SAQA: Spatial and Attribute based Query Aggregation in Wireless Sensor Networks

Yang Jie[1], Yan Bo[2], Sungyoung Lee[1], Jinsung Cho[1]

[1] Department of Computer Engineering
Kyung Hee University, Korea
[2] Intelligent Engineering Lab
Institute of Software, Chinese Academy of Sciences, China
[1] {yangjie, sylee}@oslab.khu.ac.kr, chojs@khu.ac.kr
[2] yan_bo04@mails.gucas.ac.cn

Abstract. In most wireless sensor networks, applications submit their requests as queries and wireless sensor network transmits the requested data to the applications. However, most existing work in this area focuses on data aggregation, not much attention has been paid to query aggregation. For many applications, especially ones with high query rates, query aggregation is very important. In this paper, we design an effective query aggregation algorithm SAQA to reduce the number of duplicate/overlapping queries and save overall energy consumption in the wireless sensor networks. This new aggregation algorithm focuses on the duplicate/overlapping spatial and attribute information in the original queries submitted by the applications. Our performance evaluations show that by applying our query aggregation algorithm, the overall energy consumption can be significantly reduced and the sensor network lifetime can be prolonged correspondingly. [1]

1. Introduction

Wireless sensor networks consist of large numbers of devices, each capable of some limited computation, communication and sensing, operating in an unattended mode. One unifying view is to treat them as distributed databases. The applications and these distributed databases will communicate through a set of queries, which is quite similar to the concept of SQL queries in the traditional database context.

Traditionally, applications forward queries to the base station which processes the queries one by one and sends queries to proper regions of the sensor network using the underlying routing infrastructure. However, query rate can be high due to a large number of applications sending queries. These applications heavily rely on the query search so that the energy consumption spent on sending and routing queries may far exceed that due to sending the response data. In these cases, optimizing query dissemination is critical to improve the overall performance of the sensor networks.

[1] Dr. Sungyoung Lee is the corresponding author.

AODV [3] (used in Cougar [9]) is a reactive routing protocol for ad-hoc mobile networks. It builds a route between two nodes only on the demand of the source node. Directed Diffusion [4] is a data-centric communication paradigm that integrates application-specific semantics into the routing layer. Data is named as attribute-value pairs and is disseminated from source nodes to the node of a query along multiple paths for reliability. In comparison, AQUIRE [7] uses random walks which adopt a look-ahead mechanism in each step to answer one-shot, non-aggregate, complex, replicate data queries.

In TAG [8] in TinyDB approach, an aggregate is computed bottom-up in the routing tree and many optimization techniques are used to improve the performance, e.g., snooping over the shared radio channel to reduce message load and improve accuracy of aggregates, hypothesis testing and adaptive network topology maintenance. Jonathan Beaver, et al. TiNA [1] provides further optimizations over TAG. It exploits temporal coherency tolerances. The approach is to send the data only when there is a significant change in the data value. A data value can be ignored if the variation from the previous value is within the range specified by the *tct* condition.

In [2], our work has some similarities to techniques proposed. The authors proposed a multi-layered overlay-based framework consisting of a query manager and access points (nodes), where the former provides the query aggregation plan and the latter executes the plan. The main goal is to minimize the number of queries sent out, to dispatch the aggregated queries to proper regions and to prevent data transmission in the same region happening multiple times. To achieve this goal, they present an effective query aggregation algorithm, which is mainly based on reducing the duplicate/overlapping spatial information of the original queries sent by the applications.

However, there is still some redundancy in the aggregated queries. We can easily find that not only the spatial information can be duplicated/overlapped, but also the attribute information. Thus, we propose our query aggregation algorithm SAQA based on the spatial and attribute information to help consolidate the queries and reduce the overall energy consumption for query dissemination and data transmission.

The remainder of this paper is organized as follows. Section 2 introduces the query model we use in our aggregation mechanism. In Section 3, we formalize the query aggregation problem and propose our algorithm SAQA for query aggregation. In Section 4, performance evaluation and analysis results are given. Finally we conclude our study with scope for future work in Section 5.

2. Query model

In this section, we give the query models used to conduct query aggregation.

First we make the following assumptions about the network: 1) a location-based routing scheme is supported by the sensor network; 2) we assume that there is a centralized base station that connects to applications; 4) we assume that that multiple applications can simultaneously send a number of queries to the sensor network.

Applications request information from a sensor network through queries. Depending on the nature of the application, many types of queries can be delivered on the sensor network. In general, these queries can be summarized by the following tuple [2]:

Q = < S, V, T, F, D >, where

S = Spatial information, indicating the geographical locations that the application is interested in.

V = Attribute information, indicating the list of attributes which the application is interested in.

T = Temporal information, indicating the duration of the query.

F = Frequency information, indicating the frequency at which the data should be reported.

D = Deadline restriction information, indicating the urgency at which the data should be reported.

In the example: "*Report regularly temperature level and wind speed from region S_1, S_2 and S_3 from time T_1 to T_2 every second*", where $S = \{ S_1, S_2 \}$, *V* = { temperature level, wind speed }, *T* = {T_1 to T_2}, *F* = 1 second, and *D* = {not urgent}.

In order to conduct query aggregation, we make the following assumptions about the query model: *1) Query content*: each query can ask for one or several spatial information (*S*) and attributes (*V*). The list of attributes and geographical locations will be referred to as $\{V_1, V_2 \dots V_n\}$ and $\{S_1, S_2 \dots S_n\}$ respectively in this paper. We assume that most of the queries have spatial information. A query without spatial information can be processed through the traditional query processing techniques such as flooding or direct diffusion [4], etc. *2) Query arrival rate*: we assume that queries are coming at a relatively high rate, or in other words, the deadline restriction level (*D*) of queries is not high so that we can temporarily buffer queries for aggregation. *3) Query temporal information*: we assume that the majority of queries are snap-shot queries, i.e. queries that ask for current value of the sensors as opposed to continuous queries, which asks for sensor values during a period of time.

3. Query aggregation design

A. Problem Definition

We first identify problems with the current query dissemination schemes.

Generally, when receives queries from applications, the base station directly forward them to the sensor network. The transmission of these queries may naively be flooding or follow some logic that the intermediate sensor nodes apply [4] [5]. When the queries are routed to proper sensors, the sensors start sending data back to the base station, which will deliver data to the applications accordingly. When there are multiple queries from applications, this process repeats until all the queries have been satisfied. At base station or some intermediate nodes, some caching algorithms may be performed to avoid redundant query forwarding.

Fig. 1 shows some of the problems of the above scheme, where queries can be location specific and contain multiple attributes. When two queries Q_1 *and* Q_2 come simultaneously and the queried information is not available in the base station local database, both queries will be transmitted to the network. In this case, the base station just needs to send Q_1. The information of Q_2 can be inferred from Q_1. For Q_1 and Q_3, they ask for partially different attributes in overlapped query areas, recombining into three simple queries as < *(S_2, S_3) (V_1, V_2, V_3)* >, < *(S_1) (V_1, V_2)* > and < *(S_4,) (V_2, V_3)* > will be beneficial. It reduces the energy overhead of sending separate duplicate query messages to the same region and more importantly avoids much disturbing other intermediate nodes (and node in overlapped region) in the location-based routing process. When sending the queries to their corresponding regions, the combined query will be routed to a proper node (such as the cluster head) in one of these regions only once. After that, this node will separate the combined query. From here, the query part corresponding to the other region will be routed to its destination and different attributes can be collected at the same time in the same region to satisfy the original queries. Compared to the original case, the number of intermediate nodes (and nodes in the overlapping region) involved in the routing process will therefore be reduced. As the last example, when Q_1 and Q_4 ask for different attributes on the same region. Instead of sending two different queries, we can combine them into two queries as < *(S_1, S_2) (V_1, V_2, V_3)* > and < *(S_3) (V_1, V_2)* >.

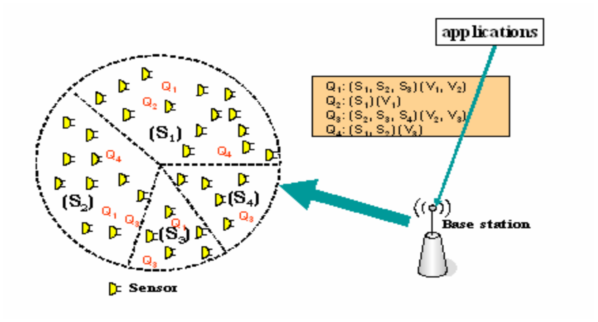


Fig. 1. Query example

As illustrated by this example, our motivation in this paper is to 1) perform the query aggregation efficiently and dispatch the aggregated queries to proper regions so that the routing process will disturb a minimal number of intermediate nodes, and 2) prevent data transmission of sensor nodes in the same region from happening multiple times, 3) collect information for different attributes at the same time to satisfy different queries when they are querying the same area. By achieving these objectives, we can reduce the overall energy overhead for both query transmission and data delivery. Thus, the lifetime of the sensor network can be prolonged.

There are *N* queries: *Q_1* … *Q_N* denoted by set *Q*. For the query aggregation operation, we mainly use two important concepts, union and intersection, in set theory. In the next section, we study the algorithms in detail.

B. SAQA

The following examples show us how our algorithm works.

Suppose we have one query Q = < (S_1, S_2, S_3), (V_1, V_2) >. According to the set theory, there are only three relationships between two sets: Disjoint, Intersectant and Inclusive.

We will give the following four queries as examples to do the aggregation, which uses the above set operations.

Q_1 = < (S_1, S_2), (V_1, V_2, V_3) >;
Q_2 = < (S_4, S_5), (V_1, V_2) >;
Q_3 = < (S_2, S_3, S_5), (V_2, V_3) >;
Q_4 = < (S_2, S_3), (V_2) >.

For Q_1, the aggregated query consists of two sub queries:

< (S_1, S_2), (V_1, V_2, V_3) … >; < (S_3), (V_1, V_2) … >

Since (S_1, S_2) is included in (S_1, S_2, S_3), we can separate the area to two parts, one is the intersection part, the other is the set of elements outside the intersected area. Also, (V_1, V_2) is included in (V_1, V_2, V_3), we integrate them to only one set. After aggregation, two queries will be sent to two areas in wireless sensor network. One are is (S_1, S_2), and the other one is S_3. Nodes in these two areas will be disturbed only once, but still gather the necessary information.

For Q_2, no aggregation is needed in that the spatial information in Q and Q_2 is disjoint information.

For Q_3, the aggregated query consists of three sub queries:

< (S_2, S_3), (V_1, V_2, V_3) … >; < (S_1), (V_1, V_2) … >; < (S_5), (V_2, V_3) … >

As for the spatial information, Q and Q_3 have one intersection (S_2, S_3). That's why we aggregate the two queries into three parts, one is the intersection, and the other two are the rest part of Q and Q_3. Then for the attribute information, in the case of (S_2, S_3), we unify the two parts from Q and Q_3 to get one set (V_1, V_2, V_3). That is, what we have introduced, the union operation. And for the rest two areas S_1 and S_5, the attribute in the new aggregated queries will be the same as in the original queries.

For Q_4, the aggregated query consists of two sub queries:

< (S_1, S_2, S_3), (V_1, V_2) … >

(S_2, S_3) is included in (S_1, S_2, S_3) and also (V_2) is included in (V_1, V_2), Q_4 can be fully aggregated into Q. So there is only one query after aggregation.

The above examples list all the possibilities of query aggregating operations. The key issue here is to efficiently find the good query merge order. In fact, as the base station has the information of all query information, it can globally calculate all overlapping regions in the whole query space and find the zones with the heaviest overlapping (calculated by as weight). Based on the overlapping weight, the queries located near to the heaviest overlapping zone will be merged together with higher priority. Thus, efficient query merge order can be easily achieved with reasonable performance.

The detail of SAQA is given in figure 2 which also shows the flowchart of SAQA:

Parameters:

Q: set of input queries with cardinality $|Q|$, each query $Q_i \in Q$ is denoted by $<S_i$, $V_i>$, where S_i represents the query region and *Vi* represents the query attributes

Y: set of output aggregated queries by SAQA algorithm

4. Performance evaluation

In this section, we use simulation to evaluate the performance of the system that uses our algorithm. We will first describe the experimental model and then report performance results.

Spatial and Attribute based Query Aggregation algorithm (SAQA)

```
Begin:
    //Process the input queries in set Q by filtering queries with full cover property
    For queries $Q_i$, $Q_j \in Q$, where $i \neq j$,
            If queries are overlapping, i.e., $S_i \cap S_j \neq \emptyset$
      //Calculate the overlapping zone and assign the weight for each zone
      //Sort the weights and assign the query merge order
Do aggregation:
    For queries $Q_a$, $Q_b \in Q$, where $a \neq b$, //$Q_a$ and $Q_b$ have the largest weight of overlapping zone
            If    $S_a \subset S_b$, the same with $S_b \subset S_a$
                    $S_{ab} = S_a \cap S_b$
                    $V_{ab} = V_a \cup V_b$
                    $S_{rb} = S_b - S_{ab}$    //$S_{rb}$ is the rest part of $Q_b$ after aggregation
                    $V_{rb} = V_b$
                    $Y = Q_{ab} + Q_{rb}$
            Else
                  if    $S_a = S_b$
                          $S_{ab} = S_a$
                          $V_{ab} = Va \cup V_b$
                          $Y = Q_{ab}$
                  Else
                    $S_{ab} = S_a \cap S_b$
                    $V_{ab} = V_a \cup V_b$
                    $S_{rb} = S_b - S_{ab}$    //$S_{rb}$ is the rest part of $Q_b$ after aggregation
                    $V_{rb} = V_b$
                   $S_{ra} = S_a - S_{ab}$    //$S_{ra}$ is the rest part of $Q_a$ after aggregation
                    $V_{rb} = V_a$
                    $Y = Q_{ab} + Q_{rb} + Q_{ra}$
```

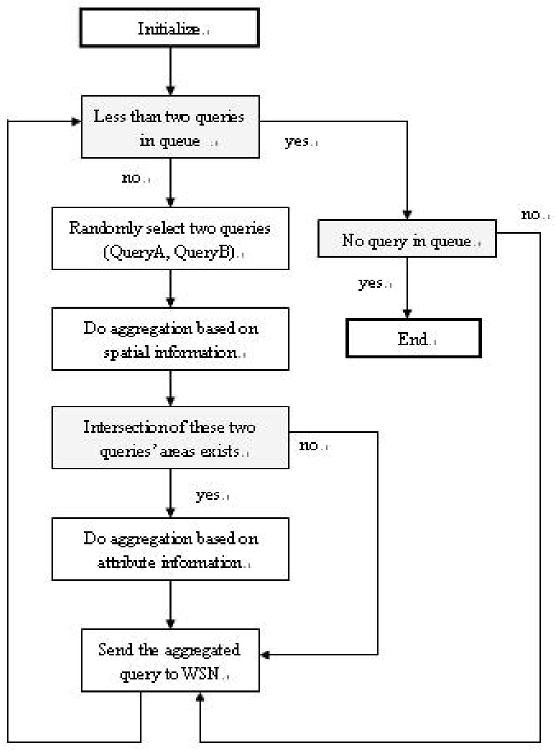

Initialize
Less than two queries in queue
yes
no
Randomly select two queries (QueryA, QueryB)
No query in queue
no
yes
End
Do aggregation based on spatial information
Intersection of these two queries' areas exists
no
yes
Do aggregation based on attribute information
Send the aggregated query to WSN

Fig. 2. Flowchart of SAQA

A. Experimental Model

1) Network and Energy Model

We assume that there are *N* queries, each of which is *m*-bit long. The queries uniformly request data from the whole network. We assume that each sensor works in free space mode with some experimental data introduced in [6]: the energy consumption of sending message is calculated by $E_{tx}(a, b) = E * a + E * a * b^2$ and the energy consumption of receiving a message is calculated by $E_{rx}(a, b) = E_{elet} * a$, where *a* is the message size and *b* is the message transmission distance between the sender and receiver, *E*= 50 *nJ*/bit, and *E* = 100PJ/bit*m^2 (1 *nJ* = 1000 *pJ* and 1 *MnJ* = 1000 *nJ*). Since the energy consumed for processing queries and sensing data consists of only a very small portion of the overall energy consumption (node that energy consumed to process 100 million instructions almost equals that to transfer 10 bits of data), we do not take it into account in our calculation.

2) Evaluation System

We define the *Query region overlapping degree R* as the ratio of overlapping region size and the original query region size. For example, two queries Q_1 (query region size S_1) and Q_2 (query region size S_2) have the overlapping region with size S_{12}. In this case, $R = S_{12} / (S_1 + S_2)$. The range of *R* is [0, 0.5], where 0 represents the case where the query regions do not overlap and 0.5 represents the case where the query regions are exactly the same. We also define the *Query attribute overlapping degree T* as the ratio of overlapping attribute number and the original query attribute number. The range of *T* is also [0, 0.5], where 0 represents the case where the query attribute do not overlap and 0.5 represents the case where the query attributes are exactly the same.

We compare the following query processing approaches with our SAQA:

1) Pure query processing (PQP): In this approach, the base station just simply forwards queries to the sensor network without any aggregation. Obviously, this approach is not optimal because intermediate sensor nodes do not have a global view of the whole network. Sensor nodes in an overlapping region may have to send the same data multiple times to reply for different queries asking for attribute in the same region.

2) Spatial based query aggregation (SQA): In this approach, the base station acts as a manager in a normal centralized system, it makes the query aggregation decision based on the spatial information of all the input queries. As a result, a number of queries which are not sharing any common regions are generated. The queries are sent to corresponding regions and executed locally. Data from sensors will be sent back to the base station by each region. No query or data forwarding between regions is implemented. The main advantage of this scheme is its simplicity and ease in implementation. However, it can generate a larger number of new queries and disturb more sensor nodes in both query dissemination and data transmission process. Besides, the queries can be aggregated more based on the attribute information in the original queries, so that the number and size of the aggregated queries can be reduced.

B. Performance Results

We report the performance results comparing the following metrics. The conclusions we draw generally hold for many other cases which we have evaluated.

1) The Sensitivity of Query Number

Fig. 3 shows the data on the sensitivity of energy performance for different input query numbers. In this Figure, *X* axis represents the total number of input queries and Y axis represents the total energy consumption. From this figure, we have the following observations: i) overall, our SAQA outperforms both PQP and SQA algorithms. For example, with a large number of queries, i.e., from 100 to 200, the SAQA can achieve around 80%-350% performance improvement over the PQP and SQA. The result matches our expectation because as SAQA adequately consolidates the queries. The energy cost for both query transmission and data delivery has been significantly reduced. ii) SQA performs better than PQP. The reason is because SQA conducts the query aggregation at the centralized base station. It can reduce the energy consumption by removing the redundant queries for the overlapping regions. iii) The overall energy consumption is sensitive to the number of queries. A larger N generally implies more queries, and therefore, more energy consumption. However, in SAQA, this problem is alleviated because, when the number of queries increases, there will be more chances of overlaps between query regions, which can be effectively reduced by our query aggregation approach.

2) The Sensitivity of Query Region Size

Fig. 4 shows the data on the sensitivity of energy performance for different query region sizes. In this figure, the X axis represents the different query region sizes and Y axis represents the total energy consumption. As the query size is enlarged, the overall energy consumption also increases. This is because a larger region size means that more sensor nodes are involved, or more query/data transmissions are performed. Regardless of the query size, our SAQA performs better than the other 2 schemes with the same reasons given above.

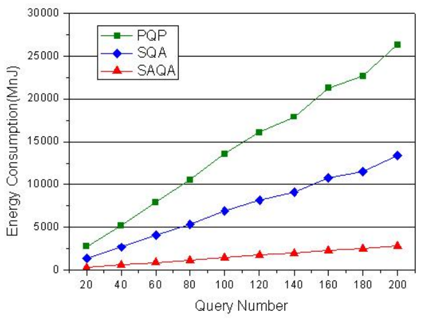


Fig. 3. Energy sensitivity of total query

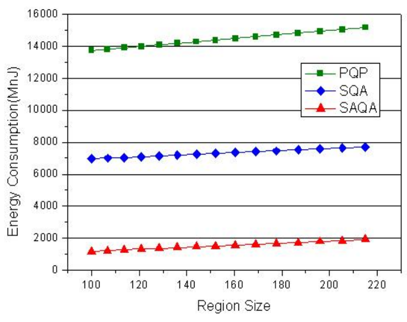


Fig. 4. Energy sensitivity of region size

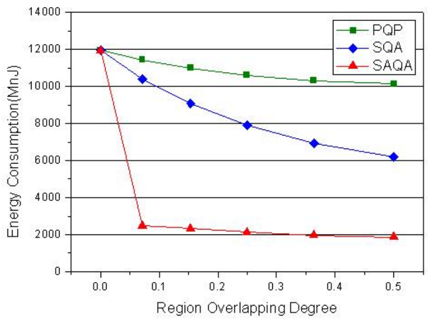
3) The Sensitivity of Query Region Overlapping Degree

Fig. 5 shows the data on the sensitivity of energy performance for different query region overlapping degrees. In this figure, the X axis represents the different query region overlapping degrees and Y axis represents the total energy consumption. When the query regions are not overlapping, SAQA can not take advantage of query aggregation. Therefore, performance of SAQA is almost the same as other algorithms.

However, as query regions become highly overlapped, SAQA shows significant improvement. As shown in this figure, when the overlapping degree becomes 0.1, overall energy consumption of SAQA reduced dramatically. The behavior of SQA is explainable because, although SQA can prevent the duplicate queries for overlapping regions, duplicated attribute information not compressed require extra energy cost for the query/data transmission.

4) The Sensitivity of Query Attribute Overlapping Degree

Fig. 6 shows the data on the sensitivity of energy performance for different query attribute overlapping degrees. In this figure, the X axis represents the different query attribute overlapping degrees and Y axis represents the total energy consumption. We can easily find that besides the spatial information, the attribute information can also be aggregated to reduce both the number and size of queries, which will consequently reduce the energy consumption of query/data transmission.



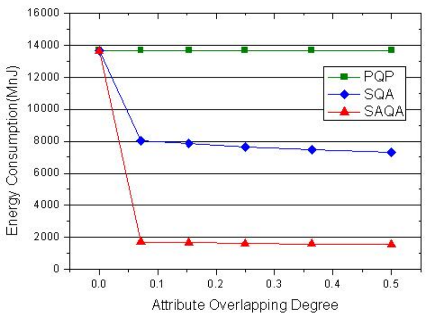


Fig. 5. Energy sensitivity of region overlapping degree

Fig. 6. Energy sensitivity of attribute overlapping degree

5. Conclusion and future work

In this paper, we propose an effective query aggregation algorithm SAQA to reduce overall energy overhead for the data services in the sensor network. To the best of our knowledge, this is the first study that leverages existing research work and address the issues in this aspect. We conduct extensive performance evaluations on different algorithms. Our evaluation results show that by applying our query aggregation algorithm, we can significantly reduce the amount of query traffic and energy consumption for data services.

There are several directions to extend our study. First, in the original model, we implicitly assume that the underlying architecture supports location-based routing. Extending our algorithm so that it can support other routing protocols would be one direction. Second, in the query aggregation algorithm, we construct our zones based on the input query zones and do not consider the existing topology and distribution of sensors in the network. Such as, adjacent spatial information can be combined together in that they may share part of the route for query dissemination. Combining both dimensions (input query zones and network topology) in our algorithm will certainly produce better results. Finally, find an efficient way to decide the query merge order is also an important issue we should consider about.

Acknowledgement

This work was supported by IITA Professorship for Visiting Faculty Positions in Korea from Ministry of Information and Communications.

REFERENCES

1. Jonathan Beaver, Mohamed A. Sharaf, Alexandros Labrinidis, Panos K. Chrysanthis.: Power-Aware In-Network Query Processing for Sensor Data, MobiCom 2003
2. W. Yu, T. Le, Dong Xuan and W. Zhao.: Query Aggregation for Providing Efficient Data Services in Sensor Networks, in Proc. of IEEE Mobile Sensor and Ad-hoc and Sensor Systems (MASS), October 2004
3. Ian D. Chakeres and Elizabeth M. Belding-Royer.: AODV Routing Protocol Implementation Design. WWAN, 2004
4. C. Intanagonwisat, R. Govindan, and D. Estrin.: Directed Diffusion: A Scalable and Robust Communication, In Proceedings of ACM MobileCom'00, August 2000
5. X. Li, Y. J. Kim, R. Govindan, and W. Hong.: Multi-dimensional Range Queries in Sensor Network, In Proceedings of ACM SenSys'03, Nov., 2003
6. H. O. Tan, and I. Korpeoglu.: Power Efficient Data Gathering and Aggregation in Wireless Senor Network, In Proceedings of ACM SIGMOD'03, Special Section on Sensor Network Technology and Sensor Data Management, 2003
7. N.Sadagopan, B.Krishamachari, and A.Helmy.: Active Query Forwarding in Sensor Networks, Accepted to Journal of Ad-hoc Networks, ELSEVIER, August, 2003
8. Samuel Madden, Michael J. Franklin, Joseph M.Hellerstein, and Wei Hong.: TAG: a Tiny Aggregation Service for Ad-Hoc Sensor Networks. OSDI, 2002
9. Yong Yao and Johannes Gehrke.: Query Processing for Sensor Networks. CIDR, 2003