



The Research of Data Storage and Retrieval Scheme for Wireless Sensor Networks

Yan Liu^{1*}, Weiliang Tao¹, Kai Liu^{2*}, Ying Xu²

¹*School of Electronic Information, Wuhan University, P.R.China*

²*State Grid Electric Power Research Institute, Wuhan, P.R.China*

* Corresponding author's Email: llyytt@gmail.com

Abstract: Recently, several data storage schemes have been developed to store massive sensor data in the wireless sensor networks (WSNs). A crucial task of WSNs is to disseminate useful information to users efficiently and handle the data storage architecture incorporating their extreme resource constraints. A Distributed Index based Multi-resolution Storage Architecture (DIMSA) for data storage and retrieval at multiple levels of detail in WSNs is proposed in this paper. The proposed scheme could automatically disseminate useful information to users in correspondence to the user defined level of detail to save energy. In the condition that some attributes are included in many events, the values for these attributes have to be replicated and stored at different places in the network for each individual event in traditional data-centric storage schemes. This will consume some extra energy. To address this problem, DIMSA deploys an attribute allocation scheme for finding a good distribution of attributes to the sensor networks. In addition, a distributed index scheme is also proposed to avoid query flooding and extra data transmission. Analysis and simulations are conducted to evaluate the performance of the proposed scheme. The results show that DIMSA outperforms the traditional data-centric storage scheme like GHT.

Keywords: Wireless Sensor Networks; data-centric storage; storage architecture; energy consumption; data dissemination; multi-resolution.

1. Introduction

Wireless sensor networks (WSNs) offer new opportunities for pervasive monitoring of the environment and ultimately for studying previously unobservable phenomena [1]. In wireless sensor networks, an extremely large number of sensor nodes are densely deployed and these sensor nodes are capable of not only measuring real world phenomena, but also storing, processing, and transferring the sensing data. Large amount of sensing data generated in WSNs are distributed across whole network and it is not always possible to immediately transmit them as they are generated. The sensing

data need to be stored temporarily within the network and used in response to dynamic queries. Thus, it is a critical problem to deal with the data storage in the sensor network. In contrast to conventional network data storage, storing data in WSNs is challenging because of the limited power, memory, computation capability, and communication bandwidth of WSNs.

Wireless sensor networks place several requirements on distributed storage architecture. One traditional storage management approach is storing the sensing data to a base-station (BS) located outside the wireless sensor network [2]. It incurs significant energy expenditure to send every sensor's reading to an external site, which severely decreases the dep-

loyment lifetime. Another common approach is to store the sensing data locally [2]. In this way, the storage resource might be exhausted quickly because of the large amount of data. However, the observer may not be interested in the sensing data of a single node but an average value of a group of sensors in a small area. Therefore, there need some strategies to aggregate sensing data and to store only aggregation results in wireless sensor network, which can store more data for longer time. In addition, it can distribute rich data to every node in the sensor network.

This paper proposed a Distributed Index based Multi-resolution Storage Architecture (DIMSA) for WSNs. The proposed scheme is designed for minimizing the communication cost involved in data storage and query in a large-scale data-intensive sensor network.

The contributions of this paper are listed as three aspects. First, in data-intensive applications, users are interested in different levels of detail data. The proposed scheme could automatically retrieve subsets of the data in correspondence to the detail requirement. Thus, users are able to retrieve data of various levels of detail from data source.

Second, the early proposed multi-resolution storage is based on spatial-temporal summarization [1] [3] or multiple temporal resolutions [4]. The data reported and stored in these schemes are at level of events. If an attribute is included in many events, then the values for that attribute have to be replicated and stored at different places in the network for each individual event [5]. In our proposed DIMSA, the data storage is at attribute value resolutions, the low level data allocation and aggregation are based on attributes. Thus the attribute need not be replicated at multiple nodes and hence saves communication cost involved in storing, replicating data, and retrieving data.

Third, users may issue different type of queries, such as “what’s the temperature in zone A at 4 am?”, “list all the areas whose temperature lie between 40 °C to 50 °C?” etc. Multi-resolution data attributes summaries could serve the queries of areas that match the definite attributes well. But it is inefficient for answering all the attributes that happened in an area for it will flood the query to all the storage nodes to find out the required attributes. To address this problem, the distributed index nodes were introduced to accomplish further query efficiency.

The rest of the paper is organized as follows. In section 2, related existing work is reviewed. Section 3 presents the preliminaries. In Section 4 and Section 5, the proposed DIMSA and the query

processing are described. Section 6 is the performance evaluation. In Section 7, the future study problems are discussed. Section 8 is the conclusion.

2. Related work

At present, many data storage methods have been proposed for sensor networks. These schemes can be divided into three categories: Local Storage (LS), External Storage (ES), and Data-Centric Storage (DCS) [6]. Studies about comparison of the above three kinds of methods in the overhead of storage, query, and update have been conducted in [6] [7].

In LS mechanism, each sensor node generates and stores its own readings locally [8] [9], therefore the update overhead is negligible. However, LS schemes are not efficient in that sinks do not know which sensor nodes store the interested data and have to collect the interested data typically use flooding over the whole network to send query packets.

In ES mechanism, each node detecting data needs to deliver its data to the external sink. Since all the sensing data are stored at the sink, the sink needs large enough memory space to store the data. Although there is no cost for sink query, it may waste much energy for transmitting the data in which the sink is not interested.

In DCS based scheme [6] [10], numbers of nodes selected from the WSNs are responsible to store the sensing data. DCS uses a Distributed Hash Table (DHT) [11] [12] which map the data of the same name to a fixed location in WSN. As an event is detected by a sensor node, the data is stored at the corresponding data-centric node and queries for that data are directly routed to the storage node without flooding the network. Therefore, it could conserve the energy for data storing and retrieving. DCS offers two primitives to send and receive the value of a corresponding data from a hashed node: (1) the *Put(dataName, dataValue)* primitive to store the value of the data corresponding to a detected event at the sensor node. The stored data is defined as *dataName* which serves as the *key* in the DHT and is typically based on the relevant event type. (2) the *Get(dataName)* primitive to retrieve the value of the data stored at the node corresponding to the given *dataName*.

In [6], Ratnasamy et al. have proposed a data-centric storage based on Geographic Hash Table (GHT) which is inspired by internet-scale Distributed Hash Table (DHT) systems such as Chord [13] and CAN [12]. GHT hashes user-defined

events into geographic coordinates and stores the event at the sensor node closest to the hashed location.

Multi-resolution data is important for energy efficiency for many sensor networks. Under some circumstance, providing data of less detail which could satisfy the users' need is energy efficiency. Several multi-resolution data summaries are proposed [1] [3] [4]. The scheme in [1] [3] is based on spatial-temporal summarization and [4] is based on multiple temporal resolutions. A Data-Centric Attribute Allocation and Retrieval (DCAAR) scheme is designed for minimizing the communication cost involved in data retrieval from large-scale sensor networks [5]. They all not collectively consider the problem of satisfying the users' queries in both level of detail and different query types. The proposed Distributed Index based Multi-resolution Storage Architecture (DIMSA) will address this problem and provide basic storage architecture for WSNs to achieve query efficiency with minimum energy consumption.

3. Preliminaries

In this paper, a large scale sensor network composed of n sensor nodes in a square is considered. To simplify the problem, we assume the following properties about the network:

- (1) the area covered by sensor networks is a two dimension area with a known approximation of the boundary and terrain effect on the detection is not taken into account;
- (2) nodes are densely and uniformly deployed;
- (3) nodes are static or with slow movements;
- (4) each node knows its own location and its neighboring nodes within its radio range d ;
- (5) Greedy Perimeter Stateless Routing (GPSR) [13] is used to support the delivery of packets to particular location. GPSR is a strongly geographic routing protocol that takes a location rather than an address to deliver packets. GPSR has two algorithms for routing [13]: a *greedy forwarding* algorithm that moves packets progressively closer to the destination at each hop, and a *perimeter forwarding* algorithm that forwards packets where greedy forwarding is impossible.

4. Storage architecture

As mentioned in Section 3, the sensor nodes are assumed to be uniformly deployed over a rectangular region. Thus, the entire network is divided into small virtual grids with equal size, as shown in Figure 1. This original dividing of the sensor

network is based on the highest resolution level. The size of the grid should be no more than the minimum size of the area that users are interested in. Besides this limit, the grid size should be small enough that the sensing ranges of all the nodes inside the grid could guarantee the coverage over the grid. Let the length of the side of a grid be a . Therefore, the value of a is selected as $a \leq \frac{d}{\sqrt{2}}$, where d is the radio range of sensor node.

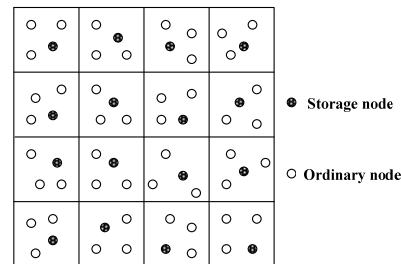


Figure.1 Dividing a sensor network into grids.

The proposed scheme is based on a three-resolution-level architecture shown in Figure 2. The number of resolution levels is determined by application. At every resolution level, each grid is assigned an identifier denoted as g_{ij} , where i denotes the resolution level and j denotes the grid number. Level 0 is the highest resolution level, level 1 is the lower resolution level and level 2 is the lowest resolution level. The less the resolution level is, the larger the grid size. The data of the highest resolution level is as detailed as the raw sensing data. While the data in lower resolution is processed with aggregation, data retrieval in these lower resolution levels is more energy efficient than the higher resolution level at the cost of losing more data fidelity.

At highest resolution level (level 0), inside each grid a storage node is elected to collect and store the sensing data from all the nodes in the grid. At lower resolution level (level 1 or level 2), there is not only a storage node but also an index node in each grid. Index node preserves the index information of the data generated from the area covered by the grid where this index node lies in. Index nodes are responsible to servicing the user queries. In order to make Figure 2 to be seen clearly, Figure 2 only shows the ordinary nodes in g_{01} while the ordinary nodes in other grids are omitted.

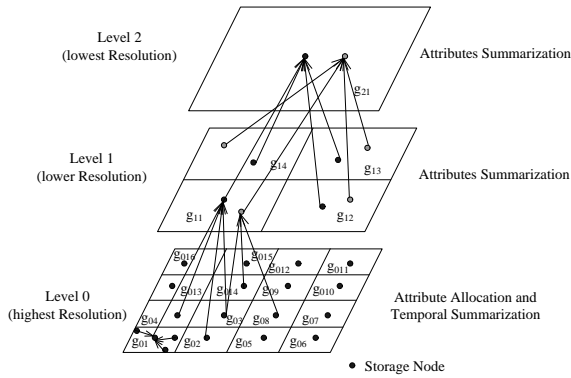


Figure.2 Data storage architecture.

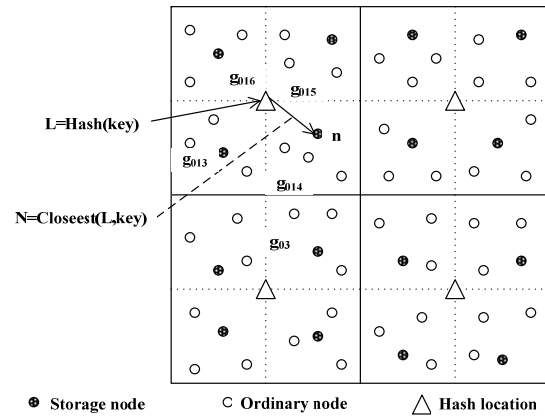


Figure.3 Attribute allocation.

4.1 Attribute allocation

In literature [4], the authors have proposed an attribute allocation methodology in which the query priorities are used to determine correlations between each pair of attributes and the correlations are used to determine the distribution of attributes. This approach differs from some existing data-centric approaches in that it attempts to distribute attributes instead of specific user-defined events. The proposed DIMSA scheme is similar to this attribute allocation methodology. If an attribute is included in many events, the attribute needs not be replicated at multiple places and hence saves communication cost involved in storing and retrieving data.

The process to compute and create correlation tree in literature [4] is too complex for resource constraint sensor network. If the attribute is stored near the node from where it is generated, it can reduce the communication cost in the network. Therefore, allocating attribute to the grid in which it is most probably to be sensed is a way to simplify the problem. However, if the sensing data in the network is skew, the storage node in the grid generating large amount of data will be exhausted quickly. The random hash function in GHT [5] could distribute the sensing data randomly to the whole network. The hash function is predefined and sensor readings may be hashed to any point in the network, making most of the readings stored far away from their generating locations. It leads to more energy cost to deliver the sensing data.

Under the consideration of energy efficiency, the hash function similar to GHT [5] is used in the attribute allocation, while it has a change to implement.

Here it presents how attribute allocation is implemented in the sensor network. Every attribute has a predefined name which serves as the *key* value in the hash function. The corresponding hash function looks like this: $L = Hash(key)$. Where L is the hash location represents a geographical coordinate which may not represent an actual sensor node. It should be noted that the hash location does not indicate a definite storage location but an area that contains several grids, which is different from the GHT scheme. As shown in Figure 3, the network has four hash locations and each hash location represents the area covered by four grids. Take the attribute A_i that most probably generated in grid g_{03} as an example. Its hash location L can be obtained by implementing the hash function and L will be any four hash location in the network. Suppose the calculated location L is the area covered by the grids of g_{013} , g_{014} , g_{015} and g_{016} , as shown in Figure 3. Then, the next step is using the function of $N = Closest(L, key)$ to calculate the storage node which will store A_i . The feature of $Closest(L, key)$ is to find out a definite storage node closest to the data source of the attribute represented by *key* in the area covered by L . Since g_{014} is closest to g_{03} , attribute A_i is assigned to storage node n in g_{014} in Figure 3. The probability of an attribute generated in a grid can be obtained from the statistic analysis from the historic data.

After the storage node of an attribute is calculated, the attribute is delivered from its generated node to the storage node if the delivery path is connected. If a path is not available from the originating node to storage node, the functions of

$L = Hash(key)$ and $N = Closest(L, key)$ are computed repeatedly until finding a valid storage node. The number of computation times is limited to a value t which is predefined. If it could not find a valid storage node after processing the allocation functions t times, the attribute will be stored in the storage node in its generating grid.

4.2 Multi-resolution data summaries

Since users may be interested in sensor data at different granularity in many data-intensive wireless sensor network applications, a fundamental design goal of DIMSA is the ability to extract sensing data in a multi-resolution manner from a sensor network. To this end, multi-resolution data summaries are deployed in our proposed scheme.

The original data summary, which is called as temporal summarization based on attribute, is done by the storage node in each grid at level 0. It should be noted that level 0 is the highest resolution level. As in Figure 2, the storage node in g_{01} collects sensing data from all the nodes in the grid. Then it aggregates the data into different attributes each of which is in sequence of its sensing time. After this original temporal summarization, each storage node computes the storage locations of its collected attributes and sends them to their corresponding storage nodes.

The phase of attribute summarization constructs a grid-based data compression. At each higher level, each grid covers a larger spatial scale, but the data stored in it are more compressed, and are therefore more lossy. Figure 2 illustrates this construction. At level 1, the region covered by g_{11} is composed of g_{01} , g_{02} , g_{03} and g_{04} , storage node in g_{11} collects attributes from these four grids, then, compresses them to a less detailed representation. Note that, due to the attributes allocation in level 0, the data source of g_{11} is not the data generated in its covered region but the data stored in the region. It is efficient for querying the data attributes. However, it has disadvantage of have to flood the query to all the storage nodes to find out the required attributes for queries of answering all the attributes that happened in a region. In section 4.3, a distributed index scheme will be introduced to solve this problem.

The data of the highest resolution level (level 0) is in the same detail level as the raw data but usually with more redundancy. Data retrieval in less resolution level is more energy efficient than the higher resolution level at the cost of losing more fidelity of the data.

4.3 Distributed index scheme

In this section, a Distributed Index Scheme is introduced to facilitate the queries of different types. At lower resolution level, an index node in each grid is elected to serve data queries. Index node preserves the index information of all the data generated in the region which it located grid covered. For example, as shown in Figure 2, if users want to know the attributes of the events happened in g_{11} , the query is sent to the index node in g_{11} . Then, it can be obtained from the index node in g_{11} that the data users interested in are stored in g_{03} , g_{08} and g_{014} . Next, the query could be directly routed to the storage nodes in g_{03} , g_{08} and g_{014} to acquire the most detailed data. However, if users only want to obtain the less detailed data that happened in g_{11} , they can acquire the storage locations of these data in level 1 from the index inform stored in g_{11} . Then the query is routed to the storage nodes in g_{11} , g_{12} and g_{014} to acquire the less detailed data. This Distributed Index Scheme can reduce communication between nodes, thus decrease energy consumption.

5. Query processing

Queries based on DIMSA can dramatically reduce the cost of search. Queries injected into the network from sinks firstly are processed by the sink nodes to extract attributes from the query events. Then these processed queries are sent to the index node in highest level. The processing of queries can be divided into two types, attribute query and region query.

The first type, attribute query, is to acquire the given attributes without the accurate information of their generated regions. Query is sent to the index node in highest level is processed and the result of this processing is related to the level of detail of the data defined in the query. If the data accuracy of present resolution level matches the requirement, the query is forwarded to the storage node at same level of the index node. Otherwise, the result of this processing is an approximate result that indicates which regions in the network are most likely to provide a more accurate response to the query. The query is forwarded to the index node that is located at the region in lower level. Then the index node does the same process until the query reaches the storage node whose data are most likely to satisfy it.

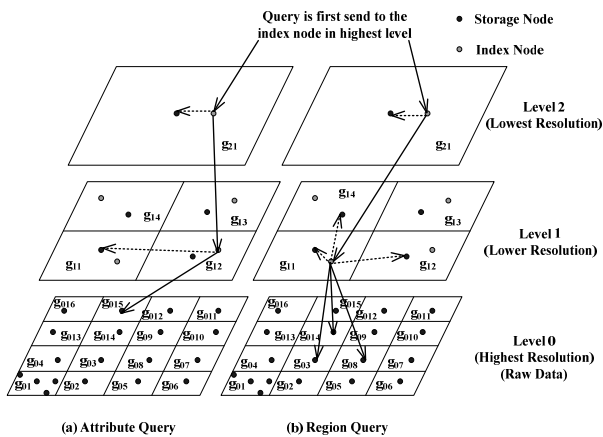


Figure.4 Query processing.

As in Figure 4 (a), query is sent to the index node in g_{21} . If the data granularity in level 2 matches the requirement in the query, the query is sent to the storage node in g_{21} . If it does not match, the index node in g_{21} searches its index information and finds out the region most likely to store the required data which is covered by g_{12} in level 1. The query is then forwarded to the index node in g_{12} and the same process is doing as in level 2. The process result is the storage node in grid g_{11} for lower resolution data or the storage region in level 0, as the figure shows, g_{015} , for the highest resolution data.

The second type is region query. It is the query to retrieve all the attributes included by the events happened in a region. The query is sent to the index node that is located at the interested region in the lower level. According to the user defined level of detail and the index information, the query is forwarded to the corresponding storage nodes in the same resolution level or to the index node of a more accurate region in lower level. At lower resolution level (level 1), if the data accuracy do not match the requirement, the query is directly forwarded to the storage nodes in finest resolution level.

Let us assume a query of retrieving all the attributes of highest resolution happened in region covered by g_{11} . The query is firstly sent to the index node in g_{21} in level 2. After some processing, it is forwarded to the index node in g_{11} in level 1. According to the index information stored in g_{11} , it can be known that the required data is stored at the

storage nodes in g_{03} , g_{08} and g_{014} . Then the query is directly forwarded to these three grids to fetch the required data. If the query is to retrieve all the attributes of lower resolution happened in region covered by g_{11} , the data stored in the storage nodes in g_{11} , g_{12} , g_{14} could satisfy the query.

6. Performance evaluation

We simulate an implementation based on OMNET++ to measure the performance of our DIMSA scheme. We compare our scheme with GHT and define the query acquired data in highest resolution level. In the simulation, MAC protocol is based on IEEE 802.11, and the transmission range of each node is 40m. 2000 sensor nodes are uniformly distributed over a 320×320 m² flat field. The field is divided into 20×20 m² grids. Once the sensor nodes are deployed, they are immovable. The whole network is a fully connected network. Each data packet has 16 bytes. Every event has 4 attributes.

Two metrics are used for evaluation of performance, average data storage energy consumption and average data query delay. Average data storage energy consumption is defined as average dissipated energy per node in the network to transmit and store the data. Average data query delay is defined as the average time between the moment a query packet is injected into the network and the moment the sink receives the data packet.

Figure 5 presents the average data storage energy consumption when the number of events varies from 20 to 100. In Figure 5 (a), DIMSA consumes more energy than GHT because we make each event includes no same attributes. The number of data packets to deliver attributes is greater than the data packets to deliver events. Therefore, forwarding and storing attributes in DIMSA consumes more energy than forwarding and storing events in GHT. While in Figure 5 (b), since we define every event has two same attributes, DIMSA consumes less energy than GHT. In GHT, the attribute included in many events have to be replicated and stored at different places in the network for each individual event. While in DIMSA, the attribute need not be replicated at multiple nodes because the data allocation and storage are based on attributes.

In both schemes, the energy consumption increase with the number of events increase.

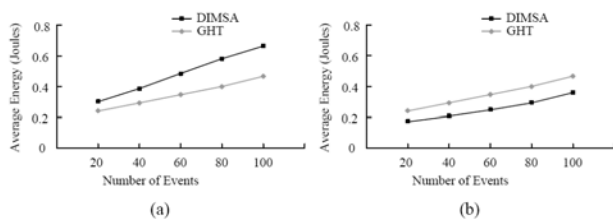


Figure.5 Average data storage energy consumption.

As shown in Figure 6, the average data query delay both in attribute query and region query of DIMSA is less than GHT. In Figure 6 (a), the average data query delay in DIMSA is dramatically improved than GHT in attribute query due to the fact that queries are lack of accurate location information of the attributes. GHT must flood the query to all the storage nodes in the network to find out the required data. As the events increases, the query delay in both schemes increase.

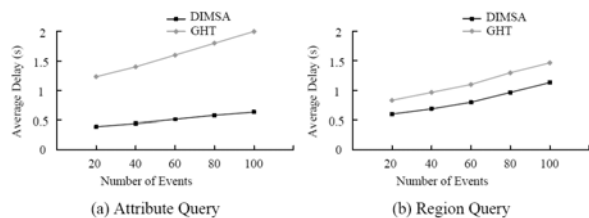


Figure.6 Average data query delay.

7. Discussion

Our work is preliminary, and there are many aspects that need to be considered such as data update problems and network load balance. We will address us to a further study of these problems and the improvement of the performance of DIMSA.

Due to the characteristics of energy-constraint and storage-constraint in WSNs, the data stored in the network, including attribute data and index data, are updated periodically. If data are updated too frequently, it must consume more energy. Or else, it could not satisfy the user queries. Data update rate is related to many factors, such as practice of application, user query style and characteristics of attributes. A finer data update rate can save energy and prolong the network life.

If the data in the network are skew, the attribute allocation function would cause “hotspots” and inefficiency. Large amount of attributes may be hashed to the same area in the network, making the workload of the network unbalanced. One of our future studies is to optimize attribute allocation considering the work load of different regions.

8. Conclusion

In this paper, we present a Distributed Index based Multi-resolution Storage Architecture (DIMSA) for wireless sensor networks. The proposed scheme is designed for minimizing the communication cost involved in data storage and query in a large-scale data-intensive sensor network. Simulations show that DIMSA is more energy efficiently when some attributes are included in many events and it is more efficient in querying data than the traditional data-centric storage scheme in WSNs.

Acknowledgments

This work is supported by the National Basic Research Program of China (973 Program) under Grant No. 2004CB318201.

References

- [1] D. Ganesan, D. Estrin, and J. Heidemann, “DIMENSIONS: Why do we need a new Data Handling architecture for Sensor Networks?”, *Computer Communication Review*, January 2003, Vol.33, pp.143-148.
- [2] S. Ratnasamy, D. Estrin, R. Govindan, B. Karp, S. Shenker, L. Yin, and F. Yu, “Data-centric storage in sensornets”, *First Workshop on Sensor Networks and Applications (WSNA)*, Atlanta, GA, September 2002.
- [3] D. Ganesan, B. Greenstein, D. Perelyubskiy, D. Estrin, and J. Heidemann, “An Evaluation of Multi-resolution Storage for Sensor Networks”, in *ACM SenSys 2003*.
- [4] Hongbo Luo, Guoliang Xing, Minming Li and Xiaohua Jia, “Dynamic Multi-resolution Data Dissemination in Storage-centric Wireless Sensor Networks”, *MSWiM’07*, 2007, pp.78-85.
- [5] R. Biswas, K.Chowdhury and D. P. Agrawal, “Attribute Allocation and Retrieval Scheme for Large-Scale Sensor Networks”, *International Journal of Wireless Information Networks*, October 2006, v 13, pp. 303-315.
- [6] S. Ratnasamy, B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin and F. Yu, “Data-centric storage in sensornets with GHT, a geographic Hash Table”, *Mobile Networks Applications*, August 2003, Vol.8, pp.427-442.
- [7] Kai Xing, Xiuzhen Cheng and Jiang Li, “Location-Centric Storage for Sensor Networks”, *IEEE International Conference on Mobile Adhoc and Sensor Systems*, 2005.
- [8] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann and F. Silva, “Directed diffusion for wireless sensor networking”, *IEEE/ACM Transactions on Networking*, February 2003, vVol.11, pp. 2-16.

- [9] S. R. Madden, M. J. Franklin, J. M. Hellerstein and W. Hong, "TinyDB: An acquisitional query processing system for sensor networks", *ACM Transactions on Database Systems*, March 2005, Vol.30, pp.122-173.
- [10] S. Shenker, S. Ratnasamy, B. Karp, R. Bovindan and D. Estrin, "Data-centric storage in sensornets", *Computer Communication Review*, January 2003, Vol.33, pp.137-142.
- [11] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable peer-to-peer lookup service for Internet applications", *Proceedings of the ACM SIGCOMM 01 Conference*, San Diego, CA, August 2001.
- [12] R. Ratnasamy, P. Francis, M. Handley, R. Karp and S. Shenker, "A Scalable content-addressable network", *Proceedings of the ACM SIGCOMM 01 Conference*, San Diego, CA, August 2001.
- [13] B. Karp, H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for wireless networks", *Proceedings of the Annual International Conference on Mobile Computing and Networking*, 2000, pp.243-254.
- [14] D. K. Ganesan, "Scalable Data Handling in Sensor Networks", Ph.D. dissertation, University of California, Los Angeles, 2004.
- [15] X. Li, Y. Kim, R. Govindan, and W. Hong, "Multi-dimensional range queries in sensor networks", *Proceedings of the 1st international conference on Embedded networked sensor systems*, Nov 2003.
- [16] W. Zhang, G. Cao and T. La Porta, "Data dissemination with Ring-Based Index for wireless sensor networks", *IEEE Transactions on Mobile Computing*, July 2007, Vol.6, pp.832-847.
- [17] B. Greenstein, d. Estrin, R. govindan, S. Ratnasamy and S. Shenker, "DIFS: a distributed index for features in sensor networks", *Proceedings of the 1st IEEE International Workshop on Sensor Network Protocols and Applications*, Anchorage, Alasha, May 2003.
- [18] M. I. Khan, W. N. Gansterer and G. Haring, "In-network storage model for data persistence under congestion in wireless sensor network", *Proceedings of the 1st International Conference on Complex, Intelligent and Software Intensive Systems*, 2007.
- [19] A. Ghose, J. Grossklags and J. Chuang, "Resilient data-centric storage in wireless ad-hoc sensor networks", *Proceedings of 4th International Conference on Mobile Data Management*, Melbourne, Australia, 2003.
- [20] Y. Yu, V. K. Prasanna and B. Krishnamachari, "Energy minimization for real-time data gathering in wireless sensor networks", *IEEE Transactions on Wireless Communications*, November 2006, Vol.5, pp. 3087-3096.
- [21] R. Biswas, N. Jain, N. Nandiraju and D. P. Agrawal, "Communication architecture for processing spatio-temporal continuous queries in sensor networks", *Special Issue of the Annals of Telecommunications on Sensor Networks*, July-Aug. 2005, Vol.60, pp.901-927.
- [22] Crossbow Technology Inc. <http://www.xbow.com/>