A Technique for Power-Aware Query-Informed Routing in Support of Long-Duration Queries for Sensor Networks

Zhiguo Zhang, Sol M. Shatz

***Abstract*--**In-network aggregation is a useful technique to mitigate the limited power problem in sensor networks. In order to further reduce power usage, many efforts have focused on the application layer. In contrast, our work integrates the application and network layers with a focus on generation of a routing topology for sensor networks. We present a grouping technique to construct a routing tree topology based on information contained within specific queries. This "query informed" method enables the query process to be efficient in terms of energy consumption. The technique is aimed at applications that require long-duration queries, and is described in terms of algorithmic behavior, simulation experiments, and comparative analysis to conventional techniques.

I. INTRODUCTION

Sensor networks consist of small sensor nodes with sensing, computation, and communication capabilities. Such networks are a promising platform for many applications ranging from environmental control, warehouse inventory, and healthcare, to military environments. Specific applications include wildlife tracking [6], habitat monitoring [8], and etc. Recent advances in hardware development have enabled widespread research in problems associated with sensor networks. Sensor nodes are getting smaller, cheaper, and more powerful. However, there are still many crucial problems in the design of sensor network systems, mostly stemming from limited storage, low network bandwidth, poor inter-node communication, limited computational ability, and low power capacity. A significant problem for the application of sensor networks is the energy constraint, which is a major objective in recent research on the design of sensor network systems, and a motivation for this work.

There are two key contributing factors of energy consumption. The first is the energy consumed by nodes when they are in an inactive state (not communicating with other nodes) [1], and the second is the energy consumed by nodes specifically due to transmission of messages (a high energy-cost operation) [3]. To address the first case, the objective is to find optimal wakeup-sleep schemes [15]. To address the second case, the objective is to reduce inter-node communication [10]. In this paper we focus on this second case. We propose a scheme to alleviate the problem of limited power by exploiting in-network aggregation query processing. Data processing is pushed inside the sensor network whenever such computation can reduce the amount of data that must be transmitted inside the network.

It is generally accepted that access to data in a sensor network should be declarative [2], meaning that users formulate queries to access the data of interest. Various techniques [7, 13, 14] have been introduced for data management and query processing over sensor networks based on in-network aggregation query processing and declarative query languages. Yao and Gehrke [14] provide a simple form of a SQL-like declarative query language for sensor networks.

As an example, in a nuclear power plant monitoring system we may employ the following query to monitor the amount of harmful radiation:

```
SELECT room, AVG(radiation) FROM sensordb WHERE
building = ERF GROUP BY room HAVING AVG(radiation)
> 100 DURATION 30 days EVERY 1 minute
```

In-network aggregation query processing is based on using a tree topology to route data information from every node. The objective is to perform aggregation in intermediate nodes of the routing tree to reduce the amount of data that must be transmitted forward to the root node. It has been observed that queries in a sensor network tend to be aggregation queries.

Aggregation-based research methods have been presented that apply to the query layer of the sensor network to save energy and maintain the quality of data [10]. These methods focus on using tolerances or precisions in every node to let nodes send less data, thus reducing the number of message transmission events and saving energy. From a related but alternate view, in this paper we propose a grouping technique, which integrates the design of the network layer and query layer for the purpose of improving energy efficiency. Our technique focuses on reducing the size of messages routed within the sensor network.

Our basic strategy is to influence the construction of the routing tree topologies by leveraging aggregation groups defined by the queries. This general approach of trying to use query information to form a routing tree has been termed *query-informed routing* [11]. In our specific work, we seek to create a routing tree topology that keeps those nodes that belong to a common aggregation group logically close to each other. This method can save energy because it can reduce the number of aggregation groups in intermediate nodes, which reduces the size of messages that those nodes broadcast. The approach is aimed at applications that require long-duration queries; such as the radiation monitoring

---

Z. Zhang is with the Computer Science department, University of Illinois at Chicago, IL 60607 USA (email: zzhang@cs.uic.edu)
S. M. Shatz is also with the Computer Science department, University of Illinois at Chicago, IL 60607 USA (email: shatz@cs.uic.edu)

example or various search-and-recovery efforts following natural disasters.

Some research works [12] have pointed out that link quality based parent selection can reduce retransmission and save energy for transmission. However, we propose that the choice of parent should include some consideration of information contained in the queries as well. In general, our technique is most applicable in situations in which there are several parents of comparable link quality. A link quality-based parent selection algorithm should be used in conjunction with the grouping technique to pre-filter the set of parents made available to the grouping technique. In the discussion of this paper, we assume all links have the same quality. This technique can be applied together with other application level schemes [10] in sensor networks.

II. BACKGROUD

In sensor networks, data processing is much cheaper than data communication in terms of time and energy consumption. This observation motivated the *directed diffusion* data-dissemination paradigm [5], which first proposed in-network processing of queries. Directed diffusion is data-centric. Data generated by a sensor node is named by attribute-value pairs. Only data that matches some interest will be retrieved from sensor nodes. Directed diffusion enables activation of catching and aggregation inside the sensor network. In directed diffusion, each node can seize a data packet it is forwarding on behalf of another node, do some processing on this packet if applicable, and then forward the newly generated packet up the path to the destination node.

TAG [7] and COUGAR [13] examined the properties of aggregation queries and proposed to use a declarative language based on a routing tree topology in sensor networks, which allows data aggregation at every intermediate node of the tree for algebraic (e.g., average) and distributive (e.g., max/min, count, and sum) aggregations. Each sensor node transmits/forwards packets to its parent node in the tree topology. The method used in [7, 13] is referred to as *in-network aggregation*. Since a tree structure allows local-aggregation at every level of the tree, the total number of messages and the total message size in the tree structure is much less than that in the cluster based structure, where local-aggregation takes place only at cluster heads. In general, in-network aggregation can reduce the power usage by pushing part of the computation into the sensor network.

In previous work [7, 13], the tree topology was formed non-deterministically. However, we can observe that the extent to which aggregation can be exploited in intermediate nodes of a query is affected by the tree topology. For example, consider a topology in which some intermediate node *N* has children nodes that belong to 10 different aggregation groups. In this case node *N* must transmit a 10-tuple sized aggregation message, where each tuple represents the information of an aggregation group, to its parent node in order to propagate information toward the network root. Alternatively, consider a topology in which all children of the intermediate node belong to the same aggregation group. In this case the intermediate node only needs to send a 1-tuple aggregation message to its parent node. So, carefully selecting a routing topology can dramatically reduce the total size of messages transmitted by intermediate nodes.

III. MOTIVATION AND BASIC METHOD

For query processing in sensor networks, there are three phases: disseminating queries into the network, sensing data, and retrieving data from the networks. In disseminating a query into the network, a tree topology will be formed to route the query result to users. In this phase, each node performs two actions: 1) according to the messages it receives, each node decides its own tree level and selects a parent node with respect to the tree topology being created, and 2) the node broadcasts its own id[1] and tree level. Once all nodes in the network have established their levels and parent nodes, the tree topology is defined.

In previous methods [7, 13], sensor nodes select the parent nodes only according to level. The grouping technique being proposed here is motivated by the fact that it is common that queries in a sensor network are aggregation queries (such as COUNT, MAX, MIN, AVERAGE, etc.) using “GROUP BY” or “WHERE” to form aggregation groups according to a specific attribute of the sensor nodes, and there are situations where such queries must remain active over long durations (even the lifetime of the sensor networks), such as the environment monitoring example in the introduction section. The basic idea is to try to force those sensor nodes with the same specific attribute, which is used to “GROUP BY”, to be logically close to each other when forming the tree topology. This will result in partial aggregates being performed as low as possible in the tree topology. Consider the simple example of Figure 1.

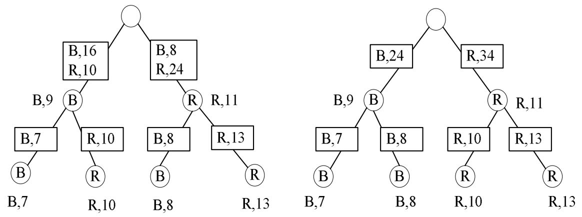

Fig. 1a. Tree formed using conventional method Fig. 1b. Tree formed using grouping method

Figure 1. Query process on different tree topologies.

In Figure 1, each node has two associated attributes, which we call color and value (value could be something like temperature or number of students). For example, the bottom-left node has color=blue and value=7. The boxes represent messages to be transmitted to parent nodes.

Suppose we want to read the total value of the nodes, grouped by color. The query is as follows:

---

[1] We adopt the common assumption that each node in the sensor network has a unique identifier, the node’s id.

SELECT color, SUM(value) FROM sensordb GROUP BY color

This query would be injected into the sensor network starting from some root node. Fig. 1a is an example tree topology that is independent of our grouping technique, while Fig. 1b is a topology that is created by our grouping technique. In Fig. 1a, when the two nodes in level 2 want to send the partial accumulated messages to the root node in level 1, each node needs to send a 2-tuple message. For example, the blue node with value 9 receives two messages and aggregates the values of 7 and 9 (giving 16) since these values are associated with nodes of the same color (i.e., they belong to the same aggregation group). The level-2 node then sends a message that contains two tuples of date: (B,16), (R,10). In contrast, each level-2 node in Fig. 1b needs to send a 1-tuple message to its parent node. From this example, we can see that the tree topology as shown in Fig. 1b will save 1-tuple of data in some messages as compared to the tree shown in Fig. 1a. Note that in Fig. 1a the blue nodes define one aggregation group, even though these nodes are somewhat separated in the routing topology. In contrast, in Fig. 1b the routing topology has been formed so that the blue nodes are clustered in the routing tree, thus defining a Network Group. The case is similar for the red node.

Another example is shown in Figure 2, for queries having WHERE clause. Suppose the query is

SELECT SUM(value) FROM sensordb WHERE color=blue

We can see that the aggregation will be completed at blue node $S_{3,3}$ if the tree topology, as shown in Fig. 2a, is formed by using our grouping technique. Nodes $S_{2,3}$, $S_{2,2}$, and $S_{1,1}$ will only need to transit the 1-tuple result to the user. There are only 7 nodes (including those blue nodes) sending messages and each node only needs to send a 1-tuple message. Other nodes can be in an idle state. However, if sensor nodes select parent nodes non-deterministically, we may end up with a tree topology like that shown in Fig. 2b. In this situation, nearly all the sensor nodes need to receive and transit a 1-tuple message. Blue node $S_{3,4}$ will need to transfer its result through $S_{2,5}$-$S_{1,4}$-$S_{1,3}$-$S_{1,2}$-$S_{1,1}$ to the user; node $S_{4,3}$'s result travels through $S_{4,2}$-$S_{3,1}$-$S_{2,1}$-$S_{1,1}$; $S_{3,3}$'s result goes through $S_{2,3}$, where the result combines with the result obtained from $S_{4,4}$ (which travels through $S_{3,5}$-$S_{2,4}$-$S_{2,3}$) and then moves thru $S_{2,2}$-$S_{1,1}$ to the user. Nearly all nodes take part in transmitting messages. This uses more energy.

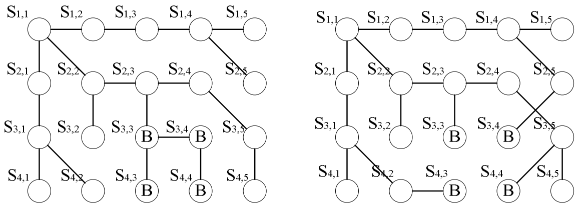

Fig. 2a. Tree topology formed using the grouping technique method

Fig. 2b. Tree topology formed using the conventional method

Figure 2. A processing example for queries having WHERE clause.

IV. DEFINITIONS AND ALGORITHM

From the previous section, we can see that a tree topology formed by our grouping approach can reduce the size of transmitted messages and thus save energy associated with intermediate nodes. How to form such a tree topology is the focus of our grouping technique algorithm. Before describing the details of the algorithm, we first provide some definitions.

The first three terms only relate to the deployment of sensor nodes and a specific query – they are independent of any specific routing tree topology.

Definition 4.1 *Attributes*

Attributes are properties of sensor nodes. For an attribute *A*, the value associated with this attribute at some node *N* is denoted as *N.A*. For example, for some node *P*, *P.temp* may have a value of 75˚F.

Definition 4.2 *Grouping Attribute*

A *Grouping Attribute* (*GA*) is an attribute of sensor nodes based on which the given aggregation query is performed. For example, for the query "SELECT color, SUM(temp) FROM sensordb GROUP BY color", color is the Grouping Attribute.

Definition 4.3 *Network Group*

Given a deployment of sensor nodes, a *Network Group* is a maximal set of sensor nodes that share a common Grouping Attribute value and may connect (communicate) to each other by general broadcast, possibly via a multi-hop path, and all nodes on such a connection path have the same Grouping Attribute value. We denote the Network Group of node *P* as *NG(P)*.

As an example, consider Figure 2. The nodes $S_{3,3}$, $S_{3,4}$, $S_{4,3}$ and $S_{4,4}$ define a Network Group in the 4 × 5 grid deployments for both Fig. 2a and Fig. 2b, since these nodes can connect by broadcast (independent of the routing trees that are shown in Fig. 2) and all nodes along the broadcast paths have the same Grouping Attribute value, "Blue".

The next two terms relate to a specific routing tree topology. The first one, Topology Group, is basically analogous to a Network Group, but is defined with respect to a specific routing tree.

Definition 4.4 *Topology Group*

Given a deployment of sensor nodes and a specific routing tree topology, a *Topology Group* is a maximal set of sensor nodes that share a common Grouping Attribute value and are connected to each other by the routing tree, possibly via a multi-hop path, and all nodes on such a connection path have the same Grouping Attribute value.

As an example, consider again Figure 2. The nodes $S_{3,3}$, $S_{3,4}$, $S_{4,3}$ and $S_{4,4}$ define a Topology Group in Fig. 2a but not in Fig. 2b, because these nodes form a subtree of the routing tree in Fig. 2a but not in Fig. 2b.

Definition 4.5 *Group-Root Node*

Given a Topology Group, the *Group-Root Node* is the node with the minimum tree level (i.e., the node closest to the root of the routing tree).

As an example, in Fig. 2a, $S_{3,3}$ is the Group-Root Node of the Topology Group formed by $S_{3,3}$, $S_{3,4}$, $S_{4,3}$ and $S_{4,4}$. Node $S_{3,3}$ has a tree level value of 4.

Given a routing tree topology, the total size of messages used in any retrieval epoch increases with the total number of Topology Groups that exist in the routing tree. In other words, by reducing the number of Topology Groups, we reduce the total size of messages used during query processing. For example, in Figure 1a, the tree topology has four Topology Groups, while the tree topology in Figure 1b only has two Topology Groups, and the total numbers of message tuples required are 8 and 6, respectively. Therefore, for a given deployment and query, we seek a tree topology that minimizes the number of Topology Groups.

Observation 1: Given a deployment, the number of Topology Groups for a tree topology is greater than or equal to the number of Network Groups of the deployment. The number of Network Groups in a deployment is the minimum number of Topology Groups among all the tree topologies that can be formed in the deployment.

With the previous observation in mind, the goal of our grouping technique is to ensure that the formed tree topology will minimize the number of Topology Groups. As we have already discussed, the tree topology is formed during the process of disseminating a query into the sensor network. In the conventional method, each node selects its parent node from its neighbor nodes, and sets its own level with respect to the tree topology being formed. Then, the node broadcasts its own topology information. In the conventional method, the parent selection process is nondeterministic (as in the techniques in [13]). In this case, it is likely that nodes in the same Network Group belong to different Topology Groups. Note that this is the case for $S_{3,3}$ and $S_{3,4}$ in Fig. 2b. To force the tree topology to have the property that there exists one Topology Group for each Network Group, we need to create a deterministic parent selection algorithm. A node will not select its parent node arbitrarily from its neighbor nodes, but instead will select as its parent a node that is in the same Network Group as itself, if there is such a node.

To help put our algorithm in context, we first present a basic parent selection algorithm based on the conventional method. See Algorithm 1. *NT* denotes a neighbor table that is maintained at each node. A neighbor of a node *P* is a node from which *P* can directly receive broadcast messages. Each node updates its NT whenever it receives a broadcast message.

Algorithm 1. Parent selection procedure of Conventional Method

```
1:  Procedure:
2:  if neighbor table (NT) has changed since last period then
3:      if P has not selected a parent node then
4:              select the neighbor Q with minimum tree level.
5:              Set Q as P's parent node.
6:              P.treelevel = Q.treelevel + 1.
7:              broadcast P's topology information.
8:      end if  // end if line 3
9:  end if      // end if line 2
10: End Procedure
```

For the above procedure, the broadcast messages that are carried through the network during the query dissemination phase contain basic information such as the query itself, and the id and tree level of the sender node. To implement our group-oriented tree formation procedure, each broadcast message is extended to include 1) the Grouping Attribute value of the sender node, and 2) the *group level* and *group id* of the sender node, which are defined as the tree level and id of the Group-Root Node associated with the sender's Topology Group. For example, in Fig. 2a, the group level of node $S_{4,4}$ is 4 (the tree level of Group-Root node $S_{3,3}$), although its tree level is 6, and the group id of $S_{4,4}$ is $S_{3,3}$.

The parent selection algorithm for our grouping technique is presented as Algorithm 2. As with Algorithm 1, the algorithm is periodically executed by each node *P* in the sensor network.

Each node *P* periodically executes the following procedure:

Algorithm 2. Parent selection procedure of Grouping technique

```
1:  Procedure:
2:  if neighbor table (NT) has changed since last period then
3:      if any neighbor Q in NT such that Q.GA== P.GA then      //  NG(Q) = NG(P)
4:              select the neighbor Q with minimum group level and group id.
5:              if P has not selected a parent or P.grouplevel > Q.grouplevel or
                (P.grouplevel==Q.grouplevelthen and P.groupid > Q.groupid) Then
6:                      set Q as P's parent node.
7:                      P.grouplevel = Q.grouplevel.
8:                      P.treelevel = Q.treelevel + 1.
9:                      P.groupid = Q.groupid.
10:                     broadcast P's topology information.
11:             end if  // end if line 5
12:     else
13:             if P has not selected a parent node then
14:                     select the neighbor Q with minimum tree level.
15:                     Set Q as P's parent node.
16:                     P.treelevel = Q.treelevel + 1.
17:                     P.grouplevel = P.treelevel.
18:                     P.groupid = P.id
19:                     broadcast P's topology information
20:             end if  // end if line 12
21:     end if  // end if line 3
22: end if      // end if line 2
23: End Procedure
```

Observation 2: The tree formed by our grouping technique algorithm minimizes the number of *Topology Groups.*

Proof:

> Assume our algorithm forms a tree topology in which a node *n1* in Network Group *g1* does not form a Topology Group with other nodes in *g1*. Then the parent selected by *n1* (say *p1*) is in a different Network Group, which means no neighbor node of *n1* is in the same Network Group with *n1*. However, since at least one neighbor node of *n1* is in *g1* (since *n1* being in *g1* means that *n1* can receive broadcasts from at least one node in *g1*), by contradiction we conclude that the assumption is false and each Network Group forms a Topology Group in the tree topology formed by our grouping technique. The tree topology formed minimizes the number of Topology Groups. QED

V. SIMULATION AND ANALYSIS

As an initial step toward evaluation of the effectiveness of our grouping technique, we performed some simulation studies using the network simulator tools NS2 [16].

In our simulations, we place a 9 × 9 grid with each node at one grid point. Each node can only directly (one hop) communicate with the immediate surrounding 8 nodes (except those boundary nodes). We use an attribute "color" to represent the grouping attribute. The query used in our experiments is:

SELECT color, AVG(temperature) FROM sensordb GROUP BY color EVERY 1 second Duration 1000 second

We increase the Network Groups but keep Network Groups equal to aggregation groups for simplification, i.e. in our experiment, nodes with same "color" (defining an aggregation group) will belong to exactly one Network Group. Figure 3 shows a comparison of our grouping technique and the conventional method for the average bytes (of broadcast messages) sent by a node.

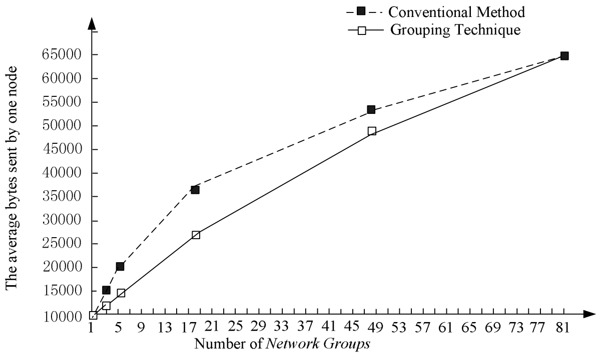

Figure 3. The average bytes of messages sent by each node for tree topologies formed by conventional method and the grouping technique.

Based on a comparison of the two approaches using our example query, we observe that our grouping technique approach always outperforms the conventional approach. The maximum observed improvement of the grouping technique was nearly 35%.

We have noted that the communication cost of a sensor network application is due to two components:

1) Query dissemination cost is the communication cost associated with setting up the routing tree topology. This is a pre-processing cost, which we denote as P.

2) Data retrieval cost is the communication cost associated with retrieving requested data from sensor nodes using the routing tree topology. This is a retrieval cost, which we denote as R.

So, the total communication cost is $C = P + R$.

With our grouping technique, we spend extra communication cost in setting up an efficient routing tree to try and save communication cost when retrieving data using the routing tree topology. The extra communication cost can be viewed as overhead that is necessary in order to set up a tree topology with the group property. We can define the pre-processing cost P as $P = S + \partial$, where S is the communication cost needed by conventional methods to setup the tree topology, and ∂ is the overhead needed by our grouping technique. In the conventional approach each node selects a parent nondeterministically based on broadcast messages received from its neighbors, a node does not broadcast until it has selected a parent, and a node does not change parents once it has selected a parent. Thus, the complexity, in terms of broadcast messages, to form the routing tree is O(n), where n is the number of nodes in the network. For our grouping technique, a node may need to reselect a parent node and thus rebroadcast a message. With the assumption that each node has k neighbors, the worst case complexity to form the routing tree is O(kn), but since k is dependent on the broadcast range of a node, not the number of nodes in a network, and $k << n$ (a reasonable assumption for large-scale sensor networks), the complexity is still O(n). Therefore the overhead ∂ is not an amount that increases the order of complexity.

Since transmission cost is proportional to the total size of messages sent by each node. The relationship between the number of Network Groups and the cost of communication in each query epoch can be characterized as in Figure 4, which illustrates theoretical performance trends associated with our grouping technique and with the conventional technique. A key observation is that the group technique generally provides a topology that is guaranteed to use fewer messages (power) than the conventional approach, i.e., $d \geq 0$. The exception to this case occurs when the number of Network Groups is equal to 1 or the number of nodes in the network – in this case, our grouping technique degenerates to the same nondeterministic behavior as the conventional technique, so the expected value of d in this case is 0.

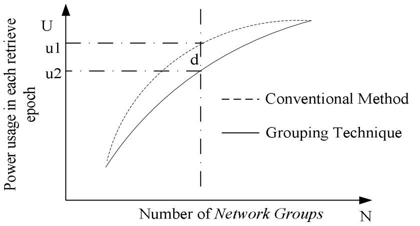

Figure 4. Power usage comparison for tree topologies formed by conventional method and the grouping technique.

Now we can provide an equation for the total data retrieval communication cost R associated with a deployment and query:

$$R = U * (T/e),$$

where U is the cost to route data through the topology tree, T is the value in the DURATION clause of the query and e is the time span in the EVERY clause of the query. Thus, T/e is the total number of retrieval epochs for the source query. By simple substitution, we can express the cost savings due to the grouping technique method as follows:

$$\Delta\text{Cost} = (S + U1*(T/e)) - (S + \partial + U2*(T/e)) = (U1-U2)*(T/e) - \partial = d*(T/e) - \partial$$

This tells us that the grouping technique results in an overall cost savings under the condition that $d * (T/e) > \partial$. We previously discussed the nature of the overhead ∂.

In the equation of ΔCost, d is greater than 0 except when the number of Network Groups is equal to one or equal to the number of nodes in the network. As Figure 4 shows, as the number of Network Groups increases, d increases from 0 then decreases toward 0 as the number of Network Groups approaches the number of nodes in the network. As we noted earlier, when the number of Network Groups is the same as the number of nodes, our grouping technique degenerates to the same nondeterministic behavior as the conventional technique, so the expected value of d in this case is 0.

From the analytical power saving equation, and from simulation experiments, we can make some useful observations regarding the applicability of our grouping technique. In particular, it appears that the technique is most useful in sensor networks with the following characteristics:

1) There are many aggregation groups in the sensor network under some query. Usually, more aggregation groups mean more information tuples should be sent by intermediate nodes, and then more bytes will be saved by the use of our grouping technique.
2) Nodes in the same aggregation group are physically close to each other. If two nodes within the same aggregation group in a deployment cannot communicate directly with each other, our grouping technique algorithm will nearly have no advantage over a conventional non-deterministic algorithm, as observed in our simulation when the number of Network Groups equals 81.
3) A query is used for a long time interval, i.e., there will be many epochs of information retrieval using the same routing tree topology. In this case, more bytes of messages are saved after spending some additional bytes in setup.

In practice, sensor network applications such as those monitoring the average temperature or moisture of each room in a building tend themselves well to our grouping technique. In these *monitoring applications*, sensor nodes in the same aggregation group, such as room, are physically close, and the query computing the average temperature or moisture of a room is constant during the use of the sensor network. In contrast, in typical *detecting applications*, in which queries are like "at time T, send me the number of every four-legged animal in the sensor field", our grouping technique would not be very suitable. In this situation, sensor nodes in the same aggregation group may not be physically close; also, those queries do not last for a long period, and the expense in setting up the routing tree may be significantly larger than the savings gained by retrieving information through the routing tree constructed by our grouping technique.

VI. CONCLUSION AND FUTURE WORK

We introduced a grouping technique for use in in-networking aggregation query processing over sensor networks. The technique is "query-informed" since it applies aggregation group information obtained from queries to the formation of a routing tree topology. This allows for a reduction in the size of messages forwarded by intermediate nodes in the tree topology. Preliminary experimental results indicate that our grouping technique can reduce the data that is transmitted by sensor nods during the phase of sensor query processing.

Since our grouping technique is based on local (neighbor) information, it may not provide a globally optimal routing tree. Thus, we need to characterize the algorithm in comparison to an optimal one, and then develop some performance proofs that define the "optimality" of our approach.

For other future work, we will perform more simulation analysis and seek to further enhance our grouping technique by exploring the use of the technique in the context of multi-query processing in which queries have arbitrary Grouping Attributes. We also want to investigate how to optimize the tree topology formed by the grouping technique to attempt to simultaneously reduce response time and message size.

REFERENCES

[1] A. Cerpa and D. Estrin. ASCENT: Adaptive self-configuring sensor networks topologies. *Proc. of IEEE INFOCOM*, June 2002.
[2] D. Estrin, M. Srivastava, and A. Sayeed. Tutorial on Wireless Sensor Networks. *ACM MOBICOM*, September 2002.
[3] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. *Hawaii International Conference on System Sciences*, January 2000.
[4] J. Hill and D. Culler. Mica: A wireless platform for deeply embedded networks. *IEEE Micro.*, 22(6), 2002.
[5] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. *ACM MOBICOM*, August 2000.
[6] P. Juang, et al. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet. *International Conference on Architectural Support for Programming Languages and Operating Systems*, Oct. 2002.
[7] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. Tag: A tiny aggregation service for ad-hoc sensor networks. *5th Symposium on Operating Systems Design and Implementation*, December 2002.
[8] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson. Wireless sensor networks for habitat monitoring. *First ACM International Workshop on Wireless Sensor Networks and Applications,* Sep. 2002.
[9] C. Schurgers, V. Tsiatsis, and M. Srivastava. STEM: Topology Management for Energy Efficient Sensor Networks. *Proc. IEEE Aerospace Conf., pp. 78-89*, Mar. 2002.
[10] M. A. Sharaf, J. Beaver, A. Labrinidis, and P. K. Chrysanthis. Tina: A Scheme for Temporal Coherency-Aware In-Network Aggregation. *The Third International ACM Workshop on Data Engineering for Wireless and Mobile Access*, Sep. 2003.
[11] A. Woo, S. Madden, and R. Govindan. Networking support for Query Processing in Sensor Networks. *Communications of the ACM*, June 2004/Vol. 47. No.6.
[12] A. Woo, T. Tong, D. Culler. Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks. *ACM Sensys 2003*, Los Angeles, November 2003.
[13] Y. Yao and J. Gehrke. The cougar approach to in-network query processing in sensor networks. *SIGMOD Records*, 31(3), 2002.
[14] Y. Yao and J. Gehrke. Query processing for sensor networks. *Conference on Innovative Data Systems Research*, January 2003.
[15] R. Zheng, J. C. Hou and L. Sha. Asynchronous Wakeup for Ad Hoc Networks. *MobiHoc'03*, June, 2003.
[16] The network simulator tools NS2. http://www.isi.edu/nsnam/ns/