Efficiently Window Queries Processing in Dynamic Wireless Sensor Networks

Ning-Han Liu

Department of Management Information Systems
National Pingtung University of Science & Technology
Taiwan, R.O.C.
gregliu@mail.npust.edu.tw

Abstract

Due to the proliferation of low cost wireless sensors, there is growing research interest in their applications, for example, in home healthcare and location tracking. However, due to sensors' energy resource constraint, some possible applications of sensors have been restricted. In particular, in applications concerning deployment of mobile sensors in dynamic environments, high amounts of energy are consumed by sensors to maintain routing tables. Although existing methods have been proposed to query data from sensors without the use of any routing tables, these methods typically require redundant data to be sent back to the sink and not all of the aggregation functions could be executed precisely. In this paper, we modify an existing method to provide more accurate query answers and extend the lifetime of a wireless sensor network (WSN). According to our simulation, this method outperforms the existing method our approach modifies.

Keywords: *Wireless Sensor Networks, Window Query, Query Processing, Aggregation Function*

1. Introduction

Wireless sensor networks [2, 5] consist of low-powered sensors, limited in computation, communication and detection ability, made to operate in unmanned environments to detect and gather information that is consequently passed to the sink. The main function of these sensors is to detect environmental changes indicated by temperature, humidity, sound, pressure, *etc.* [7, 8, 9]. Generally speaking, detected information, which may be dealt with by basic sensor operations, are transmitted to midpoint or the base station, which is further processed and analyzed [6, 14].

There are various researches undergoing in the area of wireless sensor networks. These researches cover work from improving sensor hardware components to systematic structures of the software that the sensors run on. However, a more important underlying research issue is the limited battery capacity of wireless sensors. A sensor cannot provide a service if it has no remaining power, and thus how to reduce energy consumption is an important issue to be addressed [5].

With the development of WSNs, the location of sensors can be obtained by the global positioning system (GPS) and, thus an increasing amount of applications are associated with the geographical position of the sensors. For example, to answer queries about the average temperature in a certain area, window query is a particular type of function that can be used. It answers spatial queries, *i.e.*, queries about all of the information detected by sensors in any given rectangular area. Generally speaking, the data obtained by a window query often needs further processing in order to retrieve useful information; for example, to get maximum, minimum, average values, *etc*. This is the same as the application of an aggregation function to calculate the data.

In the traditional database system, all the data may be easily acquired from the server and calculated by aggregation functions. However, for a WSN, doing so will result in excessive energy consumed to transfer all necessary data back to the base station. For this reason, the research presented in [3, 4, 13] hopes to perform such operations in the network directly, and only route the final result back to the query node in order to reduce energy consumption. The methods used can be divided into two primary steps. The first step is query propagation, which sends the query via the specific route structure (such as the tree structure) to sensors in the area of interest. This is followed by data collection. During the process of data retrieval, the relaying node in the network uses aggregation functions to calculate the collected data.

These methods usually need to maintain the network topology; when the sensors are static, the maintenance cost is acceptable. However, the wireless sensors may also be portable; for example, sensors placed on vehicles or the movable objects. In such environments, the maintenance of the wireless network topology will expend significant amounts of resources. If the positions of sensors change rapidly, the use of a spatial query will be a problem. In this regard, Perkins, *et al.*, [18] developed a method that operates under the condition of movable sensors. This method allows the window query to operate without any information of the network topology as well as query and perform the aggregation operation within the network. This method utilizes sensors that know their own positions to send queries within their querying range, collect data coming back from the queries and delegate aggregation operations to selected nodes in the querying range. This method restricts the maximum and minimum number of aggregation functions used in operation due to collection of repeated data. Nevertheless, this drawback is resolved by other approaches, such as the data sketch [12] that enables data compression and data filtering. However, such approaches lose the advantage of obtaining aggregation operations in the sensor network and fail to reduce energy consumptions because of repeated data transmissions.

Our research is based on the condition of the environment with portable sensors. Through putting in a serial number and calculating distance between nodes, the queried node determines whether to pass the data back or to the sink in order to prevent repeated transmission of data. It can obtain the majority aggregation operations such as maximum, minimum, single value count, the average and total, *etc*. In the case of sending unrepeated data, it also reduces the energy consumption and improves the service time of sensors.

The rest of this paper is structured as follows. The review of related research work of window query is given in Section 2. Section 3 discusses research questions arising from present methods and proposes our improved measures. The experimental results are reported and discussed in Section 4. Finally, the conclusion and the future work are provided in Section 5.

2. Related Works

In the research of the WSNs, the techniques dealing with window query processing can be divided into two classes. One is the infrastructure-based query processing technique and the other, the infrastructure-free query processing technique. These techniques are discussed as follows.

2.1 Infrastructure-based Query Processing Technique

Generally, queries (including the window queries in the WSN) are processed by propagating query messages directly to the sensors, which then send the data back to query nodes to process the data altogether. More resources are consumed in this way. At present, current methods which can efficiently deal with the window query in the WSN are typically built to operate on the infrastructure network. The approach used is divided into two steps. Firstly, the routing method is utilized to construct the information transmission route (*e.g.*, Tree or Cluster), and propagate query messages to the nodes in the query range. After the route has been calculated, the data would then be processed using the aggregation function by the mid-node which returns the results to the query nodes in order to reduce energy consumption. According to the infrastructure size, these kinds of methods can be further divided into the network spanning infrastructure (NSI) and window spanning infrastructure (WSI) [18].

The NSI approach [3, 11] is to construct and maintain a network system. Its shortcoming is obvious in the maintenance of a dynamic network because a large amount of calculations are incurred to maintain the correct routes. In addition, query messages are sent to too many relaying nodes when the window query is implemented on it (as shown in Figure 1(1)). The process is as follows. The sink node sends query messages along the network topology to the sensor nodes in the query window. As nodes in a query window lie in different branches of the tree, query messages are sent from variant routes to the nodes in the query window. The sensor nodes in the query window then return data to the upper level nodes, which send them query messages. These nodes then send data directly or results calculated by aggregation function (for example, maximize) up to the upper level nodes. This process continues until all results have been sent back to the sink node.

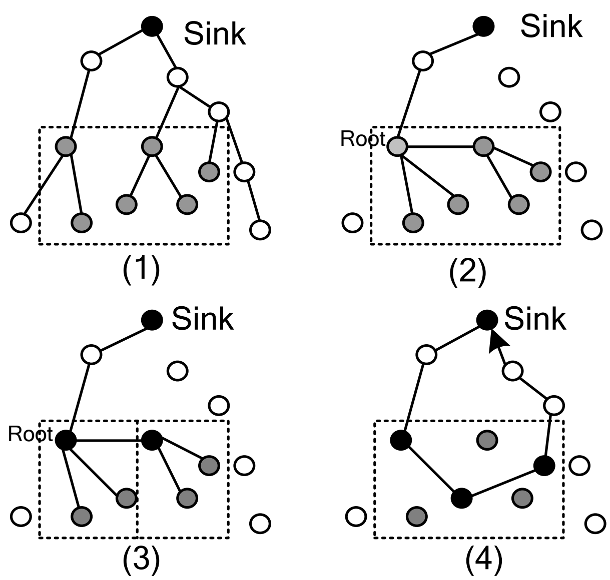

Figure 1. Window Query Processing

In contrast, the WSI method [1, 4, 16] looks for one route to the query window, in which only one message is sent to a node of the query window. Then, the node generates a route within the query window from the routing table and sends query messages to other nodes in the query window. Here, the data collection and aggregation operations are similar to those of

NSI. Referring to Figure 1(2), this data is propagated along the tree topology. In Figure 1(3), the nodes in query window are further clustered whereby the cluster head collects the data.

2.2 Infrastructure-free Query Processing Technique

When the sensors are movable (*i.e.*, dynamic environment), applying the routing method for data collection causes excessive energy consumption due to the maintenance of the network. The infrastructure-based technique is not suitable for query processing under this environment. As an example, consider an infrastructure-free window query processing technique that utilizes flooding. In this approach, query messages are sent by the query node or root to the neighbor nodes (which are in the broadcasting range) such that all the nodes which receive the query messages then broadcast outwards in the same way until all nodes of the WSN receive the query messages. Following this, the nodes in the query windows then send data back to the query node whereby the query node will perform the aggregation operation further if needed. As data is sent back to the query node separately a large amount of energy is consumed and, as a result, the service time of the WSN is shortened.

IWQE (Itinerary-based Window Query Execution) [18] is a window query approach designed for the dynamic network. It utilizes the geo-routing protocol [17] to send the query to the nodes in the query window. The query window is split into several sub zones. The nodes in these zones are divided into two sets: (1) a set of sensor nodes that are query node (*i.e.*, Q-nodes); and (2) the other set of nodes that are data nodes (*i.e.*, D-nodes). A Q-node is responsible for collecting and aggregating the data sensed by nearby D-nodes, and then forwards the aggregated result and query message to the next proper Q-node. The itinerary in the sub zone is planned in advance. By setting the appropriate itinerary width and itinerary route, most of the data sensed by the D-nodes can be included in the aggregation operation. This effectively lowers the resource consumption for that the query propagation and data collection is executed along one itinerary and combined in one stage, shown as Figure 1(4). In addition, the Q-nodes in the sub zones are chosen dynamically so the method does not need to set up the route or maintain the route table in advance.

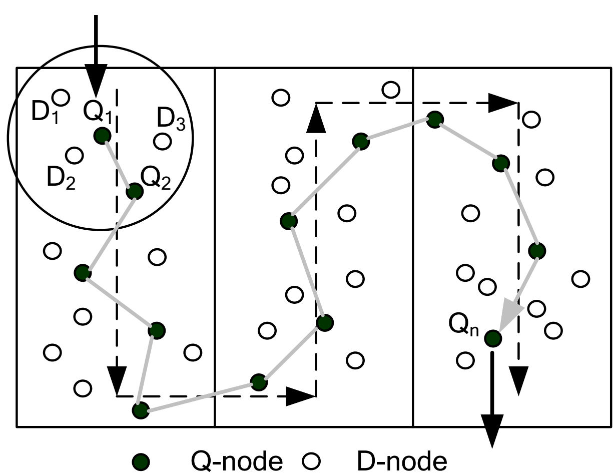

Figure 2. IWQE Itinerary

With reference to Figure 2, we now show how IWQE propagates the query and collects data in a query window.

The sink node splits the query window into several zones (*e.g.*, Figure 2 shows three zones) and plans the direction about the query propagation (*e.g.*, the direction is downwards in the first sub area. As it passes through to the second sub area, the direction in the second sub area goes upwards and, in the third sub area, the direction goes downwards (illustrated as dotted lines in Figure 2). Q_1 in Figure 2 is the first node to receive the query (*i.e.*, gets the maximum of sensed data by the sensors in the query window) sent from the sink node. At this moment, Q_1 is treated as Q-node and it broadcasts the query to the neighbor nodes. In order to avoid waiting for the responses of neighbor nodes indefinitely, Q-node restricts the waiting time. When the waiting time expires, the query is then sent to the next node. Equivalently, we can limit the nodes that should respond to Q_1 in a round area. Then, Q_1 will obtain the aggregation operation after receiving the data sent back from these nodes (*e.g.*, Q_2, D_1, D_2 and D_3 in Figure 2). For instance, if the data sent back by the previous four nodes is 10, 12, 9, 18 and Q_1's data is 8, then Q_1 will get the maximum 18 from these data.

According to the data (including the nodal position) passed by the neighbor nodes, Q_1 looks downward for the next Q-node. The farthest node will be chosen to avoid too much overlapping area between Q_1 and the next Q-node, such as Q_2, as shown in Figure 2. At this moment, Q_2, the D-node in the query area of Q_1, is changed to Q-node, and Q_2 will receive the query and the result (maximum at present) sent from Q_1. The steps mentioned above are repeated. Eventually, an itinerary goes out along the planned direction (the gray line in Figure 2), and the maximum is finally obtained. After the last Q-node finishes its process (Q_n node in Figure 2), the result is sent back to the sink node by the Geo-routing method and the query processing finishes entirely.

The method to separate the sub zones and the parameter of the waiting time affects the correctness and efficiency. However, these details will not be discussed here. Readers are referred to the original paper [18] for further information.

3. Improvement of Infrastructure-free Query Processing Technique

From related work discussed in Section 2, IWQE is a more ideal method for processing aggregation operation of window query in a dynamic network. Nevertheless, some problems exist in IWQE. For one, IWQE is unable to obtain some aggregation operations. In this section, we state these problems and propose a new mechanism to solve these problems. This mechanism reduces overall energy consumption, which is a key design criterion of WSN.

3.1. Issues in IWQE

In IWQE, data sensed by a D-node is conveyed repeatedly to different Q-nodes because the data is collected within an area step-by-step and forwarded along the planned route. The data collected by this method is unable to be processed by some of the aggregation functions. Figure 3 illustrates why mistakes happen in the data collection and how these mistakes cause computation errors in some aggregation functions.

In Figure 3, Q_1 sends the query to neighbor nodes. These include D_1, D_2, D_3 and Q_2 that are in the transmission range of Q_1 so that Q_1 receives the data sensed by the four nodes (e.g. 10, 12, 9, 18 and 8 sensed by itself). Then, Q_2 is selected as the next Q-node and sends the query to the neighbor nodes. D_2, D_3 and D_4 would then fall into the transmission range of Q_2 (Q_1 which was a Q-node does not need to process the query) and transmit sensed data (*e.g.*, 12, 9, 7) to Q_2.

D_2 and D_3 are in the transmission ranges of Q_1 and Q_2 and so they will report their sensed data twice. Nevertheless, if the query of the sink node is to obtain aggregation operations such as the maximum and the minimum of the collected data, the result will still be correct. For example, Q_1 propagates the aggregated maximum and minimum 18 and 9 respectively to Q_2.

Then, Q_2 recalculates the maximum and minimum with newly collected data to be 18 and 7. In this instance, the IWQE method obtains the correct aggregate results of the maximum and the minimum. However, in contrast, the total or average aggregated would be incorrect because the repetitive propagation of D_2 and D_3 causes mistakes in the answers. Furthermore, these query errors magnify as the nodal quantity increases for the overlapping query ranges.

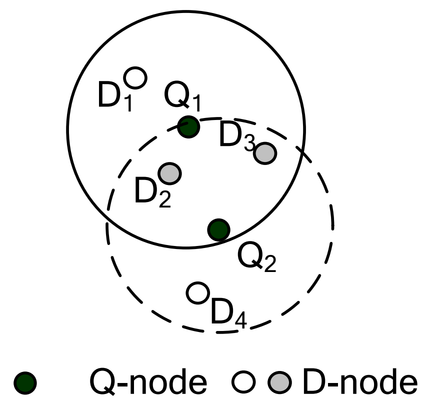

Figure 3. The Problem of IWQE to Convey Data Repeatedly

There are existing methods to store the collected data and serial number of propagating nodes efficiently by using hashing functions [10, 12] and convey to the following Q-node to get rid of repeated data collected. In theory, in order to eliminate the mistakes, the nodes that have returned their data should be recorded. Nevertheless, it needs to be noted that simplifying the quantity of records will inevitably cause collision at the time of hashing and produce errors. These methods may solve the problem of repeated data propagation but the drawback is that the amount of data propagated among Q-nodes increases heavily and will cause excessive energy consumption. Furthermore, the problem of repeated propagation of some D-nodes will still be prominent.

3.2. The First Improvement Approach of IWQE

From the above analysis of IWQE, it can be observed that the preventing of duplicate sensor readings can improve the query limitation and increase the lifetime of the WSN.

In order to prevent data retransmissions, each D-node needs to decide whether sensed data should be sent back or not. We use a serial number to represent a query. When the sink node propagates a query and then simultaneously generates a serial number at random, each query is associated with a different serial number. The Q-node broadcasts the query with the associated serial number and the D-node records the serial number upon receiving the data until the next serial number arrives. Utilizing this serial number can help the D-node to discriminate a new query from processed queries and decide whether the sensed data has already been responded to by the corresponding query or not. If the D-node receives the same serial number, it is then not required to process the same query again.

As shown in Figure 3, D_2 and D_3 record the serial number (*e.g.*, 11028) upon receiving the query transmitted by Q_1. Following this, they receive the same query transmitted by Q_2. By comparing the recorded serial number, they will then dismiss the newly accepted query because the query would have been processed before.

Utilizing this simple technique appears to have solved the problem of data retransmission and some unworkable aggregation functions in IWQE. However, a new problem arises from this approach. Some D-nodes have already succeeded in passing the data back and no longer respond to the same queries propagated from other Q-nodes, i.e. they fail to transmit data back. This causes errors in data aggregation because the data sensed by the above sensors is not included. Figure 4 shows how this problem occurs.

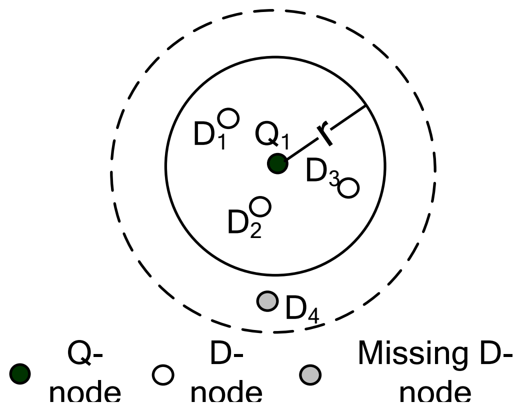

Figure 4. Some Data are Unable to Transmit Back After Adding Serial Numbers

According to the characteristic of the radio wave, the query produced by IWQE Q-nodes is broadcast to the larger range than the default delimitation of radius, denoted by the dotted circle in Figure 4. D_4 receives the query propagated by Q_1 and sends data back to Q_1. The time pertaining to data received is behind Q_1's receiving time and so the data is excluded from the aggregating result. As the result is calculated by aggregating the sensed data of D_1, D_2, D_3 and Q_1, it may have already been propagated to the next Q-node. D_4 will not know Q_1 abandons this data, but will record the serial number. After that, D_4 will not respond to the same queries propagated from other Q-nodes. Nodes, such like D_4, are termed Missing D-nodes. The problem is the absence of some sensor reading data which in effect, causes the result to be miscalculated.

3.3. The Second Improvement Approach of IWQE

Intuitively, letting the Q-node notify the D-node while abandoning the propagated data seems to solve the problem. However, such a method incurs the overhead of wireless traffic and causes the problem of missing D-nodes. For example, a D-node, which receives the notification after refusing other Q-nodes' queries, will become a missing D-node.

As a solution, we let each D-node decide whether the serial numbers of the queries should be recorded. If the D-node does not lie in the effective query range of the Q-node, sensed data passed by the D-node is then abandoned by the Q-node and the D-node need not to respond the Q-node's queries. This method can solve the missing D-node problem. For example, D_4 in Figure 4 obtains the distance to Q_1 and then abandons the serial number of Q_1's query because the distance is larger than the radius r, the effective query range of Q_1. On the contrary, D_1, D_2 and D_3 would lie in the effective query range and would record the serial number.

The following describes how the D-node learns if one's position is within the range of a valid radius:

(1) If all nodes have GPS coordinates, the Q-nodes propagate their own positions and the valid radiuses while transmitting queries to the neighbor nodes. According to the GPS coordinates of a Q-node, a D-node then calculates the distance between them and compares the distance with the valid radius to decide whether it is in the effective range or not.

(2) If nodes do not have GPS coordinates, the free space propagation model [15] can be used to calculate the distances between the radio wave transmitters and the receivers and then the obtained distances can be compared with the valid radiuses passed by the Q-nodes. The transmitting distance formula is shown as follows:

$$Po - Co + Ao - 92.4 - 20\log F - 20\log D + Ar - Cr = Rr \quad (1)$$

where
Po : Transmitter's radio power output in dbm
Co : Transmitter's cable attenuation loss in db
Ao : Transmitter's Antenna gain in dbi
F : Used frequency in GHz
D : Distance in km
Ar : Receiver's Antenna gain in dbi
Cr : Receiver's cable attenuation loss in db
Rr : Receiver's signal strength in dbm

Summarizing the above-mentioned approaches, we have utilized the method of adding serial numbers and calculating the distance between the Q-node and the D-node to determine whether the D-node should response to the query of the Q-node or not. This reduces the nodal computation and transition to not only support more kinds of aggregation functions than IWQE but also reduces the energy consumption to prolong the WSN lifetime.

4. Experiment and Discussion

It is expensive to construct a full-scale, real-time sensor network for our experiments. Thus, we utilize a simulation to carry out our experiments on a large numbers of nodes. In the experiments, we refer to the improved IWQE as E-IWQE (Enhanced-IWQE). We compare the performance of IWQE with E-IWQE with respect to the degree of improvement and the energy consumed.

4.1. Experiment Environment and Parameter Settings

We use Microsoft Visual Studio 2005 to develop the simulation programs for the experiments. The hardware is Pentium 4 (with 1.5 GHz CPU, 1 Gig RAM) and operating system is Window XP SP3.

Wireless sensors are located randomly in a simulation map (namely, the distribution area of sensors) of size 3200 m x 3200 m to simulate 1000, 2000 and 3000 nodes. The sensors move in random directions and speed, and the highest moving speed of each node is 5 m/sec. The data processing time of a query by the sensor mote has been set to 0.5 seconds. The size and the position of a query window has been generated and situated randomly in the simulation map. The quantity of sub zones in the query window and Q-node’s query range has been set according to the settings described in [18].

4.2. Experimental Results from Energy Consumption Simulation

Based on the performance datasheet of wireless sensors today (*i.e.*, the products of Crossbow Company, such as mica, iris, imote 2, *etc.*), the energy cost of radio communication is thousands to tens of thousands times that of any other operations performed by sensors [14]. Thus, in our experiments, energy consumption is measured by the communication times between nodes rather than by the combined cost of all operations performed by sensor nodes. It follows that the greater the amount of radio communication, the higher is the overall energy consumption. In order to observe the effect of some unusable nodes, the communication ability of each node has been set to 100 times at most. When the communication times have been exceeded, the sensor node is then rendered inoperable and would be unable to communicate further. 100 simulation runs have been performed in the experiment. The state of each sensor is reset and the position is relocated randomly at every new simulation run. Every simulation run generates 100 window queries including aggregation functions. The statistics of the experiment have been summarized in Table 1.

As shown in Table 1, after one hundred queries, the number of inactive sensors has been reduced quite substantially because E-IWQE reduces the amount of the useless transmissions from D-nodes to Q-nodes. Apart from this, it can be observed that the number of inactive nodes grows proportionately to the increase in the number of nodes in a query window. This is mainly because, with more nodes in a unit area, the employed nodes increase when doing query propagation.

Table 1. Comparison of Nodal Energy Consumption

Number of Nodes	Average inactive nodes in a simulation run	
	IWQE	**E-IWQE**
1000	10.2	0.4
2000	30.9	1.3
3000	50.6	3.4

Using IWQE, there could not be any node in a larger area on a default route. Therefore, no Q-node can be found in this area. Accordingly, queries can not be propagated and no result can be sent back. At the beginning, deploying sensors randomly is less of an issue but, as the processing of queries continues, the number of inactive nodes making unfinished queries increases. Table 2 shows the number of unfinished queries with the simulation of 10,000 queries.

Table 2. Comparison of the Number of Unfinished Queries

Number of Nodes	The number of unfinished queries	
	IWQE	**E-IWQE**
1000	320	23
2000	175	15
3000	84	9

Table 2 demonstrates that E-IWQE outperforms IWQE. It also indicates that the inactive nodes not only influence the accuracy of data collection but also affect the smoothness of the query processing process. As a result, unfinished processes which cannot gather the query results to the sink node further consume excessively energy.

4.3. Inquire About the Experimental Result of Wrong Rate

Both IWQE and E-IWQE have drawbacks during window queries processing. For IWQE, the drawbacks come from two situations. One situation is the loss of data. Some nodes may be in the query window but fail to be covered by the effective ranges of any Q-nodes. As a consequence, the data collected by these nodes are not being passed back or calculated. Such situations include nodes that are originally in the query window but had to move away while query processing. The other situation is the repeated transmission of data. For E-IWQE, only the first situation applies.

For further experimentation, using the same simulation environment as described in Section 4.2, we simulate the aggregation function (SUM) to sum up the sensed data in the query window. The result of the aggregation function (average) is omitted, because it can be derived from calculating the sum divided by the node number in the query window. We then calculate the difference in simulated data to real data as a mean percentage error. Table 3 shows the experimental results.

As shown in Table 3, the mean percentage errors of IWQE are all over-evaluated due to repeatedly transmitted data. Although the missing D-nodes fail to transmit their sensed data, the influence of missing data on aggregated results is less than repeatedly transmitted data. In addition, as the number of nodes increases, the mean percentage error also changes. It is a consequence of the increase in repeated transmitted data, due to the probabilities of D-nodes covered by different Q-nodes increase. In contrast, the mean percentage errors of E-IWQE are underestimated because some nodes are not covered by any Q-nodes, and are only slightly changed with the increasing sensor density.

Table 3. The Experimental Result of Aggregation Function (Sum)

Number of Nodes	Mean percentage errors of queries including aggregation function (sum)	
	IWQE	**E-IWQE**
1000	+24.2%	-4.2%
2000	+39.8%	-3.4%
3000	+48.6%	-3.9%

**The negative value shows the simulated result lower than the real data, and the positive value shows the simulated result higher than the real data.*

The experimental results obtained demonstrate that our simple approach decreases the energy consumption and provides more correct results of queries, including aggregation functions.

5. Conclusion and Future Work

The advent of sensor hardware technology and growing research in WSN technology has the potential to broaden the applications of WSNs. In these applications the support for various queries in a dynamic network is a primary concern. Furthermore, simplifying data collection and communication processes can prolong the lifetime of a WSN. In this paper, we have applied a simple approach to improve on a powerful window query method used in dynamic networks and, to that effect, enable important information to be generated to aid D-nodes in determining the need to transmit sensed data. The simulation experimental results have demonstrated that E-IWQE outperforms the original method with respect to energy consumption and query correctness.

Nevertheless, though E-IWQE has already made apparent contributions, we note that there are still errors in the query results. For future work, this will be further investigated to create a method that can obtain the best query results with no resultant increase in network traffic.

Acknowledgements

This work was supported by the NSC of Taiwan under contract numbers NSC96-2218-E-020-004 and NSC98-2221-E-020-025-MY2. Furthermore, thank Tien-Cheng Huang for his implementation of the experiment.

References

[1] A. Coman, J. Sander and M. A. Nascimento, "Adaptive processing of historical spatial range queries in peer-to-peer sensor networks", Distributed and Parallel Databases, vol. 22, no. 2-3, **(2007)**, pp. 133-163.
[2] A. Hac, "Wireless Sensor Network Designs", John Wiley & Sons, Inc., **(2003)**.
[3] D. Goldin, M. Song, A. Kutlu, H. Gao and H. Dave, "Georouting and delta-gathering: Efficient data propagation techniques for geosensor networks", In Proceedings of GeoSensor Networks Workshop, Portland, Maine, **(2003)**.
[4] D. Petrovic, R. C. Shah, K. Ramchandran and J. Rabaey, "Data funneling: Routing with aggregation and compression for wireless sensor networks", In Proceedings of IEEE Sensor Network Protocols and Applications, Anchorage, AL, **(2003)**.
[5] G. J. Pottie, "Wireless sensor networks", IEEE Information Theory, **(1998)** June, pp.139-140.
[6] I. F. Akyildiz, W. Su and Y. Sankarasubramaniam, "A survey on sensor networks", IEEE Communications Magazine, vol. 40, no. 8, **(2002)**, pp.102-114.
[7] M. Bawa, A. Gionis, H. Garcia-Molina and R. Motwani, "The price of validity in dynamic networks", Journal of Computer and System Sciences, vol. 73, Issue 3, **(2007)** May, pp. 245-264.
[8] N. H. Liu, C. Y. Li, S. J. Hsieh, C. F Tsai and M. H. Shao, "Long-term audio observation by wireless sensor networks with filtering strategies", Ad Hoc & Sensor Wireless Networks, vol. 12, no. 1-2, **(2011)**, pp. 125-149.
[9] N. H. Liu, C. A. Wu and S. J. Hsieh, "Long-term animal observation by wireless sensor networks with sound recognition", Lecture Notes in Computer Science, Springer-Verlag, vol. 5682, **(2009)**, pp. 1-11.
[10] N. H. Liu, C. Y. Li and S. J. Hsieh, "Design of wireless sensor networks with filtering duplicate audio signals", Lecture Notes in Computer Science, Springer-Verlag, vol. 5879, **(2009)**, pp. 469-479.
[11] S. Madden, M. J. Franklin, J. M. Hellerstein and W. Hong, "TAG: a tiny aggregation service for ad-hoc sensor networks", SIGOPS Oper. Syst. Rev., vol. 36(SI), **(2002)**, pp. 131–146.
[12] S. Nath, P. B. Gibbons, S. Seshan and Z. R. Anderson, "Synopsis diffusion for robust aggregation in sensor networks", In Proceedings of ACM SenSys, Baltimore, MD, **(2004)** November.
[13] S. R. Madden, M. J. Franklin, J. M. Hellerstein and W. Hong, "The design of an acquisitional query processor for sensor networks", In Proceedings of ACM SIGMOD Conference, San Diego, CA, **(2003)** June, pp. 491-502.
[14] S. Tilak, N. B. Abu-Ghazaleh and W. Heinzelman, "A taxonomy of wireless micro-sensor network models", ACM SIGMOBILE Mobile Computing and Communications Review, vol. 6, Issue 2, **(2002)**, pp. 28-36.
[15] T. S. Rappaport, "Wireless Communications: Principles and Practice", Communications Engineering and Emerging Technologies Series, Prentice Hall, **(1999)**.
[16] Y. Xu and W. -C. Lee, "Window query processing in highly dynamic geosensor networks: Issues and solutions", GeoSensor Networks, edited by A. Stefanidis and S. Nittel. CRC Press LLC, ISBN: 0-41532-404-1, **(2004)**, pp. 31-52.
[17] Y. Xu, W. -C. Lee, J. Xu and G. Mitchell, "PSGR: Priority based stateless geo-routing in wireless sensor networks", In Proceedings of IEEE International Conference on Mobile Ad-hoc and Sensor Systems, Washington, DC, **(2005)** November.
[18] Y. Xu, W. -C. Lee, J. Xu and G. Mitchel. "Processing Window Queries in Wireless Sensor Networks", In Proceedings of the 22nd IEEE Int. Conf. on Data Engineering (ICDE '06), Atlanta, GA, **(2006)** April.

Author

Ning-Han Liu received the BS and MS degree from National Taiwan Normal University in 1992 and 1995, respectively and the PhD degree in computer science from National Tsing-Hua University in 2005. He is an associate professor at Department of Management Information Systems, National Pingtung University of Science and Technology, Taiwan since 2010. His research interests include multimedia databases, audio processing in WSN, and artificial intelligence. Dr. Liu is a member of the IEEE Consumer Electronics Society. He is also a member of TAAI (Taiwanese Association for Artificial Intelligence).