REQUEST: Region-Based Query Processing in Sensor Networks

Dong-Wan Choi and Chin-Wan Chung

Department of Computer Science,
Korea Advanced Institute of Science and Technology(KAIST)
335 Gwahangno, Yuseong-gu, Daejeon, Republic of Korea
dongwan@islab.kaist.ac.kr, chungcw@kaist.edu

Abstract. In wireless sensor networks, node failures occur frequently. The effects of these failures can be reduced by using aggregated values of small groups of sensors instead of the values of individual sensors. However, most existing works have focused on computing the aggregation of all the nodes without grouping. Only a few approaches proposed the processing of grouped aggregate queries. However, since groups in their approaches are disjoint, some relevant regions to the query can be missing. In this paper, we propose *REQUEST*, region-based query processing in sensor networks. A region in our approach is defined as a maximal set of nodes located within a circle having a diameter specified in the query. To efficiently construct a large number of regions covering the entire monitoring field, we build the *SEC* (Smallest Enclosing Circle) index. Moreover, in order to process a region-based query, we adapt a hierarchical aggregation method, in which there is a leader for each region. To reduce the communication cost, we formulate an optimal leader selection problem and transform the problem into the set-cover problem. Finally, we devise a query-initiated routing tree for the communication between the leader and non-leader nodes in a region. In the experimental results, we show that the result of our region-based query is more reliable than that of the query which is based on individual nodes, and our processing method is more energy-efficient than existing methods for processing grouped aggregate queries.

Keywords: Sensor networks, Regional query, Group-by aggregate query.

1 Introduction

Wireless sensor networks are widely used in various environmental monitoring applications. By using these applications, we can find some phenomena of the monitoring area, and detect some events corresponding to a given set of conditions. For example, if farmers could collect the information about the distribution of nutrients in the soil or locations colonized by many insects in real-time, the farmers could predict where and how much water, pesticide, and fertilizer are needed currently[1]. We can collect these information by gathering sensing values from the sensor nodes deployed in the monitoring area.

J.X. Yu, M.H. Kim, and R. Unland (Eds.): DASFAA 2011, Part II, LNCS 6588, pp. 266–279, 2011.

```
select node
from sensors
where t_l < temp < t_u
and h_l < humid < h_u
```

(a) Query based on individual nodes

```
select region
from sensors
group by region(10m)
having t_l < AVG(temp) < t_u
and h_l < AVG(humid)< h_u
```

(b) Region-based query

Fig. 1. Example queries

However, each sensing value can have some noises, as sensor nodes are prone to failure inherently. Moreover, managing a large number of individual sensor nodes is ineffective when only a macro view of the monitoring area is required.

To overcome these problems, we can construct small groups of sensor nodes, and use an aggregated value for each group. Previous works on grouping nodes [7,15] in the sensor networks address the problem by partitioning or clustering nodes with appropriate criteria such as the geographic location. In these works, there can be missing areas since they do not allow groups to overlap.

It is natural to group sensor nodes with regions of the same size. This is because, in the sensor network applications, we are not interested in a node itself, but a region in which the node is located.

Considering the above grouping method, we propose the region-based query processing method in sensor networks (hereafter called "*REQUEST*"). In REQUEST, the primitive processing unit is a region instead of a node. In addition, regions can overlap to cover every possible region generated by sensor nodes. Fig 1 shows example queries that find nodes or regions with certain temperature and humidity. Note the difference between our proposed region-based query(Fig 1(b)) and the query that is based on individual nodes(Fig 1(a)).

There are some challenging problems in REQUEST. First, since regions overlap and the number of regions is fairly large, it is not trivial to efficiently construct regions with a specified size in the query. A naive approach is to move a circle representing a region as a certain step size. However, this approach is too inefficient, and it is not easy to find appropriate step size. To solve this problem, we create the *SEC* (Smallest Enclosing Circle) index structure in the preprocessing phase, and construct regions by using the SEC index.

Second, the communication costs of forwarding sensing values and aggregated values can be considerably high due to a large number of regions. In order to process the region-based queries energy-efficiently, we use a hierarchical aggregation method [6] as a basic processing scheme. In the hierarchical aggregation method, we have a leader node and several non-leader nodes in each group, and the aggregation of each group is computed locally in a group. Since there are numerous regions in our environment, it is more beneficial to calculate an aggregated value for each region as early as possible. By doing so, we can reduce the size and the total number of messages to send to the basestation. Moreover, to minimize communication costs while gathering aggregated values, we need an algorithm that selects optimal leader nodes efficiently in terms of energy consumption. To determine optimal

leader nodes, we consider some criteria such as the hop counts between nodes, the size of messages, and the number of regions covered. Based on these criteria, we formulate a leader selection problem, and design an algorithm that uses the idea of transformation into the set-cover problem.

Finally, we need a topology for communication between leader nodes and non-leader nodes. TAG-based global routing tree [10] is not appropriate for intra-region communication, since it is constructed in order for the basestation to collect the data of the entire network. For intra-region communication, it is required to construct a tree in order for the leader node to collect the data of non-leader nodes inside the region. Therefore, for each leader node, we build a new routing tree whose root node is the leader node, called query-initiated routing tree.

Our contributions in this paper are as follows:

- We propose region-based queries, a new type of queries which use a region as a primitive data unit. This type of queries are useful especially when individual sensor readings are not reliable or only a rough view of the monitoring environment is required. By adjusting the sizes of regions in the query, we can collect the data in various degrees of circumstantiality. Moreover, since regions can be overlapped in our approach, we can avoid that some important regions are missing. To the best of our knowledge, the region-based query is the first type of grouped aggregate query in which groups can be overlapped.
- We propose algorithms to efficiently process region-based queries. To construct regions efficiently in real time, we present an algorithm that utilize the Smallest Enclosing Circle index. In addition, we address the optimization problem for leader selection, and propose a corresponding algorithm to the problem by using the algorithm that solves the set-cover problem.
- Lastly, we present extensive experimental results that show the effectiveness and efficiency of our method.

The rest of the paper is organized as follows. In Section 2, we discuss the related work. In Section 3, we propose REQUEST, and explain our processing method. In Section 4, we show experimental results, and we conclude our work in Section 5.

2 Related Work

Our region-based queries are similar to the spatial queries in sensor networks. These spatial queries in sensor networks have been actively reported [3,4,5,12]. For instance, in [12], they propose a distributed spatial index, called SPIX, over the sensor nodes to process spatial queries in a distributed manner. However, most of works about spatial queries in sensor networks have focused on using the predefined regions. In [12], spatial queries are used to answer questions such as "what is the average temperature in room-1?". In contrast, our region-based queries ask questions such as "which regions with a fixed size have the average

temperature lower than a certain threshold?". Thus, in REQUEST, regions are not predefined before the query is posed, but redefined whenever the size of region specified in the query is changed.

Our query processing scheme is also related to aggregate queries in sensor networks. Even if many works have been proposed to process aggregate queries, only a few works deal with grouped aggregate queries.

TAG [10] propose a grouped in-network aggregation method. In this method, as climbing up the global routing tree from the leaf nodes to the basestaiton, partial aggregated values for each group are computed and forwarded respectively at each node. However, this method has a problem if there are many groups to be maintained at each node. There have been other works [11,13] to improve the grouped in-network aggregation method, which are based on TAG. In these works, they focus on modifying the routing protocol, in order to reduce the size of messages. In [11], they propose group-aware network configuration algorithms. The key idea of these algorithms is selecting a node in the same group as a parent. By doing so, the number of partial aggregations that should be maintained at each node can be reduced. In [13], a multipath routing protocol is proposed in order for each node to send its data to different parents. These grouped in-network aggregation methods based on TAG consider only disjoint groups. However, in REQUEST, a large number of overlapped regions (groups) can be generated. The methods based on TAG are not directly applicable to our environment, since a node can belong to several groups simultaneously.

Zhuang et al. propose the max regional aggregate query [16] which is the most similar and related query with the proposed region-based query. The max regional aggregate query is for finding a region with the maximum aggregated value. To do that, they propose a sampling-based approach in which regions and nodes are sampled within a certain accuracy. However, they only consider regions at which individual nodes are centered. Therefore, some relevant regions to the user's interest can be missing. In addition, they only focus on reducing the data to process but do not deal with an efficient collection of the data.

Lee et al. in [9] present a framework to efficiently process group-by aggregate queries in sensor networks. They propose the compression scheme that uses the Haar wavelet in order to reduce the size of messages. However, they assume that groups and leader nodes are pre-determined, and only consider pre-clustered groups which are disjoint each other.

3 Region-Based Query Processing

In this section, we propose REQUEST, region-based query processing in sensor networks. The goal of REQUEST is gathering regional aggregated values energy-efficiently so that we can find some interest regions satisfying several conditions. The specification of the region-based query is as follows:

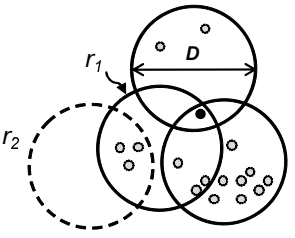


Fig. 2. 3 different regions in REQUEST

select	{region, aggfunc(attributes)}
from	sensors
group by	region(diameter D)
having	{having predicates}
sampling rate	{time of sampling interval}
duration	{maximum of sampling time}

We regard a region as a circle located in any places in the sensor network monitoring field. For the simplicity, we assume that the diameter of the circle is not larger than the communication range. We leave the problem which deals with regions with a larger size than the communication range for the future work. In fact, there are infinite number of regions in the monitoring field, even if every region has the same size. However, the number of regions can be limited since the number of sensor nodes is finite. In order to limit the number of regions reasonably, we formally define a region as follows:

Definition 1. *Let D be a diameter specified in the query. A region r is a maximal set of sensor nodes which are located within a circle having a diameter D. Since r is maximal, it is not contained in any other regions.*

Fig 2 shows that 3 different regions with the size D in REQUEST. We discard region r_2 since it is included in region r_1. Unlike existing grouped aggregate methods, regions can overlap in our approach. Through this, we can cover all areas which contain the sensor nodes deployed uniformly in the monitoring field. A node can belong to several different regions.

When there are a large number of groups or especially the selectivity of the having predicates in the query is relatively low, the earlier aggregation can have more benefits in terms of energy consumption. Therefore, we adapt a hierarchical aggregation method to process region-based queries more energy-efficiently. In this method, there is a leader node for each group, and non-leader nodes in the same group forward their sensing values, and finally aggregation of each group is computed in the leader node. Since regions overlap in our approach, a leader node can represent more than one region. In Fig 2, a dark-colored node is a leader node which covers 3 different regions.

The overall process of REQUEST comprises the following steps:

1. Regions and leader nodes are decided at the basestation.
2. The initial query message with the leader notification is sent to the entire network.
3. A query-initiated routing tree is constructed.
4. Every non-leader node sends its data to the leader node in the same region.
5. Leader nodes compute and forward the aggregation value for each region to the basestation.

3.1 Region Construction

The region construction in REQUEST is to find every possible disjoint combination of sensor nodes located within a circle corresponding to the query. For a naive idea, we can find regions by moving the circle from the top left corner to the bottom right corner. However, it is difficult to determine the appropriate step size for covering the entire region since a region can be placed at an arbitrary position. Moreover, if the area of monitoring field is large, this naive method is extremely time-consuming.

Therefore, we propose an efficient region construction method that utilizes the SEC(Smallest Enclosing Circle) index. Our intuitive idea is that every region can be identified by a SEC which encloses every node inside the region. Moreover, every SEC can be defined by using at most 3 points [14].

For example, Fig 3 shows the relationship between regions and the corresponding SECs. Fig 3(b) shows every possible SECs given a set of nodes. To construct regions with a diameter D is identical to find maximal sets of nodes inside a circle having a diameter D by Definition 1. In Fig 3(a), there are 2 regions, which are sets of nodes. To generate these sets, we find the largest SECs among the SECs smaller than a circle with a diameter D. In Fig 3(b), the SECs in the solid line correspond to the regions in Fig 3(a).

In summary, first we build the SEC index of the sensor network, and then construct regions by using the SEC index.

To build the SEC index, we need an algorithm of finding a SEC that completely contains a given set of points. Finding SEC problem has been well-studied in the research area of mathematics. We use an algorithm from [14], which is

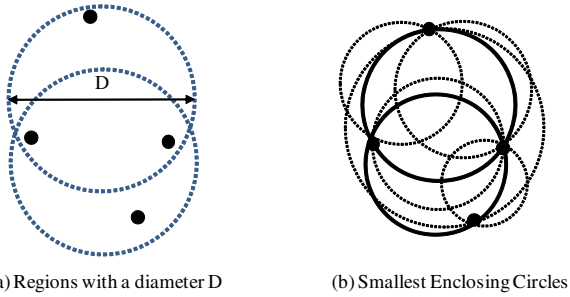


Fig. 3. Regions and the corresponding SECs

```
Algorithm BuildSecIndex
Input The set of all nodes N = {n_1, n_2, ..., n_m}
Output The SEC index I which is sorted by a diameter
begin
1.    for each node n_i in N
2.       for each node n_j in N
3.             for each node n_k in N
4.                   sec = FindSmallestEnclosingCircle(n_i, n_j, n_k)
5.                   nodeSet = The set of nodes which are contained in sec
6.                   Insert an index entry (sec.diameter, sec, nodeSet) to I
7.    return I
end
```

Fig. 4. Algorithm of Building SEC Index

simple to implement and has linear average time complexity. Fig 4 shows the algorithm of building the SEC index. Since every region can be defined by at most 3 nodes, the number of the SEC index entries is at most m^3, where m is the number of nodes. We manage the SEC index to be sorted by a diameter so as to construct regions efficiently in runtime. Each SEC index entry consists of a diameter, a SEC, and a set of nodes which are located inside SEC(Line 5). We assume that every node has a static position. Therefore, once the SEC index is built, we do not need further modification to the index.

By means of the SEC index, we can construct corresponding regions when the region-based query is posed. Fig 5 presents the algorithm of region construction. First, we find the largest SEC among the SEC index entries which have a smaller diameter than the diameter specified in the query(Line 1). From the largest SEC to the smallest SEC, we generate regions unless they are sub regions of already generated regions(Line 3∼5).

```
Algorithm ConstructRegions
Input The diameter of regions given in the query D, The SEC index I
Output The set of regions R = {r_1, r_2, ...}
begin
1.    MaxSEC = A largest SEC among SECs in I having a smaller diameter than D
2.    MinSEC = A smallest SEC, thus, a node itself with a diameter 0
3.    for each SEC sec from MaxSEC to MinSEC in I
4.       if sec is not redundant then construct a region r using sec
5.             Insert r to R
6.    return R
end
```

Fig. 5. Algorithm of Region Construction

3.2 Leader Selection

To minimize the communication of REQUEST, it is important to select optimal leaders of each region. There are several requirements for optimal leader selection as follows:

- Leader nodes should be close enough to the basestation.
- Leader nodes should represent as many regions as possible.
- Distances between leader and non-leader nodes should be short enough.
- Leader nodes should cover all regions and sensor nodes.

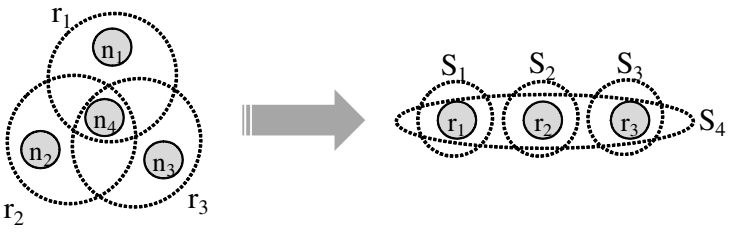


Fig. 6. Transformation from the leader selection problem to the set-cover problem

Based on these requirements, we formulate the leader selection problem as follows:

$$Minimize \quad \sum_{j \in L} (dist(root, j) \cdot |R_j| + \sum_{i \in N_j} dist(i, j))$$

$$Subject\ to: \quad \bigcup_{j \in L} R_j = \mathbb{R},\ where\ \mathbb{R}\ is\ the\ set\ of\ entire\ regions.$$

L : *The set of leader nodes.*
R_j : *The set of regions which include node j.*
N_j : *The set of nodes which are in the same region with node j.*
$dist(i, j)$: *The distance (hop counts) from node i to node j.*

In this formulation, we want to find the optimal L while minimizing the objective function at the same time. The objective function is the summation over j of the expected cost when we select a certain node j. Note that the size of messages between the basestation and the leader node j is $|R_j|$ times larger than that of messages between the leader node j and the non-leader nodes i. A unique constraint is for covering all regions and nodes.

To solve this problem, we adapt an idea that the facility location problem can be transformed into the weighted set-cover problem [8]. To transform the problem into the set-cover problem, it is required to define the set and the cost of the set. Intuitively, selecting a node as a leader in the leader selection problem is identical to choosing a set in the set-cover problem. Therefore, each node becomes a set and regions become elements of a set. Fig 6 shows that the sensor network in the left-side can be transformed into the instance of the set-cover problem in the right-side by means of our idea. If we select n_4 as a leader, we can cover 3 regions, which are r_1, r_2, and r_3. This is identical to choosing S_4 which can cover 3 elements in the set-cover problem.

The cost of each set is naturally derived from the objective function in our formulation.

$$C(S_j) = dist(root, j) \cdot |S_j| + \sum_{i \in N_j} dist(i, j)$$

```
Algorithm SelectLeaders
Input The set of all nodes N, The set of all regions R
Output The set of leader nodes L
begin
1.   U = R
2.   X = ∅
3.   for each node n_i in N
4.      S_i = {r | r is a region which includes node n_i}
5.      Insert S_i to X
6.      C(S_i) = dist(root, n_i)·|S_i|
7.      for each region r_j in S_i
8.           for each node n_k in r_j
9.                C(S_i) = C(S_i) + dist(n_i, n_k)
10.  L = GreedySetCover(U, X, C)
11.  return L
end
```

Fig. 7. Algorithm of leader selection

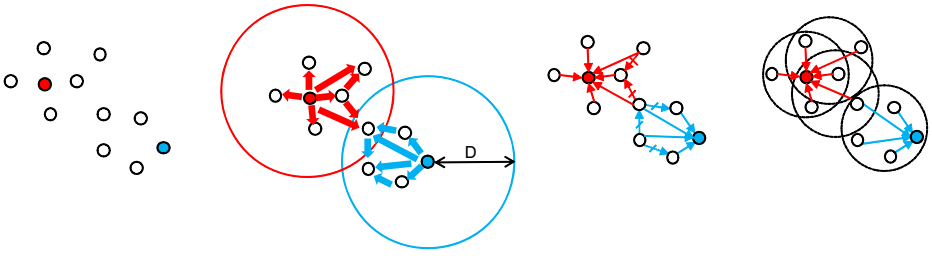
The set-cover problem is a well-known NP-complete problem, and has been actively studied in the algorithmic research fields. Among the several approximation algorithms that solve the set-cover problem in polynomial time, we use the set-greedy algorithm [2] which is the best known for the simplicity. In the set-greedy algorithm, we pick a set that covers the greatest number of elements not yet covered at each step. We skip the detail explanation of the set-greedy algorithm since it is beyond our work.

Fig 7 presents the algorithm of the leader selection, in which the set-greedy algorithm is called as a sub function.

3.3 Query-Initiated Routing Tree

In order for a leader node to communicate with non-leader nodes in the same region, it is required to build a routing tree for each leader node, called the query-initiated routing tree. When a new region-based query is posed, we construct a new routing tree for each leader node since the requested diameter can be changed. Basically, we apply to each leader node the routing method which is similar to the method used when the basestation constructs the global routing tree. The query-initiated routing tree construction performs the following steps:

1. Query messages with the leader nodes information are flooded in the entire network. (Fig. 8(a))
   - Each node is only aware whether itself is a leader or not.
2. The routing request messages ($leader\_id, hop\_count$) are broadcasted to the non-leader nodes within distance D from the leader nodes. (Fig. 8(b))
   - Receiver nodes should increase *hop_count* and broadcast to its neighbors.
   - We assume that every node can identify another node's location from the node id.
3. Each node designates the sender node as its parent node. (Fig. 8(c))
   - If multiple messages with the same leader node arrive at a node, the sender node of the message with the smallest *hop_count* is selected as the parent of the node.



(a) Flooding the query message with leader information

(b) Flooding routing request messages

(c) Parent selection

(d) Local region construction

Fig. 8. The process of query-initiated routing tree construction

4. For each leader node, the local region construction is performed. (Fig. 8(d))
   - When the leader nodes received data message from the other nodes for the first time, they perform the local region construction based on the sender nodes of arrived messages.

Fig 8 shows these steps sequentially. Local region construction algorithm is almost same as the algorithm in Fig 5 except that we can only consider nodes inside the regions that the leader node belongs to and SECs are dynamically generated at each leader node.

4 Experiments

In order to investigate the effectiveness and efficiency of the proposed method in REQUEST, we conduct experimental evaluations.

4.1 Experimental Environment

We implement our method and comparison methods using our own simulator.

As a topology for experimental evaluations, we deploy total 100 sensor nodes in a grid environment. The area of each grid segment is $100m^2$, and 2 sensor nodes are randomly located in each segment. We set the communication range to $10m$.

Since our experiments are not affected by the spatial or temporal correlation in sensor networks, we use the synthetic data that is generated randomly. In our simulator, sensing values for each node at each sampling time are randomly changed in the range from 0 to 10.

For the convenience, we assume that a packet has a simple header information which comprises of a source address and destination addresses. Note that destination addresses can be more than one, if a node belongs to several regions simultaneously. A message consists of the region or node identifier and the corresponding sensing value(s). We set the node identifier and the region identifier to a sequential number and coordinates of the region center, respectively. A region

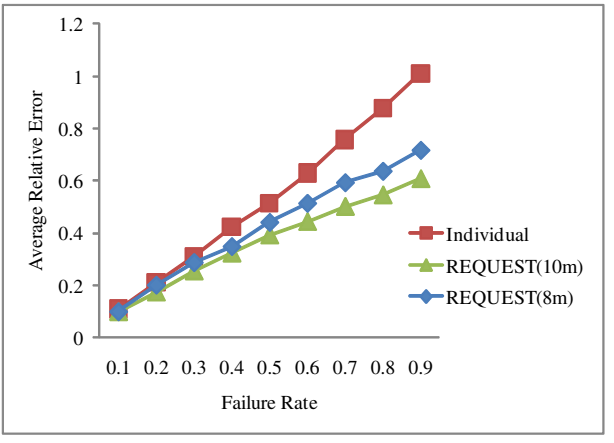


Fig. 9. Reliability

center can be decided from the corresponding SEC index entry. However, in our proposed method, we do not need the region identifier for a message, since the basestation can identify regions from the leader node id.

4.2 Reliability

We conduct experiments to evaluate the effectiveness of REQUEST. As a metric of the reliability, we use the average relative error rate. This metric is calculated as follows:

$$Average(relative\ error) = \frac{\sum_{i=1...n} \frac{|v_i - v_i'|}{|v_i|}}{n}$$

In the above formula, n is the number of values, v is the original value, and v' is a value with noises due to the failure. Using the average relative error, we conduct experiments with various failure rates from 10% to 90%. We use the following region-based query for this experiment.

select region, AVG(temp) **from** sensors
group by region(D)
sampling rate 1 **duration** 100.

Fig 9 shows the results of experiments on the reliability. "Individual", "REQUEST(8m)", and "REQUEST(10m)" are the cases that D is 0m, 8m, and 10m, respectively. As the failure rate increases, the average relative error rates of all the cases also increase. However, REQUEST(8) and REQUEST(10) show better accuracy than Individual especially when the failure rate is high. This result shows that using the aggregated values in the regions is effective to reduce the effect of the node failures.

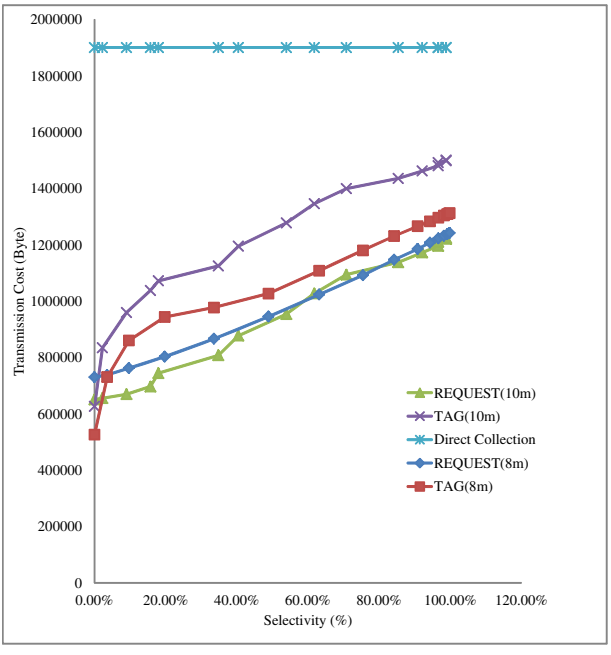


Fig. 10. Energy-efficiency

4.3 Energy-Efficiency

In sensor networks, consumed energy is the primary performance measure. Since the communication is the dominant factor in consuming energy, we use the amount of transmission as an efficiency metric. For the comparison system, we implement the grouped in-network aggregation method which is proposed in TAG [10]. In fact, in REQUEST, a node can belong to several groups at the same time, and each node can not know those groups before the query is posed. Groups can change according to the size of regions that is specified in the query. Additional communication to notify the group information to each node is needed. However, for the simplicity, we assume that every node knows its corresponding groups in advance for the comparison system, TAG.

The query used in this experiment is as follows:

select region, SUM(temp) **from** sensors
group by region(D)
having $0 \leqq$ SUM(temp) $\leqq t$
sampling rate 1 **duration** 100

We conduct experiments with varying the selectivity of the having predicate. To do that, we change t in the range from the minimum of SUM(temp) to the maximum of SUM(temp). If t increases, the selectivity also increases. To apply effects of changing the selectivity to the comparison system equally, we

implement TAG to exploit suppressing messages in the intermediate node. Like the experiments on the reliability, we test on two region sizes which are 8m and 10m.

Fig 10 shows that the cost of REQUEST is smaller than those of TAG and direct data collection in most cases. In the case that D is 8m, the difference between REQUEST and TAG is not large, because the number of regions that each node belongs to is smaller than that in the case that D is 10m. When the selectivity is close to zero, the communication cost of TAG is smaller than that of REQUEST. This is because in REQUEST, even if every aggregated values are filtered by the having predicate, the communication inside regions is still required. However, except for these cases, REQUEST is always better than TAG and direct data collection with regards to energy consumption.

5 Conclusion

In this work, we proposed a new type of query in sensor networks, called the region-based query. This type of queries are helpful to overcome noises in the sensor data, and to provide a macro view of a monitoring area. By permitting overlapping regions, we could deal with every possible regions which are generated by sensor nodes. In order to construct numerous regions efficiently, we used the SEC index that is built in the preprocessing phase. Moreover, to efficiently process the region-based query, we used a hierarchical aggregation method and addressed an optimization problem for leader selection with a solution algorithm by mapping the problem to the set-cover problem. Also, we built a new routing tree for each leader node, a query-initiated routing tree that enables intra-region communication. Finally, we showed that our proposed approach is effective and energy-efficient from the experimental results.

We plan to extend our work to deal with big size regions and aggregation of aggregation queries such as finding the average value among the maximum value of each region.

Acknowledgments. This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government (MEST) (No. 2010-0000863).

References

1. Precision agriculture, `http://en.wikipedia.org/wiki/Precision_agriculture`
2. Chvatal, V.: A greedy heuristic for the set-covering problem. Mathematics of Operations Research 4(3), 233–235 (1979)
3. Demirbas, M., Ferhatosmanoglu, H.: Peer-to-peer spatial queries in sensor networks. In: P2P 2003, pp. 32–39 (2003)
4. Dyo, V., Mascolo, C.: Adaptive distributed indexing for spatial queries in sensor networks. In: DEXA Workshops 2005, pp. 1103–1107 (2005)

5. Gupta, H., Zhou, Z., Das, S.R., Gu, Q.: Connected sensor cover: self-organization of sensor networks for efficient query execution. IEEE/ACM Transactions on Networking 14(1), 55–67 (2006)
6. Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: Energy-efficient communication protocol for wireless microsensor networks. In: HICSS 2000, pp. 8020–8029 (2000)
7. Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: An application-specific protocol architecture for wireless microsensor networks. IEEE Transactions on Wireless Communications 1(4), 660–670 (2002)
8. Hochbaum, D.S.: Heuristics for the fixed cost median problem. Mathematical Programming 22(1), 148–162 (1982)
9. Lee, C.H., Chung, C.W., Chun, S.J.: Effective processing of continuous group-by aggregate queries in sensor networks. Journal of Systems and Software 83(12), 2627–2641 (2010)
10. Madden, S., Franklin, M.J., Hellerstein, J.M., Hong, W.: Tag: a tiny aggregation service for ad-hoc sensor networks. SIGOPS Oper. Syst. Rev. 36(SI), 131–146 (2002)
11. Sharaf, A., Beaver, J., Labrinidis, A., Chrysanthis, K.: Balancing energy efficiency and quality of aggregate data in sensor networks. The VLDB Journal 13(4), 384–403 (2004)
12. Soheili, A., Kalogeraki, V., Gunopulos, D.: Spatial queries in sensor networks. In: GIS 2005, pp. 61–70 (2005)
13. Song, I., Roh, Y.J., Kim, M.H.: Content-based multipath routing for sensor networks. In: Kitagawa, H., Ishikawa, Y., Li, Q., Watanabe, C. (eds.) DASFAA 2010. LNCS, vol. 5981, pp. 520–534. Springer, Heidelberg (2010)
14. Welzl, E.: Smallest enclosing disks (balls and ellipsoids). In: Maurer, H.A. (ed.) New Results and New Trends in Computer Science. LNCS, vol. 555, pp. 359–370. Springer, Heidelberg (1991)
15. Younis, O., Fahmy, S.: Distributed clustering in ad-hoc sensor networks: A hybrid, energy-efficient approach. In: INFOCOM 2004, pp. 629–640 (2004)
16. Zhuang, Y., Chen, L.: Max regional aggregate over sensor networks. In: ICDE 2009, pp. 1295–1298 (2009)