Energy-Efficient Aggregate Query Evaluation in Sensor Networks

Zhuoyuan Tu and Weifa Liang

Department of Computer Science,
The Australian National University,
Canberra, ACT 0200, Australia
{zytu, wliang}@cs.anu.edu.au

Abstract. Sensor networks, consisting of sensor devices equipped with energy-limited batteries, have been widely used for surveillance and monitoring environments. Data collected by the sensor devices needs to be extracted and aggregated for a wide variety of purposes. Due to the serious energy constraint imposed on such a network, it is a great challenge to perform aggregate queries efficiently. This paper considers the aggregate query evaluation in a sensor network database with the objective to prolong the network lifetime. We first propose an algorithm by introducing a node capability concept that balances the residual energy and the energy consumption at each node so that the network lifetime is prolonged. We then present an improved algorithm to reduce the total network energy consumption for a query by allowing group aggregation. We finally evaluate the performance of the two proposed algorithms against the existing algorithms through simulations. The experimental results show that the proposed algorithms outperform the existing algorithms significantly in terms of the network lifetime.

1 Introduction

Wireless sensor networks have attracted wide attention due to their ubiquitous surveillant applications. Recent advances in microelectronical technologies empower this new class of sensor devices to monitor information in a previously unobtainable fashion. Using these sensor devices, biologists are able to obtain the ambient conditions for endangered plants and animals every few seconds. Security guards can detect the subtle temperature variation in storage warehouses in no time. To meet various monitoring requirements, data generated by the sensors in a sensor network needs to be extracted or aggregated. Therefore, a sensor network can be treated as a database, where the sensed data periodically generated by each sensor node can be treated as a segment of a relational table. During each time interval, a sensor node only produces a message called a tuple (a row of the table). An attribute (a column of the table is either the information about the sensor node itself (e.g., its id or location), or the data detected by this node (e.g., the temperature at a specific location). There is a special node in the network called the *base station* which is usually assumed to have constant energy supply. The base station is used to issue the queries by users and collect the aggregate results for the whole network. In such a sensor

X. Jia, J. Wu, and Y. He (Eds.): MSN 2005, LNCS 3794, pp. 31–41, 2005.

network, users simply specify the data that they are interested in through the various SQL-like queries as follows, and the base station will broadcast these queries over the entire network.

SELECT	{attributes, aggregates}
FROM	sensors
WHERE	condition-of-attributes
GROUP BY	{attributes}
HAVING	condition-of-aggregates
DURATION	time interval

To respond to a user aggregate query, the network can process in either centralized or in-network processing manner. In the centralized processing, all the messages generated by the sensor nodes are transmitted to the base station directly and extracted centrally. However, this processing is very expensive due to the tremendous energy consumption on the message transmission. By virtue of the autonomous, full-fledged computing ability of sensor nodes, the collected messages by each sensor node can also be filtered or combined locally before transmitted to the base station, which is called *in-network aggregation*. In other words, a tree rooted at the base station and spanning all the sensor nodes in the network will be constructed for the data aggregation. Data collected by each node is aggregated before being transmitted to its parent. Ultimately the aggregate result will be relayed to the base station. To implement data aggregation, each node in the routing tree will be assigned into a group according to the distinct value of a list of group by attributes in a SQL-like query. Messages from different nodes are merged into one message at an internal node if they belong to the same group [10]. For example, if we pose a query of "the average temperature in each building", each sensor node will first generate its own message and collect the messages from its descendants in the tree, and then use `SUM` and `COUNT` functions in SQL to compute the average temperature for each group (each building) before forwarding the result to its parent. In the end, all the messages in the same building will be merged into one message so that the transmission energy consumption will be dramatically reduced. Therefore, the number of messages finally received by the base station is equal to the number of buildings.

Related work. *Network lifetime* is of paramount importance in sensor networks, because one node failure in the network can paralyse the entire network. Network lifetime of a wireless sensor network can thus be defined as the time of the first node failure in the network [1].

To improve the energy efficiency and prolong the network lifetime, several existing protocols for various problems have been proposed in both ad hoc networks and sensor networks [1, 2, 3, 4, 5, 6, 7, 11]. For example, in ad hoc networks Chang and Tassiulas [1, 2] realized a group of unicast requests by discouraging the participation of low energy nodes. Kang and Poovendran [6] provided a globally optimal solution for broadcasting through a graph theoretic approach. While in sensor networks, Heinzelman *et al* [3] initialized the study of data gathering by proposing a clustering protocol `LEACH`, in which nodes are grouped into a number of clusters. Within a cluster, a node is chosen as the cluster head which will be used to gather and aggregate the data for other members and forward the aggregated result to the base station directly. Lindsey and Raghavendra [7] provided an improved protocol PEGASIS using a chain concept, where all the nodes in the network form a chain and one of the nodes is chosen as the chain head in turn to report the aggregated result to the base station. Tan and Kórpeoğlu [11] provided a protocol PEDAP for the data gathering problem, which constructs a minimum spanning tree (MST) rooted at the base station to limit the total energy consumption. Kalpakis *et al* [5] considered a generic data gathering problem with the objective to maximize the network lifetime, for which they proposed an integer program solution and a heuristic solution.

This paper provides the evaluation of an aggregate query in a sensor network with an objective to prolong the network lifetime. The pervasive way to do this is to apply the in-network aggregation to proceed the query evaluation, which has been presented in [8], where the information-directed routing is proposed to minimize the transmission energy consumption while maximizing data aggregation. Yao and Gehrke [12] have generated efficient query execution plans with in-network aggregation, which can significantly reduce resource requirements. Apart from these, the query semantics for efficient data routing has been considered in [9] to save transmission energy, in which a semantic routing tree (SRT) is used to exclude the nodes that the query does not apply to. Furthermore, group aggregation has been incorporated into the routing algorithm GaNC in [10], where the sensor nodes in the same group are clustered along the same routing path with the goal of reducing the size of transmitted data. However, an obvious indiscretion in some of the routing protocols, such as MST and GaNC, is that a node is chosen to be added into the tree without taking into account its residual energy during the construction of the routing tree. As a result, the nodes closer to the root of the routing tree will exhaust their energy rapidly due to the fact that they serve as relay nodes and forward the messages for their descendants in the tree. Thus, the network lifetime is shortened.

Our contributions. To evaluate an aggregate query in a sensor network, we first propose an algorithm by introducing the node capability concept to balance the residual energy and the energy consumption of each node in order to prolong the network lifetime. We then present an improved algorithm which allows the group aggregation to reduce the total energy consumption. We finally conduct the experiments by simulation. The experimental results show that the proposed algorithms outperform the existing ones.

The rest paper is organized as follows. Section 2 defines the problem. Section 3 introduces the node capability concept and a heuristic algorithm. Section 4 presents an improved algorithm. Section 5 conducts the experiments. Section 6 concludes.

2 Preliminaries

Assume that a sensor network consists of n homogeneous energy-constrained sensor nodes and an infinite-energy-supplied base station s deployed over an interested region. Each sensor periodically produces sensed data as it monitors its vicinity. The communication between two sensor nodes is done either directly (if they are within the transmission range of each other) or through the relay nodes. The network can be modeled as a directed graph $M = (N, A)$, where N is the set of nodes with $|N| = n+1$ and there is a directed edge $\langle u, v\rangle$ in A if node v is within the transmission range of u. The energy consumption for transmitting a m-bit message from u to v is modeled to be $md^{\alpha}_{v,u}$, where $d_{v,u}$ is the distance from u to v and α is a parameter that typically takes on a value between 2 and 4, depending on the characteristics of the communication medium. Given an aggregate query issued at the base station, the problem is to evaluate the query against the sensor network database by constructing a spanning tree rooted at the base station such that the network lifetime is maximized. We refer to this problem as *the lifetime-maximized routing tree problem* (`LmRTP` for short).

3 Algorithm `LmNC`

In this section we introduce the node capability concept and propose a heuristic algorithm called the `Lifetime-maximized Network Configuration (LmNC)` for `LmRTP` based on the capability concept.

Capability concept. Given a node v, let $p(v)$ be the parent of v in a routing tree. The energy consumption for transmitting a m-bit message from v to $p(v)$ is $E_c(v, p(v)) = md^{\alpha}_{v,p(v)}$, where $d_{v,p(v)}$ is the distance between v and $p(v)$. Let $E_r(v)$ be the residual energy of v before evaluating the current query. Assume that the length of the message sensed by every node is the same (m-bit), then the capability of node v to $p(v)$ is defined as

$$C(v, p(v)) = E_r(v)/E_c(v, p(v)) - 1 = E_r(v)/md^{\alpha}_{v,p(v)} - 1. \tag{1}$$

If v has k descendants in the routing tree, then the energy consumption at v to forward all the messages (its own generated message and the messages collected from its descendants) to its parent $p(v)$ will be $(k+1)md^{\alpha}_{v,p(v)}$, given that there is no data aggregation at v. If after this transmission, v will exhaust its residual energy, then $E_r(v) = (k + 1)md^{\alpha}_{v,p(v)}$. From Equation (1), it is easy to derive that $k = E_r(v)/md^{\alpha}_{v,p(v)} - 1 = C(v, p(v))$. So, if there is no aggregation at v, the capability of node v to $p(v)$, $E_c(v, p(v))$, actually indicates the maximum number of descendants that it can support by its current residual energy.

Algorithm description . Since a node with larger capability can have more descendants in the routing tree (if data aggregation is not allowed), it should be placed closer to the tree root to prolong the network lifetime. Based on this idea, we propose an algorithm `LmNC`, where each time a node with the maximum capability is included into the current tree. Thus, the nodes are added one by one until all the nodes are included in the tree. The motivation behind this algorithm is that adding the node with the maximum capability innately balances the node residual energy $E_r(v)$ and the actual energy consumption for transmitting a message to its parent $E_c(v, p(v))$ (as the definition of the node capability), so that the network lifetime is prolonged. Specifically, we denote by T the current tree and V_T the set of nodes included in T so far. Initially, T only includes the base station, i.e. $V_T = \{s\}$. Algorithm LmNC repeatedly picks a node v ($v \in V - V_T$) with maximum capability to u ($u \in V_T$) and adds it into T with u as its parent. The algorithm continues until $V - V_T = \emptyset$. The detailed algorithm is given below.

Algorithm. Lifetime_Efficient_Network_Configuration (G, E_r)
/* G is the current sensor network and E_r is an array of the residual energy of the nodes */
begin
    $V_T \leftarrow \{s\}$; /* add the base station into the tree */
    $Q \leftarrow V - V_T$; /* the set of nodes which is not in the tree*/
    while $Q \neq \emptyset$ do
        $C_{\max} \leftarrow 0$; /* the maximal capability of nodes in the tree */
        for each $v \in Q$ *and* $u \in V_T$ do
            compute $C(v, u)$;
            if $C_{\max} < C(v, u)$
                then $C_{\max} \leftarrow C(v, u)$;
                    added_node $\leftarrow v$;
                    temp_parent $\leftarrow u$;
        $p($*added_node*$) \leftarrow$ *temp_parent*;
        /* set the parent for the node with maximum capability */
        $V_T \leftarrow V_T \cup \{$*added_node*$\}$; /* add the node into tree */
        $Q \leftarrow Q - \{$*added_node*$\}$;
end.

Note that, although there have been several algorithms for LmRTP considering the residual energy of nodes during the construction of the routing tree (including [1]), they failed to consider the actual transmission energy consumption from a node to its parent. This can be illustrated by the following example. Assume that there is a partially built routing tree and a number of nodes to be added into the current tree. Node v_i has the maximum residual energy among the nodes out of the tree, while the distance between v_i and its parent is much longer than that between another node v_j and its parent. Now, if node v_i is added into the tree, it will die easily in the further tree construction because of the enormous transmission energy consumption from v_i to its parent. Therefore, although v_i has more residual energy than v_j at the moment, the maximum number of the messages transmitted by v_i to its parent is less than that by v_j. We thus conclude that the lifetime of node v_i is shorter than that of node v_j.

4 Improved Algorithm LmGaNC

Although algorithm LmNC manifests the significant improvement on the network lifetime for LmRTP, the total energy consumption for each query is hardly considered during the construction of the routing tree. Because the node with maximum capability may be far away from its parent and the excess transmission energy consumption by the node will be triggered. In this section we present an improved algorithm called `Lifetime-maximized Group-aware Network Configuration (LmGaNC)` by allowing group aggregation to reduce the total energy consumption.

Algorithm description. Since group aggregation is able to combine the messages from the same group into one message, incorporating the nodes of the same group into a routing path will reduce the energy consumption and maximize the network lifetime, because the messages drawn from these nodes will contain fewer groups. With this idea Sharaf *et al* provided a heuristic algorithm (in [10]) to construct an energy-efficient routing tree. Further incorporating this idea into algorithm `LmNC`, we propose an improved algorithm `LmNC` as follows.

Algorithm. Lifetime_Efficient_Network_Configuration (G, E_r)
/* G is the current sensor network and E_r is an array of the residual energy of the nodes */
begin

$V_T \leftarrow \{s\}$; /* add the base station into the tree */
$Q \leftarrow V - V_T$; /* the set of nodes which is not in the tree*/
while $Q \neq \emptyset$ do
  $C_{\max} \leftarrow 0$; /* the maximal capability of nodes in the tree */
  for each $v \in Q$ and $u \in V_T$ do
    compute $C(v, u)$;
    if $C_{\max} < C(v, u)$
      then $C_{\max} \leftarrow C(v, u)$;
        $added\_node \leftarrow v$;
        $temp\_parent \leftarrow u$;
  $p(added\_node) \leftarrow temp\_parent$;
  /* set the parent for the node with maximum capability */
  $d_{min} \leftarrow \infty$; /* minimum distance to choose */
  for each $u' \in V_T$ and $u' \neq temp\_parent$ do
    if $group\_id(u') = group\_id(added\_node)$ and $d_{added\_node,u'} < d_{min}$
        and $d_{added\_node,u'} \leq df * d_{added\_node,temp\_parent}$
      then $p(added\_node) \leftarrow u'$;
  $V_T \leftarrow V_T \cup \{added\_node\}$; /* add the node into tree */
  $Q \leftarrow Q - \{added\_node\}$;

end.

Algorithm `LmGaNC` is similar to algorithm `LmNC`. The difference is that, during the construction of the routing tree, a child with the maximum capability chosen by `LmNC` will keep checking if there is a node in the same group as itself in the current tree in terms of `LmGaNC`. We call this node a better parent. If yes, the child will switch to this better parent. If there are more than one better parent to choose from, the closest one will be chosen. Notice that choosing a better parent far away will cause the extra transmission energy consumption. So, a concept of $distancefactor$ (*df*) is employed, which is the upper bound of the distance between a child and its selected parent. For example, if *df*=1.5, then we only consider the parent whose distance to the child is at most $df * d_{v,u} = 1.5 d_{v,u}$, where $d_{v,u}$ is the distance between v and its current parent u. Energy reduction brought by algorithm `LmGaNC` is demonstrated by the following example. In Figure 1, we have a partially built routing tree (see Fig. 1(a)).

Assume that black nodes 2 and 7 belong to Group 1, shaded node 6 belongs to Group 2, and the rest belong to Group 3. The numbers of messages are as shown in the figure (depending on the number of various groups in the subtree). Under algorithm `LmNC`, shaded node 8 (in Group 2) has maximum capability to its parent node 7 (see Fig. 1 (b)). In order to forward one message originally from node 8, all the nodes in the path from node 8 to the root, except the root 1, have to consume extra energy for this transmission. While the improved algorithm `LmGaNC` allows node 8 to switch to a better parent (node 6) which is in the same group, assuming without violation of the distance factor. As a result, none of the nodes, except node 8 itself, needs to forward one extra message for node 8, so that energy can be saved. Here, after applying group aggregation,

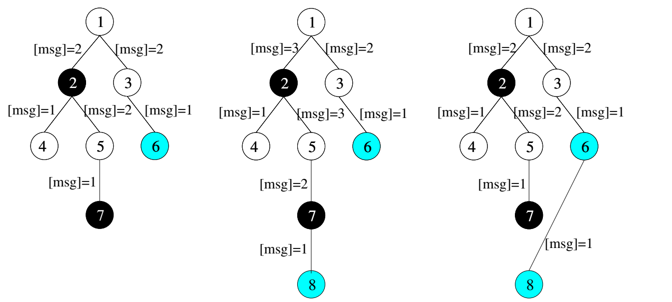


Fig. 1. Benefit of algorithm `LmGaNC`

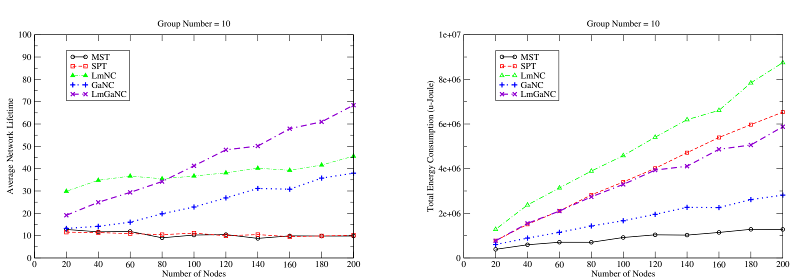
a node capability indicates the number of its descendants in different groups (excluding the group of the node itself) rather than the total number of its descendants, because the messages from the descendants in the same group can be merged into one message only.

Effect of conditional data aggregation. The Group-By clause can divide an aggregate query's result into a set of groups. Each sensor node is assigned to one group according to its value of Group-By attributes. In practice, however, this clause may not be enough to answer the query like "what is the average temperature in each room at level 5". To match the query condition, we normally employ the *where* clause originated from the SQL language to further reduce the energy consumption delivered by algorithm `LmGaNC`. So, in the above case, the condition clause "WHERE Level_no.=5" will be imposed on the query. Before each sensor node transmits its sensed data to its parent, it will check whether the data matches the query condition. If not, then the node just transmits a bit of notification information with value 0 rather than its original data to its parent, so that its parent will not keep waiting for the data from that child. This is especially true under the aggregation schema of Cougar [12], where each node holds a waiting list of its children and will not transmit its data to its parent until it hears from all the nodes on the waiting list. Since the size of the data is shrunk into only 1 bit given the mismatch, the total transmission energy consumption will be further reduced. However, even if a node matches the query condition, whether its residual energy can afford the message transmission is still questionable. One possible solution for this is that the node checks if it has sufficient residual energy to complete the transmission. If not, it will send a bit of notification information with value 1 instead of its original data to its parent to indicate the insufficiency of its residual energy.

5 Simulation Results

This section evaluates algorithms `LmNC` and `LmGaNC` against the existing algorithms including `MST (Minimum Spanning Tree)`, `SPT (Shortest Path Tree)` and `GaNC`. The experimental metrics adopted are the network lifetime and the total energy consumption, based on different numbers of groups, various distance factors, and with and without the *where* condition clause. We assumed that the network topologies are randomly generated from the *NS-2* network simulator with the nodes distributed in a $100 \times 100\ m^2$ region and each sensor is initially equipped with 10^5 *μ-Joules* energy. For each aggregate query, we assign a "Group_id" for each node randomly and take the average of the experimental results from 30 distinct network topologies for each network size.

Performance analysis of the proposed algorithms. Before we proceed, we reproduce an existing heuristic algorithm called `GaNC` [10] for the concerned problem, which will be used as the benchmark. Algorithm `GaNC` with the group aggregation concept is derived from a simple `First-Heard-From (FHF)` protocol where the nodes always select the first node from which they hear as their parents after the query specification is broadcast over the network. The main difference between `GaNC` and `FHF` is that the child under `GaNC` can change to a better parent in the same group within the given distance factor. The simulation results in Figure 2(a) show that the network lifetimes delivered by algorithms `LmNC` and `LmGaNC` significantly outperform the ones delivered by `MST, SPT` and

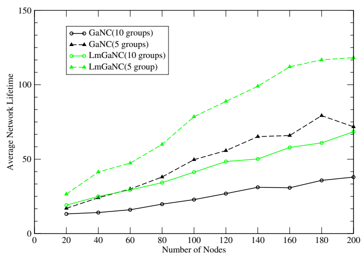


(a) Network Lifetime (b) Total Energy Consumption

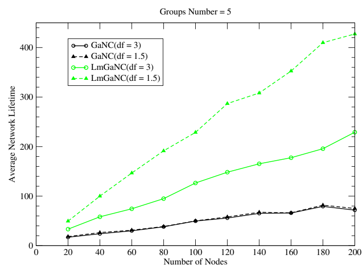
Fig. 2. Performance comparison among various algorithms

GaNC. Figure 2(b) shows that algorithm LmGaNC gracefully balances the total energy consumption of LmNC with a slight shortening on the network lifetime when the number of nodes in the network is less than 80.

Sensitivity to the number of groups. In comparison to algorithm GaNC, the average network lifetime under LmGaNC is more sensitive to the number of groups. Figure 3(a) indicates both algorithms manifest their lifetime improvements by 50% approximately, when the number of groups is decreased from 10 to 5. The reasons behind is as follows. On one hand, fewer groups mean that more sensor nodes will be in the same group, and thus the possibility of message suppression under group aggregation will be enhanced. On the other hand, fewer groups make a child node have more chances to switch to a better parent in the same group, so less transmission energy will be consumed.



(a) Different Numbers of Groups

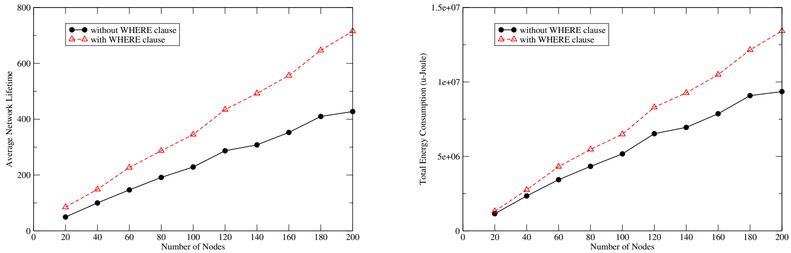


(b) Different Distance Factors

Fig. 3. Performance comparison LmGaNC vs GaNC

Sensitivity to the distance factor. As discussed earlier, the distance factor is introduced to limit the maximum distance that is acceptable when a child node switches to a better parent in the same group. It avoids unnecessary energy dissipation resulting from this switching. As such, the smaller the distance factor is, the less the energy dissipation will be, therefore, the longer the network can endure. Figure 3(b) shows that when distance factor is decreased from 3 to 1.5, algorithm LmGaNC exhibits its sensitivity immediately and the network lifetime is significantly prolonged, while GaNC reacts much more rigidly.

Sensitivity to the where condition clause. The experiments here aim to further reduce the energy consumption of evaluating an aggregate query through allowing the nodes that mismatch the query condition to send a 1-bit notification to their parents instead of the sensed data. Figure 4 (a) and (b) illustrate the effects of the *where* clause in an aggregate query on both the network lifetime and the total energy consumption. The experimental results show that the network lifetime increases by more than 50% while the total energy consumption only goes up by around 25% under LmGaNC.



(a) Network Lifetime (b) Total Energy Consumption

Fig. 4. Performance of `LmGaNC` with the **`where`** condition clause

6 Conclusions

This paper considered the aggregate query evaluation in a senor network database by exploring the node capability concept. Based on this concept we first proposed a heuristic algorithm to prolong the network lifetime, then presented an improved algorithm by incorporating group aggregation to reduce the total energy consumption. We finally conducted experiments to evaluate the performance of the proposed algorithms against those of the existing ones. The experimental results showed that the proposed algorithms outperform the existing algorithms.

Acknowledgment. It is acknowledged that the work by the authors was supported by a research grant from the Faculty of Engineering and Information Technology at the Australian National University.

References

1. J-H Chang and L. Tassiulas. Energy conserving routing in wireless ad hoc networks. *Proc. INFOCOM'00*, IEEE, 2000.
2. J-H Chang and L. Tassiulas. Fast approximate algorithms for maximum lifetime routing in wireless ad hoc networks. *IFIP-TC6/European Commission Int'l Conf.*, Lecture Notes in Computer Science, Vol. 1815, pp. 702–713, Springer, 2000.
3. W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. *Proc. 33th Hawaii International Conference on System Sciences*, IEEE, 2000.
4. C. Intanagonwiwat, D. Estrin, R. Govindan, and J. Heidemann. Impact of network density on data aggregation in wireless sensor networks. *Proc. 22nd International Conference on Distributed Computing Systems (ICDCS'02)*, IEEE, 2002.
5. K. Kalpakis, K. Dasgupta and P. Namjoshi. Efficient algorithms for maximum lifetime data gathering and aggregation in wireless sensor networks. *Computer Networks*, Vol. 42, pp. 697–716, 2003.
6. I. Kang and R. Poovendran. Maximizing static network lifetime of wireless broadcast ad hoc networks. *Proc. ICC'03*, IEEE, 2003.

7. S. Lindsey and C. S. Raghavendra. PEGASIS: Power-efficient gathering in sensor information systems. *Proc. Aerospace Conference*, IEEE, pp. 1125–1130, 2002.
8. J. Liu, F. Zhao, and D. Petrovic. Information-directed routing in ad hoc sensor networks. *Proc. 2nd international conference on wireless sensor networks and applications*, ACM, 2003.
9. S. Madden, M. Franklin, J. Hellerstein and W. Hong. The design of an acquisitional query processor for sensor networks. *Proc. of SIGMOD'03*, ACM, 2003.
10. M. A. Sharaf, J. Beaver, A. Labrinidis and P. K. Chrysanthis. Balancing energy efficiency and quality of aggregate data in sensor networks. *J. VLDB*, Springer, 2004.
11. H. Ó. Tan and İ. Kórpeoğlu. Power efficient data gathering and aggregation in wireless sensor networks. *ACM SIGMOD Record*, Vol. 32, No. 4, pp. 66–71, 2003.
12. Y. Yao and J. Gehrke. Query processing in sensor networks. *Proc. 1st Biennial Conf. Innovative Data Systems Research (CIDR'03)*, ACM, 2003.