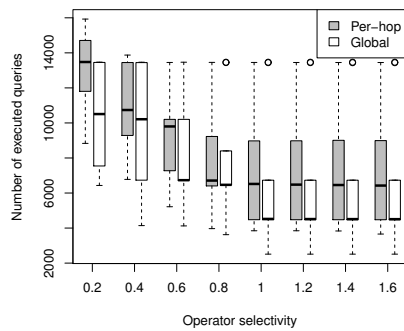


Fig. 6. Average number of queries injected for a streaming size of 10 KB

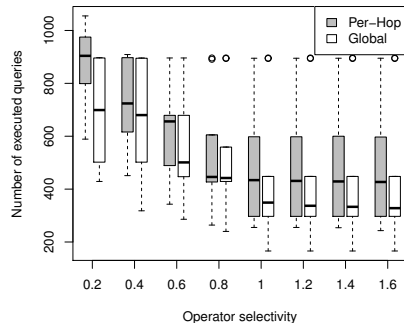


Fig. 7. Average number of queries injected for a streaming size of 150 KB

whiskers are considered outliers and drawn separately, as small circles. As we can see in all cases, our cost model (grey bars) performs significantly better compared to the global energy model. When the operator selectivity is larger than one, both approaches present a lower limit. In this case, the key limitation is the battery available in the sink's one-hop neighbors. The global approach emplaces both operators (OP1 and OP2) at the sink. So, the data flows converge in the sink's one-hop neighbors. Again, our approach performs a load balancing between sink's neighbors, when possible, in order to prolong the overall network lifetime.

## VI. CONCLUSIONS

Cost estimation is a key factor for the success of in-network query streaming architectures. The final decision about the operator distribution for the queries in sensor networks is crucial, especially, if operator migration is possible to prolong the network lifetime.

A review of the literature shows that when dealing with in-network query processing in sensor networks, most of the presented solutions allow a good overall energy estimation, but do not take all communication costs into account. Our key contributions are the comprehensive study of important factors like node distance, operator size and selectivity, and the available remaining energy; as well as the resulting definition of a novel cost model taking all the energy constraints into account. The per-hop cost model especially focuses on the operator migration cost and all communication related energy demands. As can be seen from our experiments, our model

is significantly outperforming related solutions, proving its effectiveness for optimal operator placement.

## REFERENCES

- [1] J. Gehrke and S. Madden, "Query Processing in Sensor Networks," *IEEE Pervasive Computing*, vol. 3, no. 1, pp. 46–55, January–March 2004.
- [2] C. Avin and C. Brito, "Efficient and robust query processing in dynamic environments using random walk techniques," in *3rd ACM/IEEE International Symposium on Information Processing in Sensor Networks (IPSN 2004)*. Berkeley, CA: ACM, April 2004, pp. 277–286.
- [3] L. Golab and M. T. Ozsu, "Issues in Data Stream Management," *ACM Sigmod Record*, vol. 32, no. 2, pp. 5–14, 2003.
- [4] I. Dietrich and F. Dressler, "On the Lifetime of Wireless Sensor Networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 5, no. 1, pp. 1–39, February 2009.
- [5] K. Kalpakis and S. Tang, "Maximum lifetime continuous query processing in wireless sensor networks," *Elsevier Ad Hoc Networks*, vol. 8, no. 7, 2010.
- [6] M. Strübe, R. Kapitza, K. Stengel, M. Daum, and F. Dressler, "Stateful Mobile Modules for Sensor Networks," in *6th IEEE/ACM International Conference on Distributed Computing in Sensor Systems (DCOSS 2010)*, vol. LNCS 6131. Santa Barbara, CA: Springer, June 2010, pp. 63–76.
- [7] M. Daum, M. Fischer, M. Kiefer, and K. Meyer-Wegener, "Integration of Heterogeneous Sensor Nodes by Data Stream Management," in *10th International Conference on Mobile Data Management: Systems, Services and Middleware (MDM 2009)*. Taipei, Taiwan: IEEE, May 2009, pp. 525–530.
- [8] M. Strübe, M. Daum, R. Kapitza, F. Villanueva, and F. Dressler, "Dynamic Operator Replacement in Sensor Networks," in *7th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS 2010), Demo Session*. San Francisco, CA: IEEE, November 2010, pp. 748–750.
- [9] I. Galpin, C. Y. Brenninkmeijer, F. Jabeen, A. A. Fernandes, and N. W. Paton, "Comprehensive Optimization of Declarative Sensor Network Queries," in *21st International Conference on Scientific and Statistical Database Management (SSDBM 2009)*, vol. LNCS 5566. New Orleans, LA: Springer, June 2009, pp. 339–360.
- [10] U. Rohm, M. M. Gaber, and Q. Tse, "Enabling resource-awareness for in-network data processing in wireless sensor networks," in *19th Conference on Australasian Database (ADC 2008)*, Darlinghurst, Australia, January 2009, pp. 107–114.
- [11] B. J. Bonfils and P. Bonnet, "Adaptive and Decentralized Operator Placement for In-Network Query Processing," *Telecommunication System*, vol. 26, no. 2-4, pp. 389–409, 2004.
- [12] G. Chatzimilioudis, H. Hakkoymaz, N. Mamoulis, and D. Gunopulos, "Operator Placement for Snapshot Multi-predicate Queries in Wireless Sensor Networks," in *10th International Conference on Mobile Data Management: Systems, Services and Middleware (MDM 2009)*. Taipei, Taiwan: IEEE, May 2009, pp. 21–30.
- [13] A. Coman and M. A. Nascimento, "A Distributed Algorithm for Joins in Sensor Networks," in *19th International Conference on Scientific and Statistical Database Management (SSDBM 2007)*. Banff, Canada: IEEE, July 2007, p. 27.
- [14] J.-h. Zhang and F.-j. Shao, "BF-K: a Near-optimal Operator Placement Algorithm for In-network Query Processing," *Journal of Networks*, vol. 5, no. 10, 2010.
- [15] D. Klan, K. Hose, and K.-U. Sattler, "Developing and deploying sensor network applications with AnduIN," in *6th International Workshop on Data Management for Sensor Networks (DMSN 2009)*. Lyon, France: ACM, August 2009, pp. 1–6.
- [16] M. Mallinson, P. Drane, and S. Hussain, "Discrete Radio Power Level Consumption Model in Wireless Sensor Networks," in *4th IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS 2007)*. Pisa, Italy: IEEE, October 2007, pp. 1–6.
- [17] F. Khadar and D. Simplot-Ryl, "From theory to practice: topology control in wireless sensor networks," in *10th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2009)*. New Orleans, Louisiana: ACM, May 2009, pp. 347–348.
- [18] C. AS, "Chipcon SmartRF CC2420, 2.4 GHz IEEE 802.15.4/Zigbee-ready RF Transceiver datasheet," 2006.
- [19] N. Chohan, "Hardware Assisted Compression in Wireless Sensor Networks," Department of Computer Science, University of California, Santa Barbara, Technical Report, 2008.