Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 728521, 11 pages
doi:10.1155/2012/728521

Research Article

Intelligent Collaborative Event Query Algorithm in Wireless Sensor Networks

Rongbo Zhu

College of Computer Science, South-Central University for Nationalities, Wuhan 430074, China

Correspondence should be addressed to Rongbo Zhu, rongbozhu@gmail.com

Received 15 June 2011; Accepted 2 August 2011

Academic Editor: Yuhang Yang

Event query processing is a very important issue in wireless sensor networks (WSNs). In order to detect event early and provide monitoring information and event query timely in WSNs, an efficient intelligent collaborative event query (ICEQ) algorithm is proposed, in which sensor nodes that are near to the boundary of events are selected to accomplish complex event monitoring and query processing through intelligent collaboration. ICEQ will select range-nearest neighbors as the basic components of surrounding nodes. Then it will identify the gaps between the surrounding nodes and try to select the nearest neighbor collaborative nodes for enclosing the event in the node selection phase, which can avoid redundant sensor nodes to join surrounding nodes via identifying a set of association surrounding nodes between the nearest sensor nodes and the query events. Detailed experimental results and comparisons with existed algorithm show that the proposed ICEQ algorithm can achieve better performance in terms of query-processing time, average number of selected collaborative nodes, and query message consumption.

1. Introduction

The rapid development in computing, sensing, and wireless communication technologies has made the availability of wireless sensor networks (WSNs) [1, 2]. Their low cost, small size, and untethered nature make them sense information at previously unobtainable resolutions [3]. WSNs can be deployed in battlefield applications, and a variety of vehicle health management, habitat monitoring, environment monitoring, and condition-based maintenance applications on industrial, military, space platforms [4, 5].

In WSNs, an important task is to monitor dynamic and unpredictable events. Since the sensor network can be viewed as a distributed database [6, 7], due to the distributed nature and resource constraint of WSNs, we cannot maintain a centralized index to support query processing in WSNs [8]. Meanwhile, because of limitations imposed by impoverished computing environment, data collection and query in WSNs must support an unusual set of software requirements. Several previous works [8, 9] proposed declarative SQL-like query which enable users to acquire the information about the network through issuing queries to the sink. Even though each sensor node will be rather limited in terms of storage, processing, and communication capabilities, they will be able to accomplish complex event monitoring and query processing through intelligent collaboration, especially in large-scale WSNs.

Since sensor nodes have rigid energy constraints, it is hard to displace sensor nodes in the monitoring region [8, 10] due to unattended and untethered deployment. Most existing data collection systems [11–14] are query-based ones. Most existed event query schemes will decrease the lifetime of WSN greatly due to power consumption for the real-time monitoring. Traditional query-processing techniques of WSNs mainly deal with retrieving sensor node locations, sensing values, and aggregating the sensed values. However, in a lot of applications, users expect event and data information about areas of their interests. If a sensor is queried by many users, it may experience congestion and great power consumptions. Thus, a natural requirement is that each user sets a proper query range to both avoid overhead and achieve a global optimality at the same time.

In order to balance the inherent tradeoff between query reliability versus energy consumption in query-based

wireless sensor systems, an adaptive fault-tolerant quality of service (QoS) control algorithms based on hop-by-hop data delivery utilizing "source" and "path" redundancy is proposed in [11] to maximize the lifetime of the system. In order to allocate the multihop query range for each user such that a certain global optimality is achieved, Han et al. [15] investigated the NP complete scheme in its generic form and proposed a distributed heuristic to resolve the query problem. A data-querying scheme was proposed in WSN [16] where queries formed for each sensing task are sent to task sets, and the sensed data is retrieved from a sensor network in the level of detail specified by users, and a tradeoff mechanism between data resolution and query cost is provided. To disseminate data required for processing monitoring queries in a WSN, the notion of event-monitoring queries and algorithms were proposed in [17] for building and maintaining efficient collection trees that provide the conduit to minimize important resources such as the number of messages exchanged among the nodes or the overall energy consumption. In order to improve the performance of area query-processing in wireless sensor networks, an energy-efficient in-network area query processing scheme is proposed in [18], which partitioned the monitored area into grids and constructed a reporting tree to process merging areas and aggregations and conserve energy consumption. In [19], two approaches were proposed for processing such queries in WSN in-network instead of collecting all data at the base station of the spatiotemporal queries in WSN.

In order to improve the query performance, range nearest-neighbor (RNN) query [20], and nearest-surrounder (NS) queries [21], retrieve data based on location information in sensor networks. This kind of query schemes may enable us to find out surrounding nodes needed and is an efficient way to monitor event with less power consumption. In [22], a distributed Bayesian algorithm was proposed based on the concept of spatial correlation. However, it assuming that event measurements are either much larger or much smaller than normal measurements. In order to find out approximate real boundary, an efficient event query scheme is proposed in [23], which considers that WSN is composed of two distinct homogeneous regions. In order to achieve the boundary node efficiently, the localized fault-tolerant event boundary detection scheme was proposed in [24]. An efficient noise-tolerant event boundary detection algorithm was proposed in [25], which defined boundary nodes as sensor nodes which lie within real boundary with certain confidence interval guarantee. The problem of in-network processing and queries of trajectories of moving targets in a sensor network is investigated in [26], which exploits the spatial coherence of target trajectories for opportunistic information dissemination with no or small extra communication cost, as well as for efficient probabilistic queries searching for a given target signature in a real-time manner. In [27], collaborative query processing among multiple heterogeneous sensor networks was investigated and formulated into an optimization problem with respect to energy efficiency. WinyDB [28], a relational query-processing system on Windows CE-based personal digital assistants (PDAs) for sensor networks, is proposed to improve both the energy efficiency and the data quality collaboratively. To overcome the faulty data query problem to improve the accuracy of data query, an efficient fault-tolerant event query algorithm (FTEQ) was proposed in [29], which takes the short-term and the long-term spatial and temporal similarities between sensors and environment into consideration to decrease faulty detection rate and data query cost.

Although a number of event query schemes have been proposed to improve the query performance in WSNs, event query processing is still a very challenging task due to its complexity and ill-posed nature, and all of these works do not comprehensively consider the correlation between sensors and environment. And the most existing research work has focused on data aggregation to provide efficient data transmission. The overhead of query processing is generally ignored with the assumption that query transmission contributes to only a small portion of overall data transmission in the sensor network. However, there are many cases where this assumption does not hold any more. Therefore, the methods mentioned above all use statistical methods to differentiate whether sensor nodes are boundary nodes or not.

Another problem is that existed work always assumes that the monitoring nodes are often interested in obtaining either the actual readings or their aggregate values; from sensor nodes that detect interesting events, the detection of such events can often be identified by the readings of each sensor node. In such scenarios, each sensor node is not forced to include its measurements in the query output at each epoch, but rather such query participation is evaluated on a per epoch basis, depending on its readings and the definition of interesting events. However, in actual complex environment, due to the characteristics of WSNs, sensors are usually deployed in a noneasily accessible or harsh environment, and sensors are prone to failure, and these faulty sensors are likely to report arbitrary data very different from the true environmental phenomenon, and the faulty data of sensors are very common, which greatly influence the accuracy of data query. Hence, how to select appropriate nodes to accomplish complex event monitoring and query processing through intelligent collaboration is an important task.

Motivated by the above reasons, an efficient intelligent collaborative event query (ICEQ) algorithm is proposed, in which sensor nodes that are near to the boundary of events are selected to accomplish complex event monitoring and query processing through intelligent collaboration. ICEQ includes initial phase and node selection phase. In initial phase, ICEQ will select range nearest-neighbors as the basic components of surrounding nodes. Then, it will identify the gaps between surrounding nodes and try to select nearest neighbor collaborative nodes for enclosing the event in node selection phase, which can avoid redundant sensor nodes to join the surrounding nodes via identifying a set of association surrounding nodes between the nearest sensor nodes and query events. The main contributions of ICEQ may be summarized as follows.

(i) To retrieve a set of the nearest collaborative nodes of a specific event, ICEQ can identify a set of association surrounding nodes between the nearest sensor nodes and the query events that frequently appear in the system, which converts the demographic values and sensed data items presented in each query transaction into demographic types and event categories, respectively. Hence, ICEQ can select the nodes appropriately to decrease the number of selected nodes and prolong the lifetime of WSNs.

(ii) ICEQ is able to identify where gaps exit between surrounding nodes by finding large or frequent demographic query itemsets of query, and then try to select proper collaborative nodes for enclosing the event with rule decision and computing confidence between rules. Hence, ICEQ can select the appropriately nodes according to the network topology and environment.

The rest of this paper is organized as follows. The proposed intelligent collaborative event query algorithm is given in Section 2. Performance studies are conducted in Section 3. This paper concludes with Section 4.

2. Proposed Algorithm

2.1. Problem Description. In a distributed WSN, assume that that each sensor node s_i has a unique identity (ID) and is aware of their locations via global positioning system (GPS) devices. Each sensor node s_i has a fixed communication range c_i and a fixed sensing range r_i. And the communication range of a sensor node s_i follows unit disk graph model. Therefore, a sensor node s_i can communicate with a sensor node s_j if they are in each others' communication range. Otherwise, the sensing range of a sensor node s_i is also a disk and smaller than its communication ranges generally. The deployment of sensor nodes is random (or grid) and dense enough over a two-dimensional monitoring region. Euclidean distance is used as a metric to measure the distance between nodes. The Euclidean distance between any two nodes s_i and s_j is denoted by $d(s_i, s_j)$. The goal of the proposed intelligent collaborative event query (ICEQ) algorithm is to appropriately select a set of nearest collaborative nodes of a specific event. When given a set of rough boundary nodes B_S which are near to a real event boundary, an approximate boundary of the event can be obtained to bound the event region. Hence, ICEQ is to find out a set S of the nearest sensor nodes to such area that sensing ranges of adjacent nodes in S must be overlapping to enclose the event. In other words, adjacent sensor nodes s_i, s_j in S must satisfy the condition

$$d\left(s_i, s_j\right) < r_i + r_j, \quad s_i \in S,\ s_j \in S, \tag{1}$$

where $d(s_i, s_j)$ is the Euclidean distance between adjacent nodes s_i and s_j; r_i and r_j are sensing ranges nodes; s_i and s_j individually.

In order to select the nodes appropriately, the proposed ICEQ algorithm will identify a set of association surrounding nodes between the nearest sensor nodes and the query events that frequently appear in the system, which will consider the spatial and the temporal correlation between sensors and environment. Suppose there are k demographic attributes with domains being D_i ($i \in [1, k]$). Let $B = \{b_1, b_2, \ldots, b_r\}$ be the set of sensed data items of sensor nodes. An aggregation hierarchy on the ith demographic attribute, denoted $H(D_i)$, is a tree with leaf nodes corresponding to the different D_i values and internal nodes representing groupings of D_i values. A taxonomy on B, denoted $H(B)$, is a tree with the set of leaves being equal to B, and internal nodes indicate sensor node categories. A link represents and is a relationship. To facilitate mining sensing data profile association rules, we group the aggregated sensor nodes of the same demographic information, resulting in a new type of query transaction called demographic query transaction. Specifically, the demographic query transaction of the ith query is represented as a tuple:

$$t_i = \langle d_{i,1}, d_{i,2}, \ldots, d_{i,k}, b_{i,1}, b_{i,2}, \ldots, b_{i,s} \rangle, \quad i \in [1, n], \quad k \geq 1, \quad s \geq 1, \tag{2}$$

where $d_{i,j}$ is a leaf in $H(D_j)$ that represents the jth demographic attribute value of the aggregated sensor nodes, and $b_{i,t}$ is a leaf in $H(B)$ that represents the sensed data items of sensor nodes that is the ith query.

Since the goal is to identify the associations between demographic types and event categories; the demographic values and sensed data items presented in each query transaction must be converted into demographic types and event categories, respectively, resulting in an extended query transaction. Here we include all demographic types of each demographic value and all sensed data categories of all item appeared in the sink node. Therefore, the ith query transaction can be translated to the extended query transaction:

$$t_i' = \left\langle d_{i,1}', d_{i,2}', \ldots, d_{i,u}', b_{i,1}', b_{i,2}', \ldots, b_{i,m}' \right\rangle, \quad i \in [1, n], \quad u \geq 1, \quad m \geq 1, \tag{3}$$

where $d_{i,j}'$ and $b_{i,j}'$ are internal nodes in $H(D_j)$ and $H(B)$, respectively. Note that a demographic type could be a conjunction of several primitive demographic types. We use $d = (d_1, d_2, \ldots, d_l)$ to denote a complex demographic type d' that is a conjunction of $d_1, d_2, \ldots, d_l$. We say that the query transaction t_i supports a demographic type $d' = (d_1, d_2, \ldots, d_l)$ if $\{d_1, d_2, \ldots, d_l\} \subset t_i'$, where t_i' is the extended query transaction of t_i.

Similarly, we say that t_i supports a query event category c if $c \in t_i'$. A generalized profile association rule is an implication of the form $X \rightarrow Y$, where X is a demographic type and Y is a query event category. The rule $X \rightarrow Y$ holds in the query transaction set T with a confidence c% if c percent of the query transactions in T supports both X and Y. The rule $X \rightarrow Y$ also has support s% in the query transaction set T if s percent of the query transactions in T supports both X and Y. Therefore, given a set of query transactions T and several demographic aggregation hierarchies $H(D_j)$, $j \in [1, k]$, and one sensor taxonomy

$H(B)$, the problem of mining generalized profile association rules from query transaction data is to discover all rules that have support and confidence greater than the query-specified minimum support called Min_{sup} and minimum confidence called Min_{conf}. These rules are named strong rules.

2.2. Intelligent Collaborative Event Query Algorithm. The proposed ICEQ algorithm consists of two phases: initial phase and node selection phase. ICEQ will select range nearest neighbors as the basic components of surrounding nodes in initial phase. Node selection phase is to identify gaps between the rough surrounding nodes and then try to select proper surrounding nodes for monitoring the event by intelligent collaborative processing among nodes to decrease power consumption.

In the initial processing step, let Q denote a priority queue, let S_{E_i} denote a randomly selected end node, let $S_{E_i,\text{NN}}$ denote the nearest neighbour node of S_{E_i}, let L_i denote a query line, let S_{S,L_i} be the start node of a query line L_i, let $S_{S,L_i,\text{NN}}$ be the nearest neighbour node of S_{S,L_i}, let S_{E,L_i} be the end node of a query line L_i, let $S_{E,L_i,\text{NN}}$ be the nearest neighbour node of S_{E,L_i}, let A_{E,L_i} be the end node set of a query line L_i that it can divide L_i into several subsegments through these end nodes, let $A_{E,O}$ be a set of the end nodes covered by the spatial object O, let S_{E_i} denote a randomly selected end node, let D_{MAX} be the maximum distance, and let S be the results of event query. The initialization values of the parameters are as follows:

$$\begin{aligned} d(S_{E_i}, S_{E_i,\text{NN}}) &\longleftarrow \infty, \\ S_{S,L_i,\text{NN}} &\longleftarrow \text{NULL}, \\ S_{E,L_i,\text{NN}} &\longleftarrow \text{NULL}, \\ S &\longleftarrow \text{NULL}, \\ A_{E,O} &\longleftarrow \text{NULL}, \end{aligned} \tag{4}$$

where $d(S_{E_i}, S_{E_i,\text{NN}})$ is the distance between S_{E_i} and $S_{E_i,\text{NN}}$; Q is initialized with root node.

In order to differentiate different segments of corresponding query-line nearest-neighbor nodes, end nodes of subsegments of a specified query line L_i are obtained by doing the intersection between a perpendicular bisector of current scanned nodes, neighboring LNN nodes, and L_i. Thus, for each endpoint S_{E_i} which belong to the query line L_i, all points of L in $[S_{E_i}, S_{E_{i+1}}]$ have the same nearest neighbor node defined as $S_{E_i,\text{NN}}$. It is possible that sensor nodes scanned later are much closer than sensor nodes for certain subsegments in the nearest neighbor node list. Therefore, it needs to check whether this sensor node covers some endpoints which are obtained by nodes previously scanned. If there is a currently scanned sensor s_j whose distance $d(s_j, S_{E_i})$ is smaller than $d(S_{E_i}, S_{E_i,\text{NN}})$ for some S_{E_i}, it means the end node S_{E_i} is covered by s_j. Since there are endpoints of subsegments obtained from intersection of the perpendicular bisector and the specified query line, the currently scanned sensor s_j is the nearest neighbor node. The algorithm proposed removes the end node S_{E_i}, adds new end nodes S'_{E_i} by s_j, and updates $S'_{E_i,\text{NN}}$ accordingly. Also, a threshold D_{MAX} which determines the number of surrounding node candidates visited needs to be updated as maximum $d(S_{E_k}, S_{E_k,\text{NN}})$ of the current nearest neighbour list. Finally, we prepare a queue S to gather the results of the nearest neighbour lists and sort them counterclockwise with reference to the center of the approximate polygonal boundary of the event. These will be parts of the selection of our surrounding nodes of the event. And the proposed ICEQ algorithm is shown in Algorithm 1.

2.3. Collaborative Node Selection. The goal of collaborative node selection phase is to select proper sensors to enclose the event. From Algorithm 1, we can see that the rough nearest surrounding nodes of the event have been put in the selected nodes set S. Then, we construct neighbourhood relationships for each sensor node s_i in S first and find out the final collaborative nodes to monitor the event.

If we sort nodes in the queue S counter clockwise with reference to the centre point of the approximate polygonal boundary of the event, a sensor node s_i indexed i in the queue S sets its left-hand side neighbour as a sensor node indexed $i+1$ and its right-hand side neighbour as a sensor node indexed $i-1$ in the queue S.

In the phase, each sensor node s_i needs to keep information of their one-hop neighbors to construct neighborhood relationship. Each sensor node s_i will store its neighbors within communication range in its adjacency list. Because we only have partial results of nearest surrounding nodes of the event and these sensors are too few to enclose the event, there may be gaps between adjacent nodes with respect to their sensing ranges.

The proposed ICEQ algorithm will check whether gaps exit between this node and its adjacent neighbors in S. When a sensor node s_i ($s_i \in S$) is accessed, it first checks the distance $d(s_i, s_{i,\text{LH}})$ between s_i and its left-hand side neighbor $s_{i,\text{LH}}$. If $d(s_i, s_{i,\text{LH}})$ satisfies

$$d(s_i, s_{i,\text{LH}}) > r_{s_i} + r_{s_{i,\text{LH}}}, \tag{5}$$

which means that there is a gap between them.

A neighbor node s_k in adjacency lists selected as surrounding nodes must satisfy one of the following condition that:

$$\text{Min}\left(d(s_i, s_k) + d(s_k, s_{i,\text{LH}})\right), \tag{6}$$

$$\text{Min}\left(d(s_i, s_k) + d(s_k, s_{i,\text{RH}})\right). \tag{7}$$

We also construct neighborhood relationship for s_k as to two neighbors, s_i, and $s_{i,\text{LH}}$. Then, s_k will be inserted at the end of the queue S. Similarly, it will also check whether there is a gap between s_i and s_i's right-hand side neighbor, $s_{i,\text{RH}}$, and select proper s_k to enclose it. This process will continue until all elements in S have been checked.

In order to select the appropriate node, we need to identify generalized profile association rules in the rough node set S, the itemsets that will interest us are of the following form $\langle d_{i_1}, d_{i_2}, \ldots, d_{i_l}, b\rangle$, where d_{i_l} is an internal node in $H(D_{i_j})$ and b is an internal node in $H(B)$. By finding

Input: Rough event boundary B_E, and root node S_R.
Output: Selected nodes set S.
1: Initialization (4).
2: **for** each query line L_i of B_E **do**
3: Dequeue node S_j from Q;
4: **if** $(\min d(S_j, L_i) < D_{\max}$ **and** $S_{S,L_i,\mathrm{NN}}$ and $S_{E,L_i,\mathrm{NN}}$ are NULL) **then**
5: $S_{S,L_i,\mathrm{NN}} \leftarrow S_j, S_{E,L_i,\mathrm{NN}} \leftarrow S_j, D_{\max} \leftarrow \min d(S_j, L_i)$;
6: **else**
7: Add an end node S'_{E_i} into event query line set A_{Q,L_i};
8: Update $d(S'_{E_i}, S'_{E_i,\mathrm{NN}})$;
9: Add an end node S'_{E_i+1} into event query line set A_{Q,L_i};
10: $A_{E,S_j} \leftarrow \emptyset$;
11: **end if**
12: $S \leftarrow A_{Q,L_i}$;
13: **end for**
14: Collaborative node selection;
15: **return** S;

Algorithm 1: Intelligent collaborative event query algorithm.

large or frequent demographic query itemsets, we can easily derive the corresponding generalized profile association rules.

Let F_k denote the frequent itemsets of the form $\langle d_{i_1}, d_{i_2}, \ldots, d_{i_l}, b\rangle$. A candidate itemset C_{k+1} is generated by joining F_k and F_{k-1}, except that the k join attributes must include on query event type b, and the other $k-1$ demographic attributes types from $d_{i_1}, d_{i_2}, \ldots, d_{i_l}$. We first extend each query transaction t_i as expressed in (1). The set of extended query transactions is denoted ET. After scanning the data set ET, we obtain large demographic 1-itemsets $L_1(D)$ and large event 1-itemsets $L_1(B)$. If an item is not a member of $L_1(D)$ or $L_1(B)$, it will not appear in any large demographic query itemset and is, therefore, useless. We delete all the useless items in every query transaction of ET in order to reduce its size. The set C_1 of candidate 1-itemsets is defined as $L_1(D) \times L_1(B)$. Data set ET is scanned again to find the set L_1 of large demographic query 1-itemsets from C_1. A subsequent pass, say pass k, is composed of two steps. First, we use the above-mentioned candidate generation function to generate the set C_k of candidate itemsets by joining two large $(k-1)$ itemsets in F_{k-1} on the basis of their common $k-2$ demographic attribute values and the query attribute value. Next, data set ET is scanned and the support of candidates in C_k is counted. The set F_k of large k-itemsets are itemsets in C_k with minimum support.

Considering that some of the strong generalized profile association rules could be related to each other in either the demographic itemset part or the sensor nodes, and, therefore, the existence of one such rule could makes some others not interesting. To overcome the problem, let Π be the set of all demographic attribute types:

$$\Pi = \bigcup_{i=1}^{k} H(D_i). \tag{8}$$

We call a rule R_1:

$$R_1 : D' \longrightarrow b_1, \quad D' \subseteq \Pi,\ b_1 \in B. \tag{9}$$

a D-ancestor of another rule R_2:

$$R_2 : D'' \longrightarrow b_2, \quad D'' \subseteq \Pi,\ b_2 \in P, \tag{10}$$

if $b_1 = b_2$, and for all $d_1 \in D'$, there exists $d_2 \in D''$, such that d_1 is equal to or an ancestor of d_2 in the associated demographic concept hierarchy. Similarly, we call a rule $R_1 : D' \rightarrow b_1$ a P-ancestor of another rule $R_2 : D'' \rightarrow b_2$ if and is equal to or an ancestor of b_2 in the node taxonomy. Also, R_2 is called a D-descendant of R_1 if R_1 is a D-ancestor of R_2. Note that D-descendant and B-descendant together form a lattice on the generalized profile association rules.

In context of collaborative nodes selection, we say a rule is valid if it can be used for making decision. Given a set of strong rules say Ω the candidate of a generalized profile association rule $R : D \rightarrow b \in \Omega$ is the confidence of the rule:

$$D - \bigcup_{i=1}^{l} D^i \longrightarrow b - \bigcup_{i=1}^{l'} b_i \in \Omega, \tag{11}$$

where $R_i : D^i \rightarrow b$, $i \in [1, l]$ are the immediate D-descendants of R in Ω, and $R_i : D \rightarrow b_i$, $i \in [1, l']$ are the immediate B-descendants of R in Ω. Also, we say R is valid if the candidate of R is no less than $\mathrm{Min}_{\mathrm{conf}}$. From (11), we can see that the rule $R : D \rightarrow b$ is consulted only when we need to decide whether to select a node in $b - \bigcup_{i=1}^{l'} b_i$ to a query event with demographic type in $D - \bigcup_{i=1}^{l} D^i$. However, the difficulties of identifying candidate of a strong rule lie in computing the confidences of its DB-deductive rule.

Let the immediate D-descendants of R be $R_i : D^i \rightarrow b$, $i \in [1, l]$; $D - \bigcup_{i=1}^{l} D^i \rightarrow b$ let be called the D-deductive rule of R.

Let the immediate B-descendants of R be $R_i : D \rightarrow b_i$, $i \in [1, l']$, and $D \rightarrow b - \bigcup_{i=1}^{l'} b_i$ let be called the B-deductive rule of R.

Suppose that we have obtained the confidences of both the D-deductive rule and the B-deductive rule of a given rule R. Let E-Conf(DB-deductive rule|D-deductive rule) be the

estimated confidence of R, DB-deductive rule given R; let D-deductive rule and E-Conf(DB-deductive rule|B-deductive rule) be the estimated confidence of R's DB-deductive rule given R, B-deductive rule. We have

$$\begin{aligned}&\text{Conf(DB-deductive)}\\&\quad\le E\text{\_Conf(DB-deductive rule} \mid \text{D-deductive rule)}\\&\quad= \text{Conf}\left(D-\bigcup_{i=1}^{l}D^i\right)\times\frac{\left|b-\bigcup_{i=1}^{l'}b_i\right|}{|B|},\\&\text{Conf(DB-deductive)}\\&\quad\le E\text{\_Conf(DB-deductive rule} \mid \text{B-deductive rule)}\\&\quad= \text{Conf}\left(D\longrightarrow b-\bigcup_{i=1}^{l'}b_i\right).\end{aligned}\tag{12}$$

Therefore, we define the estimated interestingness of R, denoted E_Interest(R), which is the estimated confidence of R's DB-deductive rule, to be

$$\begin{aligned}&E\text{\_Interest}(R)\\&=\text{Min}\left\{\text{Conf}\left(D-\bigcup_{i=1}^{l}D^i\longrightarrow b\right)\right.\\&\qquad\left.\times\frac{\left|b-\bigcup_{i=1}^{l'}b_i\right|}{|b|},\text{Conf}\left(D\longrightarrow b-\bigcup_{i=1}^{l'}b_i\right)\right\}.\end{aligned}\tag{13}$$

We approximate the confidence of a D-deductive rule by using the following theoretic results.

Lemma 1. *Let D^i be mutually disjoint, and the confidence of $D-\bigcup_{i=1}^{l}D^i\rightarrow b$ be*

$$\begin{aligned}&\text{Conf}\left(D-\bigcup_{i=1}^{l}D^i\longrightarrow b\right)\\&\quad=\frac{\sup(D,b)-\sum_{i=1}^{l}\sup(D^i,b)}{\sup(D)-\sum_{i=1}^{l}\sup(D^i)}\\&\quad=\frac{\sup(D,b)-\left(\sup(D',b)+\cdots+\sup\left(D^l,b\right)\right)}{\sup(D)-\left(\sup(D')+\cdots+\sup(D^l)\right)}.\end{aligned}\tag{14}$$

Proof. Since D^i $(i\in[1,l])$ are disjoint, $\sup(\bigcup_{i=1}^{l}D)=\sum_{i=1}^{l}\sup(D^i)$.

Similarly, $\sup(\bigcup_{i=1}^{l}D,b)=\sum_{i=1}^{l}\sup(D^i,b)$.

Therefore,

$$\begin{aligned}&\text{Conf}\left(D-\bigcup_{i=1}^{l}D^i\longrightarrow b\right)\\&\quad=\frac{\sup(D,b)-\sum_{i=1}^{l}\sup(D^i,b)}{\sup(D)-\sum_{i=1}^{l}\sup(D^i)}\\&\quad=\frac{\sup(D,b)-\left(\sup(D',b)+\cdots+\sup\left(D^l,b\right)\right)}{\sup(D)-\left(\sup(D')+\cdots+\sup(D^l)\right)}.\end{aligned}\tag{15}$$

□

Theorem 2. *Without loss of generality, let D^i $(i\in[1,l])$ be mutually disjoint. Assume that* $\text{Conf}(\sum_{j=i+1}^{l}D^j-\sum_{j=1}^{i}D^j\rightarrow b)\ge\text{Min}_{\text{conf}}$. *If $D-\bigcup_{i=1}^{l}D^i\rightarrow b$ has sufficient confidence, we have*

$$\frac{\sup(D,b)-\left(\sup(D',b)+\cdots+\sup\left(D^l,b\right)\right)}{\sup(D)-\left(\sup(D')+\cdots+\sup(D^l)\right)}\ge\text{Min}_{\text{conf}}.\tag{16}$$

Proof. Let $D_1=\sum_{j=1}^{i}D^j$, and let $D_2=\sum_{j=1+i}^{l}D^j-D_1$. Obviously, D_1 and D_2 are disjoint. Since both $D_2\rightarrow b$ and $D-(D_1\cup D_2)\rightarrow b$ have sufficient confidences, we have

$$\frac{\sup(D_2,b)}{\sup(D_2)}\ge\text{Min}_{\text{conf}},\tag{17}$$

and, by Lemma 1,

$$\frac{\sup(D,b)-\sup(D_1,b)-\sup(D_2,b)}{\sup(D)-\sup(D_1)-\sup(D_2)}\ge\text{Min}_{\text{conf}}.\tag{18}$$

By adding the denominators and numerators, respectively, from the left-hand sides of the two equations, we can obtain

$$\frac{\sup(D,b)-\sup(D_1,b)}{\sup(D)-\sup(D_1)}\ge\text{Min}_{\text{conf}}.\tag{19}$$

Since D^i $(i\in[1,l])$ are mutually disjoint, we have

$$\frac{\sup(D,b)-\left(\sup(D',b)+\cdots+\sup\left(D^l,b\right)\right)}{\sup(D)-\left(\sup(D')+\cdots+\sup(D^l)\right)}\ge\text{Min}_{\text{conf}}.\tag{20}$$

□

Now we discuss how to compute the confidence of a B-deductive rule $R_B: D\rightarrow b-\bigcup_{i=1}^{l'}b_i$. The query transactions that support R_B must have included nodes that fall outside $\bigcup_{i=1}^{l'}b_i$. We say node categories b_i and b_j are siblings if they have a common parent in the respective concept hierarchy. Let $\text{NST}(D,b_i)$ denote the set of query transactions that support (D,b_i) but do not support any sibling of R_B. To calculate $\text{NST}(D,b_i)$, we associate a flag f_{NST} on each node category of an extended query transaction. NS(b, et), where b is a query category and et is an extended query transaction,

Input: A queue S which contains RNN results.
Output: A set S of selected collaborative nodes.
1: **for** each s_i ($s_i \in S$) **do**
2: Find out strong rules Ω;
3: Calculate sufficient confidence with (14) and (16);
4: **if** (sufficient confidence of $s_i \geq \text{Min}_{\text{conf}}$) **then**
5: **if** ($d(s_i, s_{i,\text{LH}}) > r_{s_i} + r_{s_{i,\text{LH}}}$) **then**
6: Choose s_k from s_i and $s_{i,\text{LH}}$ adjacency list with (6);
Construct neighbourhood relationship for s_k;
Insert s_k at the end of S;
7: **else if** ($d(s_i, s_{i,\text{RH}}) > r_{s_i} + r_{s_{i,\text{RH}}}$) **then**
8: Choose s_k from s_i and $s_{i,\text{RH}}$ adjacency list with (7);
Construct neighbourhood relationship for s_k;
Insert s_k at the end of S;
9: **end if**
10: **end if**
11: **end for**

Algorithm 2: Collaborative node selection.

is equal to 1 if there exists no sibling of b in et and 0 otherwise. Therefore, we have

$$\text{NSSup}(D, b_i) = \frac{\sum_{\text{et supports}(D,b_i)} \text{NS}(b_i, \text{et})}{n}, \quad (21)$$

where n is the total number of query transactions.

If $r : D \rightarrow b - \bigcup_{i=1}^{l'} b_i$ has sufficient confidence, we can get

$$\frac{\sup(D, b) - \sum_{i=1}^{l} \text{NSSup}(D, b_i)}{\sup(D)} \geq \text{Min}_{\text{conf}}. \quad (22)$$

$\text{NSSup}(D, b_i)$ for a demographic node itemset (D, b_i) can be computed when counting the support for (D, b_i) by expression (21). Expression (22) shows that $\sup(D, b) - \sum_{i=1}^{l} \text{NSSup}(D, b_i)/\sup(D)$ is an upper bound of the confidence of $D \rightarrow b - \bigcup_{i=1}^{l'} b_i$. Therefore, if the upper bound is less then Min_{conf}, we drop the rule because it cannot have sufficient confidence, and consequently R is considered not interesting.

At last, we report all sensor nodes in S as nearest surrounding nodes of the event. And the proposed collaborative node selection algorithm is shown in Algorithm 2.

3. Simulation Results

3.1. Simulation Setup. In order to evaluate the performance of the proposed ICEQ algorithm, we implemented the ICEQ in the well-known simulation tool NS-2 [30]; the range nearest neighbor (RNN) query algorithm [20] is simulated as discussed here. There are 5000 sensor nodes deployed in our monitoring region. The shape of the event that occurs in the monitoring region is a circle. The approximate polygonal boundary of the event can be obtained via the boundary nodes of the event. Thus, the deployment strategy totally ensures the assumption that there is no communication hole in the network. And the system generates critical and noncritical events randomly. The performance analysis has been done by deploying variable number of nodes on the fixed squared area, to verify the effect of node densities on the data gathering path length and average number of hop counts. The deployment of sensor nodes is dense enough so that we can find out surrounding nodes of the event. And we set the Min_{conf} as 0.6. Two kinds of deployment, grid distribution and random distribution, are applied individually for comparison. There are three metrics used to compare the performance of proposed methods which are described as follows: query processing time, selected numbers of collaborative nodes, and total message consumption. Our simulation results are all from the average of 1000 runs.

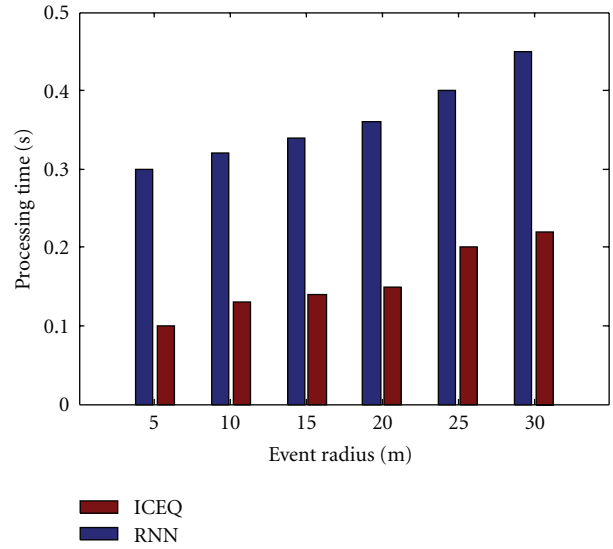

Figure 1: Processing time with varying event radius in grid topology.

3.2. Validation in Different Range of Event. In the first scenario, we vary the size of the event with varying its radius from 5 m to 30 m. Figures 1 and 2 show the query-processing time and average number of selected nodes of different algorithms, respectively.

As shown in Figure 1, it is observed that the proposed ICEQ outperforms in terms of query-processing time irrespective of the event radius. For ICEQ, the query processing is less than 0.23 s when the event radius, increases, while for RNN, the query processing time is higher than 0.3 s. The reason is that RNN needs to search RNNs edge by edge according to the approximate polygonal boundary of the event, and the cost of RNN is essentially higher than ICEQ. It is noticed that the event radius has some impact on the query-processing time for both ICEQ and RNN. It is reasonable since larger event radius means that more nodes will be evaluated in node selection. So when we increase the size of the event, the cost to find out surrounding nodes of the event will raise accordingly.

Figure 2 shows the numbers of selected nodes for the monitoring event. With the number of event radius increasing, the number of selected nodes of two schemes will

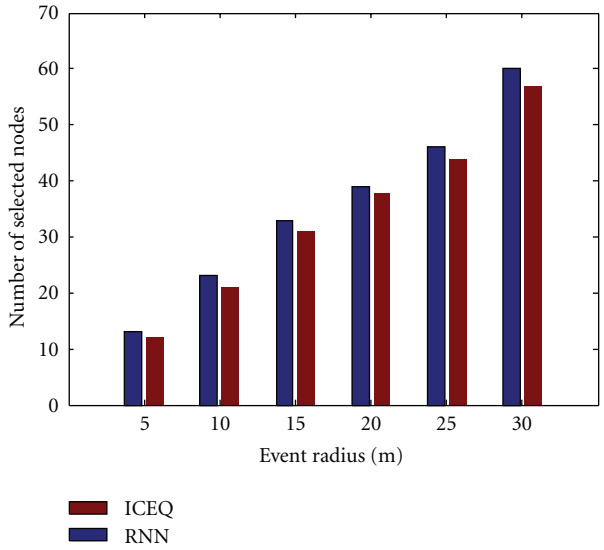

Figure 2: Number of selected nodes with varying event radius in grid topology.

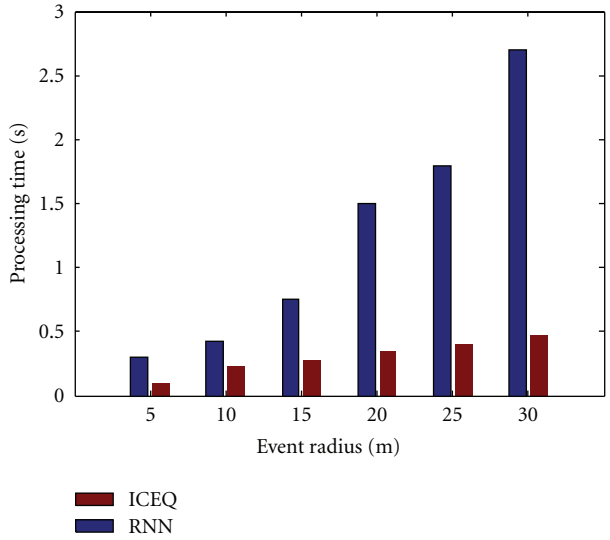

Figure 3: Processing time with varying event radius in random topology.

increase obviously. When the event radius increases from 5 m to 30 m, the number of selected nodes of RNN increases to 61, which is slightly larger than that of ICEQ. We can see that the number of selected nodes of ICEQ always is slightly less than that of ICEQ. The reason is that the objective of the ICEQ is to minimize the selected nodes for the given constrained conditions. So in collaborative node selection phase, ICEQ considers the sufficient confidence and rules between nodes, which will avoid redundant sensor nodes to join surrounding nodes via identifying a set of association surrounding nodes between nearest sensor nodes and query events.

Figures 3 and 4 show the query-processing time and average number of selected nodes of different algorithms in random topology, respectively.

From Figure 3, we can see the similar results as in Figure 1. And the proposed ICEQ outperforms in terms of query-processing time irrespective of the event radius. For ICEQ, the query processing is less than 0.5 s, and it will increase slightly when the event radius increases. While for RNN, the query-processing time is obviously higher than that of ICEQ. When the event radius increases, the query-processing time of RNN will increase suddenly. And the query-processing time will larger than 2.5 s. It is noticed that the event radius has great impact on the query-processing time for RNN. The reason is the RNN needs to search RNNs edges, which depends on the network topology. Hence, the cost of RNN is essentially higher than ICEQ. For ICEQ, the reason of processing time increasing slightly is that ICEQ will visit more candidates with the radius increasing.

The results of number of selected nodes of different algorithms with the varying event radius are shown in Figure 4. As the event radius increases, the number of selected nodes of different algorithms will increase obviously. Especially for RNN, the number of selected nodes increases to 57. The main reason is that RNN selects surrounding

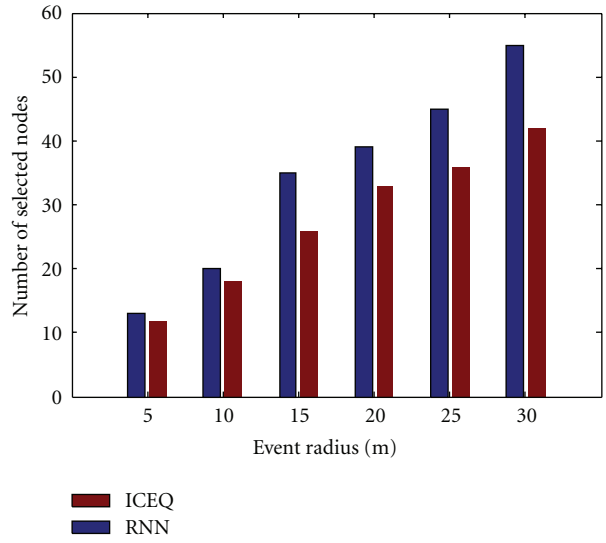

Figure 4: Number of selected nodes with varying event radius in random topology.

nodes by searching according to the approximate polygonal boundary of the event, which leads to select more nodes. So the event radius has a great influence on RNN. The event detection ratio of RNN is higher than that of ICEQ clearly. Because ICEQ can identify the gaps between surrounding nodes and try to select nearest neighbor collaborative nodes for enclosing the event in node selection phase, which can avoid redundant sensor nodes to join surrounding nodes via identifying a set of association surrounding nodes between the nearest sensor nodes and the query events. For the proposed ICEQ, the number of selected nodes is less than that of ICEQ. For example, when the event radius increases to 30m, the number of selected nodes of ICEQ increases to 42. Compared with RNN, ICEQ tries to select proper collaborative nodes for enclosing the event with rule

decision and computing confidence between rules. And the collaborative node selection scheme of ICEQ also helps to decrease the number of selected nodes.

3.3. Validation in Different Query Lines. In this scenario, we vary query lines in random topology. And sensor nodes are deployed in random topology. The radius of the circular event is fixed at 15 m, and we assume that it occurs arbitrarily in the monitoring region so that we obtain different number of edges of the approximate polygonal boundary. Figures 5 and 6 show the query-processing time and the average number of selected nodes of different algorithms with varying query lines in random topology, respectively.

The results of query-processing time of different algorithms with the varying number of query lines are shown in Figure 5. As the number of query lines increases, the query-processing time of different algorithms will increase. Especially for RNN, the query-processing time increases to 0.98 s when the number of query lines increases to 15. The main reason is that RNN needs to search edges according to the approximate polygonal boundary of the event, which leads to longer delay time and increases query-processing time. So the number of query lines has a great influence on RNN. For the ICEQ, the query-processing time is obviously lower than that of RNN. When the number of query lines increases to 15, the query processing time of ICEQ only increases to 0.19 s, which is less than that of RNN 0.79 s. Another phenomenon is that the number of query lines has no much effect on the ICEQ algorithm. The query-processing time almost keeps in the range of 0.15 s to 0.2 s. The reason is that ICEQ will determine the number of surrounding node candidate to traverse in the search process. Therefore, ICEQ can adaptively select appropriate collaborative nodes to decide the number of surrounding node candidates to form the approximate polygonal boundary. We also notice that the advantage of ICEQ over RNN becomes more evident when the number of query lines becomes large.

Figure 6 shows the number of selected nodes by different schemes when the number of query lines varies from 10 to 15. It can be seen that the number of selected nodes of two schemes oscillates slightly as the number of query lines increases. For RNN, the average number of selected nodes increases slightly, from 20 (10 query lines) to 23 (15 query lines). The reason is that RNN must find out the edge and the search surrounding nodes, which will increase the selected nodes when query lines increas. However, ICEQ always outperforms RNN due to the concurrent use of the rule mining among nodes and collaborative nodes. For instance, it achieves average 18.4% nodes saving compared with RNN scheme. Less selected nodes lead to longer network lifetime since sensors can turn to the power-saving mode once the event monitoring in their region is done.

3.4. Validation in Query Consumption. In this scenario, we investigate the total message consumption with varying network size from 100 to 10000 sensor nodes. Figures 7 and 8 show the query total message consumption by varying network size from 100 to 10000 sensor nodes. The duration of each query is set by 300 s and number of event types.

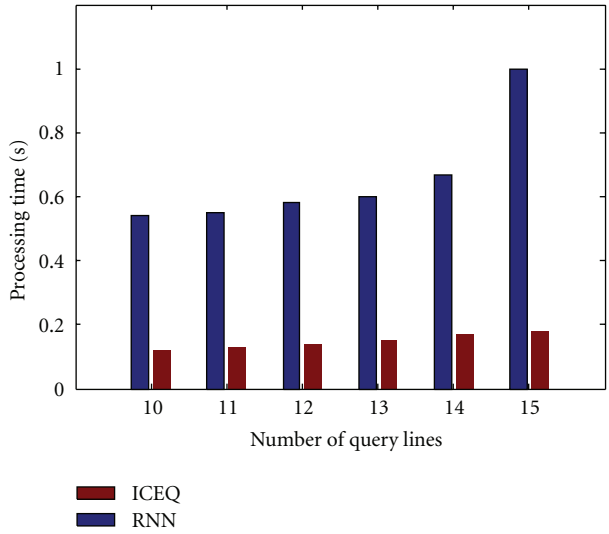

Figure 5: Processing time with varying query lines in random topology.

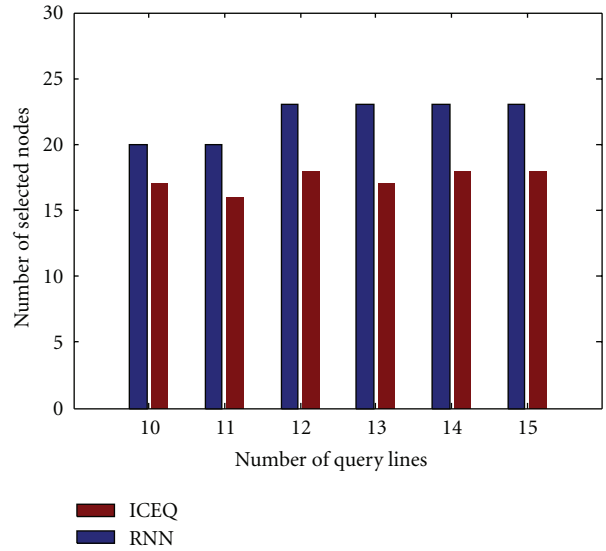

Figure 6: Number of selected nodes with varying query lines in random topology.

In Figure 7, the total number of messages of two mechanisms increases with the network size. The network size impacts on a number of messages significantly in RNN because the number of sensor nodes selected increases as the network size grows. In ICEQ, only a few numbers of reply messages exist so that the network size impact ICEQ a little due to the increasing network size. As the network size is small, the total number of messages of ICEQ is smaller than RNN. This is because overhead rose from the change of rough event boundary node larger than the benefit obtained from the nodes due to small network size. That is because network size is large enough, and the benefit of surrounding nodes is cost effective. Hence, ICEQ works efficiently in a large-scale WSN. Moreover, in ICEQ, the

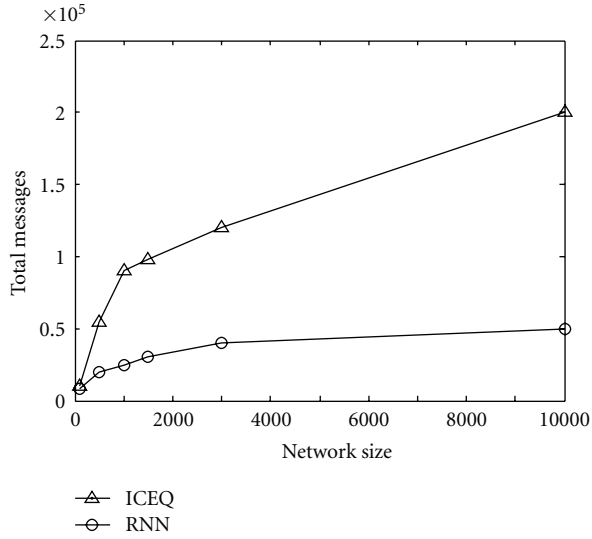

Figure 7: Total messages with varying network size.

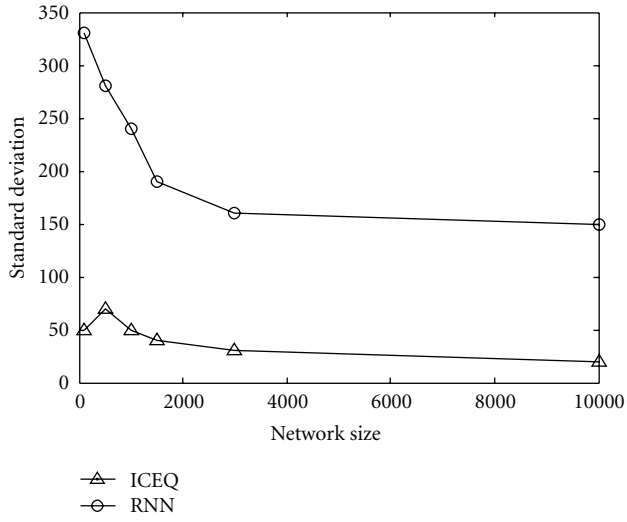

Figure 8: Standard deviation with varying network size.

involvement reduces the duplicated data packets, making that ICEQ outperforms RNN in a large-scale network.

Figure 8 compares the degree of power balance of all mechanisms in various network sizes. Each sensor node has the same probability for detecting event data, and the network traffic is uniformly distributed over the whole WSN. When the network size is smaller than 500, standard deviation of RNN is small. This is because that the network size is too small, and the sensor nodes intend to select rough event boundary nodes; therefore, the difference of query event on nodes is small. However, as the network size increases, the RNN sensor nodes that are close to event boundary have higher traffic than nodes in other location, and; therefore, their standard deviations is higher than ones of ICEQ mechanisms as the network size is 300 nodes. However, as the network size increases larger than 400, the standard deviations of RNN decrease significantly because the number of detected event is fixed and the event detected is distributed in the whole WSN. The ICEQ mechanism obtains a smaller standard deviation as network size is larger than 500. The results also validate the effectiveness of the proposed ICEQ in power consumption.

4. Conclusion

In this paper, we presented an efficient intelligent collaborative event query (ICEQ) algorithm to detect the event early and provide monitoring information and event query timely in WSNs. ICEQ can identify a set of association surrounding nodes between the nearest sensor nodes and the query events that frequently appear in the system, which converts the demographic values, and sensed data items presented in each query transaction into demographic types and event categories, respectively. Hence, ICEQ can select the nodes appropriately to decrease the number of selected nodes and prolong the lifetime of WSNs. ICEQ is able to identify where gaps exit between surrounding nodes by finding large or frequent demographic query itemsets of query, and then try to select proper collaborative nodes for enclosing the event with rule decision and computing confidence between rules. Hence, ICEQ can select the appropriately nodes according to the network topology and environment. Through ICEQ, we can select a set of surrounding nodes of the event instead of all the sensor nodes in the monitoring region to check if there is any event evolution. Therefore, sensor nodes which are not surrounding nodes can enter into sleep modes temporarily to save their battery energies and thus extend the lifetime of sensor networks. The future work will focus on the issues of query moving objects and track objects in WSNs.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (no. 60902053), the Science and Technology Research Planning of Educational Commission of Hubei Province of China (no. B20110803), and the Natural Science Foundation of Hubei Province of China (no. 2008CDB339). The author also gratefully acknowledges the helpful comments and suggestions of the reviewers.

References

[1] Y. Rachlin, R. Negi, and P. K. Khosla, "The sensing capacity of sensor networks," *IEEE Transactions on Information Theory*, vol. 57, no. 3, pp. 1675–1691, 2011.

[2] C. Y. Chong and S. P. Kumar, "Sensor networks: evolution, opportunities, and challenges," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, 2003.

[3] H. Ying, M. Schlösser, A. Schnitzer et al., "Distributed intelligent sensor network for the rehabilitation of Parkinson' s patients," *IEEE Transactions on Information Technology in Biomedicine*, vol. 15, no. 2, pp. 268–276, 2011.

[4] O. Chipara, C. Lu, J. A. Stankovic, and G.-C. Roman, "Dynamic conflict-free transmission scheduling for sensor network queries," *IEEE Transactions on Mobile Computing*, vol. 10, no. 5, pp. 734–748, 2011.

[5] W. Yu, T. N. Le, J. Lee, and D. Xuan, "Effective query aggregation for data services in sensor networks," *Computer Communications*, vol. 29, no. 18, pp. 3733–3744, 2006.

[6] G. He, R. Zheng, I. Gupta, and L. Sha, "A framework for time indexing in sensor networks," *ACM Transactions on Sensor Networks*, vol. 1, no. 1, pp. 101–133, 2005.

[7] G. S. Iwerks, H. Samet, and K. P. Smith, "Maintenance of k-nn and spatial join queries on continuously moving points," *ACM Transactions on Database Systems*, vol. 31, no. 2, pp. 485–536, 2006.

[8] J. Gehrke and S. Madden, "Query Processing in Sensor Networks," *IEEE Pervasive Computing*, vol. 3, no. 1, pp. 46–55, 2004.

[9] R. Kannan, S. Sarangi, and S. S. Iyengar, "Sensor-centric energy-constrained reliable query routing for wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 64, no. 7, pp. 839–852, 2004.

[10] J. Guang and N. Silvia, "Towards spatial window queries over continuous phenomena in sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 4, pp. 559–571, 2008.

[11] I.-R. Chen, A. P. Speer, and M. Eltoweissy, "Adaptive fault-tolerant QoS control algorithms for maximizing system lifetime of query-based wireless sensor networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 2, pp. 161–176, 2011.

[12] M. Demirbas, X. Lu, and P. Singla, "An in-network querying framework for wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 8, pp. 1202–1215, 2009.

[13] O. Chipara, C. Lu, J. A. Stankovic, and G.-C. Roman, "Dynamic conflict-free transmission scheduling for sensor network queries," *IEEE Transactions on Mobile Computing*, vol. 10, no. 5, pp. 734–748, 2011.

[14] R. Zhu, Y. Qin, and J. Wang, "Energy-aware distributed intelligent date gathering algorithm in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2011, Article ID 235724, 13 pages, 2011.

[15] B. Han, J. Leblet, and G. Simon, "Query range problem in wireless sensor networks," *IEEE Communications Letters*, vol. 13, no. 1, pp. 55–57, 2009.

[16] E. Cayirci, V. Coskun, and C. Cimen, "Querying sensor networks by using dynamic task sets," *Computer Networks*, vol. 50, no. 7, pp. 938–952, 2006.

[17] V. Stoumpos, A. Deligiannakis, Y. Kotidis, and A. Delis, "Processing event-monitoring queries in sensor networks," in *Proceedings of the 24th International Conference on Data Engineering (ICDE '08)*, pp. 1436–1438, April 2008.

[18] C. Ai, L. Guo, Z. Cai, and Y. Li, "Processing area queries in wireless sensor networks," in *Proceedings of the 5th International Conference on Mobile Ad-hoc and Sensor Networks (MSN '09)*, pp. 1–8, December 2009.

[19] M. Bestehorn, K. Böhm, E. Buchmann, and S. Kessler, "Energy-efficient processing of spatio-temporal queries in wireless sensor networks," in *Proceedings of the 18th ACM International Conference on Advances in Geographic Information Systems, (ACM SIGSPATIAL GIS '10)*, pp. 340–349, November 2010.

[20] H. Hu and D. L. Lee, "Range nearest-neighbor query," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 1, pp. 78–91, 2006.

[21] K. C. K. Lee, W.-C. Lee, and H. V. Leong, "Nearest surrounder queries," in *22nd International Conference on Data Engineering (ICDE '06)*, pp. 85–95, April 2006.

[22] Y. Yao and J. Gehrke, "The cougar approach to in-network query processing in sensor networks," *Sigmod Record*, vol. 31, no. 3, pp. 9–18, 2002.

[23] Y. Xu, W.-C. Lee, J. Xu, and G. Mitchell, "Processing window queries in wireless sensor networks," in *Proceedings of the 22nd International Conference on Data Engineering (ICDE '06)*, pp. 658–675, April 2006.

[24] M. Ding, D. Chen, K. Xing, and X. Cheng, "Localized fault-tolerant event boundary detection in sensor networks," in *Proceedings of the 24th IEEE International Conference of Computer and Communications Society (INFOCOM '05)*, vol. 2, pp. 902–913, 2005.

[25] J. Guang and S. Nittel, "NED: an efficient noise-tolerant event and event boundary detection algorithm in wireless sensor networks," in *Proceedings of the 7th International Conference on Mobile Data Management (MDM '06)*, pp. 151–153, May 2006.

[26] D. Zhou and J. Gao, "Opportunistic processing and query of motion trajectories in wireless sensor networks," in *Proceedings of the 28th Conference on Computer Communications (INFOCOM '09)*, pp. 1197–1205, April 2009.

[27] Y. He, M. Li, Y. Liu, J. Zhao, W. L. Huang, and J. Ma, "Collaborative query processing among heterogeneous sensor networks," in *Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '08)*, pp. 25–30, May 2008.

[28] T. W. Chiu and Q. Luo, "Collaboratively querying sensor networks through handheld devices," in *Proceedings of the IEEE International Conference on Mobile Data Management, (MDM '07)*, pp. 30–35, May 2007.

[29] R. Zhu, "Efficient fault-tolerant event query algorithm in distributed wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2010, Article ID 593849, 7 pages, 2010.

[30] The Network Simulator. NS-2, http://www.isi.edu/nsnam/ns/.