An Aggregation Tree Approach for Event Detection in Wireless Sensor Networks

ZHANG Shu-Kui * [1,2], GONG Sheng-Rong[1] ,CUI Zhi-Ming[1,2] , LIU Quan[1] ,FAN Jian-Xi[1]
[1]School of Computer Science and Technology, Soochow University, Suzhou, China, 215006
[2] JiangSu Province Support Software Engineering R&D Center for Modern Information Technology Application in Enterprise, Suzhou, China, 215104
Email: {zhangsk, shrgong, szzmcui,liuquan ,jxfan }@suda.edu.cn

Abstract-**Collecting data with confidence of the environment is an important application of the wireless sensor networks (WSN). In this study, we used a splay tree based event region fault-tolerant detection algorithm (STERD) for WSNs to address these problems. Rather than passing large quantities of raw data through the network, our network instead sends the coefficients of a regression polynomial that simulates the readings of the sensor nodes throughout the region of interest. These coefficients essentially summarize the relevant information over the region, in effect compressing the data, leading to a reduction in both data volume and energy expenditure. Simulation results indicate that the proposed STERD can attractively obtain the high event region detection accuracy and considerably reduce the false alarm probability.**

Keywords: **wireless sensor network, splay tree, in-network process, data aggregation, fault-tolerant detection .**

I. INTRODUCTION

One of the critical tasks in designing a wireless sensor network (WSN) is to monitor and report various useful occurrences of events in the network domain. An event can be defined as an exceptional change in environmental parameters such as temperature, pressure, humidity etc. However, an event may occur in many ways. When a particular sensor depicts a smooth variation over time, then the sensors are said to be spatio-temporally correlated just as the attributes[1],[2],[3],[4]. In accordance with different scenarios, it is necessary to exploit spatio-temporal characteristics of sensors to detect the emergence of event boundary accurately (eliminating faulty readings) and quickly convey this information to the sink node. Reporting the boundary of the event accurately is a challenging task as it may involve faulty readings from some sensors, which may affect the accuracy of the detected area of the event [5],[6],[7],[8],[9].

Many applications in which the sensor readings have a normal distribution within a bounded range, event recognition can be implemented by using a threshold-based scheme, which involves a marginal computational overhead, rather than using fairly complicated schedules [10],[11],[12],[13] Due to spatial-correlation, at a particular instant, if the sensed area is larger than the coverage of a single sensor, neighboring sensors sense similar data values. Again, a sensor's own reported readings will be similar to the reading it reported in the previous instant due to the property of temporal correlation. Therefore, identification of sudden, irregular readings deviating from its readings at the previous instants or highly different from its neighbors' readings beyond a pre-specified threshold helps detect faulty sensors [14].

In general, it is best that one deployment can satisfy the needs of a variety of applications in WSN. Take the example of a group of sensor networks deployed in the forest, biologists need to study the environment influence to the growth of the zoology and botany according to its returned detection value; environment scientists study the environment quality in this region and the influence of microclimate in the area; and what the forest managers care about is whether this region would have fire or other disaster. Different users have different needs of the detection data returned from sensors, if you were to respond to every request of the user's query, the sensor networks need to return the same value of detection many times. Therefore, it is significant to one-off creditably collect all row data in the detection region (the so-called "credible information" is that the data that the Sink node receives in the user's pre-specified error limits of credibility with 100%), which enables the different user to conduct the inquiry, analyze and process separately so as to obtain information of their respective needs.

A sensor can give faulty readings (readings different from neighboring sensors or its own readings sensed in previous time intervals, beyond a pre-specified threshold) due to several reasons. For example, the reliability of the equipment is not high or the different batches of the same factory and the sensor may give wrong readings, due to the different manufacturing process and other unforeseen reasons. These are permanent faults that could cause a node to die because of the communication hardware failure[6] .The sensor error can be divided into two categories[15]: one is positive fault, i.e. the sensors report the incident while the environment is in a normal situation; the other is negative fault, i.e., the sensors did not report while there is a specific incident. Therefore, in WSN, to guarantee the credibility of the primary data and eliminate the effects of the error readings is one of the key questions that the event region detection needs to solved.

II. CONSTRUCTION OF AGGREGATION TREE

The construction of aggregation tree is the foundation of event region detection process. The goal is to reduce

* Corresponding author: S.-K Zhang, Tel.:+86 0512 65241247,
E-mail address:zhangsk@suda.edu.cn, Postal address: No 1 Shizhi Street Suzhou China 215006.

doi:10.4304/jsw.4.8.899-906

the data transmission capacity and improve the accuracy of detection. Analysis data aggregation algorithm indicates [16],[17],[18] that, in the case of complete aggregation, seeking for the optimal aggregation tree is equivalent to solve NP-Complete problem of the minimum Steiner tree. According to this NP-Complete problem, it has to consider the balance of the computation processing energy consumption and the transmission energy consumption. In the following part, we will provide a concrete description on the algorithm.

A. Network Model

In this paper, suppose *N* static sensor nodes with resources limited randomly deployed in monitoring area $R=(r\times r)$, denoted by a set $S=(s_1, s_2, \ldots, s_N)$, where s_i is the sensor, as illustrated in Fig.1. Each node has its location information through triangulation [18], and the location of a sensor, s_i is represented by (x_i, y_i) with each node has unique *ID*, same capacity of calculation communication and energy resources. The node through Time Synchronization Service[15] to achieve the loose time synchronism, and the communication access reduces the channel conflict by means of CSMA/CA .The goal of this paper is to construct the aggregation Tree (*AT*) in this *N* nodes network, where *AT* is consisted of N_t nodes called Tree Node, which is used to receive and aggregate data, the other $(N - N_t)$ nodes are referred to as Non-Tree (*NT*) nodes. Each *NT* node senses its environmental parameter and reports it to its nearest tree node. The *AT* is well spread over the entire WSN so that N_t tree nodes are uniformly distributed on the network. In this way, it ensures that the attribute readings sent by *NT* nodes to the corresponding tree node incur a smaller hop count, thereby increasing the overall lifetime of the *NT* nodes. For simplicity, we use P_{event} (denoted by the dashed rectangle in Fig.1) to represent an event and the event region is denoted by the area, R_{event} where $R_{event} \subseteq R$. Normally all the events are assumed to have already been sensed in the network by *AT*. R' is defined as the portion of *R* not occupied by any event, i.e., $R' = R - R_{event}$.

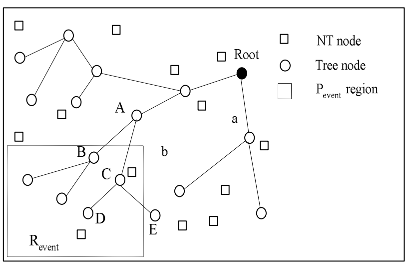

Fig. 1 Network model

B. Generation of Tree

The occurrence of some event can trigger exceptional readings of partial nodes in the WSN, possibly one which is called the isolated spot, also possibly many. In order to guarantee that *AT* diffuse to the entire network, the sensor node transmits the sensed value to the corresponding tree node by small hops, holding the topological stability of dispersion node as far as possible, to maintain the original good sensed coverage area. Therefore, we introduce the graph Voronoi as well as the Delaunay triangle network [1]related to describe the sensor network topology, and based on the definition of Delaunay triangle, construct splay tree in WSN by taking the central node as the root. The splay tree is one kind of binary tree, and its superiority is that it does not need to record the redundant information used in the balanced tree. Let *e* be a spot of plane, then

$$VR(e) = \{p \in R^n \mid d(p,e) \le d(p,e'), \forall e' \ne e, e' \in E\} \quad (1)$$

is called the polygon Voronoi. Then graph Voronoi is defines as

$$VD(E) = \bigcup_i VR(e_i) \quad (2)$$

i.e. set of all polygons Voronoi in plane, but the triangular Delaunay network is formed by the polygon centre for connecting all neighboring. The Delaunay triangle has many important properties [5], it can obtain the neighboring node information of each node through the Delaunay expression. Moreover, it can be used for searching the closest node. We can then construct the splay tree based on Delaunay description in the sensor network: let the target sector be *A*, sensation node collection in the region is

$$S = \{s_i(x_i, y_i) \mid s_i \in A\} \quad (3)$$

Where, (x_i, y_i) is the position coordinate of the known node s_i. In addition, let the weighted graph correspondent by the node collection *S* network is *G* in the region, distance of neighbor nodes is the weight of each side corresponding. Let external memory of the sensed region is in points set $K=\{k_i(x_i,y_i) \mid k_i \notin A\}$, then take node s_i in the target sector as the centre regarding the set of points *K* node extension tree is definited as *T*, has

$$T(s_i -> K) = \bigcup_i path(s_i -> k_i), k_i \in K \quad (4)$$

Where, $path(s_i -> k_i)$ is the greatest span path from node s_i to node k_i in graph *G*, its length is l [12]. In this path, the minimum distance between each node is bigger or equal to the minimum distance in any other path from s_i to k_i and the node number is the smallest in graph *G*. The greatest span path had reflected an extension circuit between two nodes. What needs to be pointed out is that in a specific undirected graph *G*, the greatest span path between two spot is not unique, possibly has multi-strips. But for the different extraterritorial node set, the splay tree of taking the root node as the centre corresponding is also not unique.

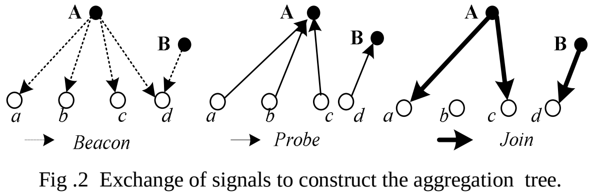

Fig .2 Exchange of signals to construct the aggregation tree.

Assume the depth of the tree is p and the tree node saves the attribute of the same type. Such tree is considered balanced, which reduced data loss and increased accuracy of data aggregation [18]. Algorithm *Form_AT* constructs splay tree with given depth constructs splay tree of aggregation nodes running this algorithm. When a node chooses its two children, it will choose the two biggest span nodes, ensuring that the tree covers more sensed regions as far as possible when aggregation. In the process of the multiple regressions, it can achieve the high accuracy, and may reduce the redundancy of the dissemination monitor value. After the splay tree is formed, in each sub-domains all surplus nodes send data to the nearest tree-node away from themselves. This paper constructs a tree through three kinds of information: Beacon, Probe and JOIN. Fig. 2 described the process about exchange of different signals to construct the query tree.

Construction of splay trees algorithm *Form_AT*(p,p'')

Input: the depth and the b-value.
Output: a binary tree T_c rooted at r of depth at most p and a unique *ID* assigned to each node of T_c.

1: Begin
2: For each level j from 0 to p- l /* l is the largest span path length of inter-node */
For each node i from l to 2^j
3: M_i is a node at level $j+l$
4: n_i is a node at level j
5: n_i sends *Beacon* packet containing n_i's *ID* a_{n_i} to M_i Where distance between n_i and $M_i <$ r /* r is the correspondence radius */
6: M_i chooses n_i as its parent with probability > p"
7: M_i sends *Probe* packet to n_i
8: n_i waits *NWAIT* time (which is a sufficiently loog fixed time period) to receive *Probe* packet from each M_i who selected n_i as parent
9: End

For the sensor network, the node in the path of the largest span has good dispersion, which reduced the influence of capacity of network-sense due to the overlapping coverage, therefore these dispersive good node needs to maintain. Through the definition of the splay tree, determined nodes set need to be maintained in the sensor network. For this set of nodes, the nodes overlapping coverage for the corresponding adjustment of the distribution network will effectively improve the overall perception of ability.

C. Data Aggregation

The main idea of a tree based data aggregation algorithm is using the transmission model that is able to fit more monitor data instead of the monitor data of transmission nodes to reduce the capacity of data transmission, as the result, it saves the energy of the sensor nodes. Hence, it needs to consider the relations between the cost of the return model and data quantity it may fit. The smaller the cost of transmission model, the more data it can express, and the more energy saves. Because the monitor value of node is often subject to many factors, we expect to fit the most data with the minimum cost mode. The multiple linear regression models are totally in line with this goal.

In splay tree, each node receives and stores data reported by the recent non-tree node cyclically to it, that is, the *NT* (Non-Tree) node is responsible for the sensation and *AT* node is responsible to store. The value saved in *AT* node is regarded as the function value of the x-y coordinate. This process describes by three- element (f, x, y), i.e. f is the attribute value transmitted by node located at (x, y). Data tuple of node i stored in *AT* produces the approach function $f_i(x, y)$, and the progressive function f (x, y) by the input of the three variables (z, x, y) forms the implementation of multi-polynomial functions, data in such tree node may denotes by multiple regression polynomial function. The following is to discuss the process of carrying out the data aggregation through the polynomial regression on the splay tree.

In general, the form of multi-dimensional linear regression function is as follows [7]:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k + \mu \tag{5}$$

Where Y is the sensed estimate value, X_j(j=1,2,…,k) are the factors impact to the sensed estimate value Y, β_j(j=0,1,2,…,k) are k +1 unknown regression parameters, μ is random error item .As parameters β_j(j=0,1,2,…,k) are unknown, we can carry on estimate to them using the sample observed value (x_{1i}, x_{2i} ,…, x_{ki} ;Y_i).Through this, we get the parameter estimated value $\hat{\beta}_j$ (j=0,1,2,…,k).Substitute the unknown parameter β_j (j=0,1,2,…, k)of the regression model with the parameters estimated value the, then multi-dimensional linear sample regression equation is:

$$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{1i} + \hat{\beta}_2 x_{2i} + \cdots + \hat{\beta}_k x_{ki} \tag{6}$$

Where $\hat{Y}_i (i = 1, 2, \cdots, n)$ is sample regression value of Y_i .Then the residual e_i between observed value Y_i and the regression value $\hat{Y}_i$ is:

$$e_i = Y_i - \hat{Y}_i = Y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_{1i} + \cdots + \hat{\beta}_{ki} x_{ki}) \tag{7}$$

We can see by the least squares that $\hat{\beta}_j$ (j=0,1,2,…,k) should make the square between all the observations Y_i and the residual e_i the regression value the smallest, even if

$$\begin{aligned} Q(\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2, \cdots, \hat{\beta}_k) &= \sum e_i^2 = \sum (Y_i - \hat{Y}_i)^2 \\ &= \sum (Y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{1i} - \hat{\beta}_2 x_{2i} - \cdots - \hat{\beta}_k x_{ki})^2 \end{aligned} \tag{8}$$

obtains the minimum. According to the extreme value theory of the multiple functions, Q makesthe first partial derivatives respectively for $\hat{\beta}_j$ (j=0,1,2,…,k), and let them equal to zero.

$$\frac{\partial Q}{\partial \hat{\beta}_j} = 0, (j = 1, 2, \cdots, k) \tag{9}$$

After simplified, we get the following equation:

$$X^t Y = X^t X \hat{\beta} . \tag{10}$$

where $\hat{\beta} = [\hat{\beta}_0, \hat{\beta}_1, \cdots, \hat{\beta}_k]^t$, let $R(X)$ = K+1, $X^t X$ is (K+1) step square formation, then $X^t X$ is non-singular, and its inverse matrix is existent, therefore the smallest square estimate vector of β is:

$$\hat{\beta} = (X^t X)^{-1} X^t Y \tag{11}$$

Using the polynomial regression, we obtain the following equation.

$$X = \begin{pmatrix} 1 & y_1 & y_1^2 & x_1 & x_1y_1 & x_1y_1^2 & x_1^2 & x_1^2y_1 & x_1^2y_1^2 \\ 1 & y_2 & y_2^2 & x_2 & x_2y_2 & x_2y_2^2 & x_2^2 & x_2^2y_2 & x_2^2y_2^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & y_n & y_n^2 & x_n & x_ny_n & x_ny_n^2 & x_n^2 & x_n^2y_n & x_n^2y_n^2 \end{pmatrix} \tag{12}$$

$$f(x,y) = \hat{\beta}_0 + \hat{\beta}_1 y + \hat{\beta}_2 y^2 + \hat{\beta}_3 x + \hat{\beta}_4 xy + \hat{\beta}_5 xy^2 + \hat{\beta}_6 x^2 + \hat{\beta}_7 x^2 y + \hat{\beta}_8 x^2 y^2 \tag{13}$$

From the equation (13), we can compute $\hat{\beta}$ through a given location (x, y) and obtain the value of f (x, y) which is property value of (x, y) nodes from (13). As $X^T X$ is certainly the m+1[15] step non-singular, in other words, n>>m+1and X cannot denote for weighted linear combination of any other row set. The data aggregation algorithm mentioned in this paper is accord to the input of the width priority with each tree node has a coefficient from the formula (13) and sends the coefficient set to its parent node. Nodes of each level use the coefficient which obtains from its children to renew sensor attribute value, and these data combine with detection value of node itself to calculate the new coefficient set, and then transfer to a higher level. In the process, to identify the even attribute value in the region is the key of the matter. Because they have a direct bearing on the accuracy of the aggregation, it could identify the region through the upper and lower bounds of the coordinates of the { x_{min} ,y_{min} ,x_{max} ,y_{max} }, where the minimum and maximum value from the Son of the father of the current node in the tree under all the sensor nodes. As in Fig. 3, we set a as the current aggregation node and data value in the region updates through node a. The scope of the region defines by the subtree of node a, which passed through the smallest and largest coordinates of the sensor nodes. Thus, node a gets the border coordinates of the region from its children.

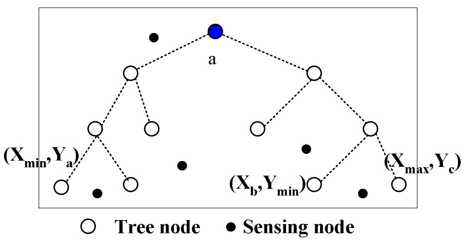

Fig. 3 a node calculate the boundary of the region for data regeneration

Through the construction based on the splay tree and the above description about the process of regression, answer queries every specified time, such as "SELECT temperature FROM sensors WHERE location = (x, y)", or “the highest temperature in target scope “of issues. In the latter case, generate set of (x, y) coordinates in the designated area, Sink firstly informed of the attribute value of each point location to calculate the maximum value. When Sink needs to know the data of (x, y), it will send this inquiry to the root, the inquiry by the AT spreading down until the leaf nodes of the last layer.

III. Event Region Detection

In case of most applications, we use the function $\sigma(i,t) = F(v_1, v_2, ..., v_k)$ to characterize the readings (e.g., temperature, humidity etc.) by a sensor i$\in N$ at instance t where v_k is a parameter that impacts the sensor reading. In most sensor applications, due to known and unknown factors, it is not clear whether the exact expression of the function can be derived. Except some particular cases, it is not easy to model these factors since they may affect the readings in a time-varying manner and in a linear or non-linear way. Rather than trying to obtain an exact expression, we can formulate the basic properties of the function F, thereby analyzing the sensor readings $(\sigma(i,t))$.

A. General Properties of Sensor Readings

The function F which characterizes sensor readings possesses the following properties:

1) F for a sensor i$\in$ N is independent of other sensors.

2) There exist two constants C_{min} and C_{max} (such that $C_{\min} \le \sigma(i,t) \le C_{\max}$) providing the lower and upper bound respectively of the normal readings from a sensor.

3) Within the range [C_{min},C_{max}], $\sigma(i,t)$ is statistically continuous and admits a probability distribution, (i.e.,a normal distribution). Thus, a continuous probability density function (i.e., $\varphi(i)$) can be used to express the distribution.

It formulates those applications in which the sensor readings follow the above properties. Property 1 intuitively explains that a sensor independently senses the environmental changes. Property 2 gives the bounded variable space of normal sensor readings. It is to be noted that different applications have different value for the parameter $\varphi(i)$ given by property 3 and variable spaces. In practice, the assumption can be ascertained by applications that can approximately fit as normal distribution such as daily air temperature, wind speed, etc. In this paper, we determine the conditions of the normal readings and error readings are as follows:

Normal Reading- In general case, $\sigma(i,t)$ is in the range [C_{min},C_{max}] and admits a given normal distribution (i.e., $\varphi(i)$).If $|\sigma(i,t) - E(i)| \ge \tau_1$ for existing variable τ_1 >0. It says that the sensor reading is beyond the normal reading ranges (specified) by τ_1 . The more number of sensors satisfy this condition, the higher probability of an event has occurred.

Faulty Reading- A sensor is classified as faulty if for a sensor i, $\varphi(i)$ satisfies any of the following three conditions.

1) $\forall t, |\sigma(i,t) - E(i)| \geq \tau_2$,for existing variables $\tau_2 > \tau_1 > 0$. When a sensor′ reading is beyond the normal range, it is possible that the sensor is faulty. If a sensor frequently or continuously gives such readings, the probability of the faulty sensor increases.

2) For several consecutive periods of data reporting, it has $|\sigma(i,(t+1) - \sigma(i,t)| \approx 0$ but for each of its neighboring sensors (i′), $|\sigma(i^{'},(t+1) - \sigma(i^{'},t)| >$c, where c is constant. This condition considers the spatial property of sensor readings. It implies that if the neighboring sensors exhibit time-varying readings and the sensor's reading remain constant all the time, there is a high probability that the sensor is faulty. With the same logic, if its neighboring sensors sense a minor change in the reading for a given period (i.e., $|\sigma(i',(t+1) - \sigma(i',t)| \approx 0$) and the sensor records a high range in the sensing reading (i.e.,$|\sigma(i^{'},(t+1) - \sigma(i^{'},t)| \geq$c, where c is a constant), the sensor i may be faulty.

3) $\forall t, C_{\min} < \sigma(i,t) < C_{\max}, \sigma(i,t+1) > \tau_2,$ $C_{\min} < \sigma(i,t+2) < C_{\max}, \sigma(i,t+3) > \tau_2$, This condition implies that the sensor is faulty if the reading of a sensor changes irregularly. If a sensor continues to exhibit irregular reading with time, it has a high probability of being faulty. This condition has some variations in irregular readings. For example, a sensor's reading changes from normal reading to the event reading and back to normal reading periodically or intermittently.

B. Single Event Detection

When a leaf node collects readings from its surrounding *NT* nodes, it computes P_{flag} for an event (i.e., event polynomial) or *P* for a normal phenomenon (i.e., normal polynomial) depending on whether τ_{th} is exceeded or not. While receiving the reports from its child nodes, the parent node has also collected readings from its own surrounding *NT* nodes and has determined whether an event has occurred or not in its region. If a parent node also lies in the same region as any of its child nodes, it regenerates the child node's reading and computes a new polynomial with its own reported data. This polynomial can again be P_{flag} or *P*, depending on whether the parent node is inside the event region or not. If the event regions of a parent node and the corresponding child nodes are different, the child node's data packet is sent up unchanged. The process continues and the root finally receives two polynomials and the corresponding ranges. From the received P_{flag} and the corresponding area from a child node, the root node can get an estimation of the event boundary from the event location information. It analyses the corresponding area, x_{Nmin}, y_{Nmin}, $x_{Nmax,}$ y_{Nmax} (where the suffix N represents an event) to get an estimation of the coverage of P_{event}. Fig. 1 shows a part of the *AT* detecting P_{event}. In Fig. 1, P_{event} has occurred inside R_{event}, at the corner of *R*. We observe that in the sub-tree with the parent node *A* as well as its children *B* and *C* fall in the event region, R_{event}. In this case, both *B* and *C* receive readings from *NT* nodes, with deviation greater than τ_{th} (i.e., $|d_i - E(d_i)| > \tau_{th}$). Again, all the children of *B* lie inside R_{event}. Therefore, the polynomials received by *B* from its two children are flagged (i.e., P_{flag}). Since *B* itself also receives readings from nearest *NT* nodes with deviation greater than the threshold, it generates a new P_{flag} with its own reported data and regenerated data (obtained from P_{flag} sent by its children). The new P_{flag} and the corresponding range are then transmitted to its parent node *A*. In Fig.1 in the sub-tree parented at *C*, one of its children, *D* lies in R_{event} whereas another child, *E* lies in *R*. *C* receives polynomials, P_{flag} ,*P* and the corresponding coordinates x_{Nmin}, y_{Nmin}, $x_{Nmax,}$ y_{Nmax} and x_{min}, y_{min}, x_{max}, y_{max} (taken over all the sensing nodes in the sub-tree under each of nodes *D* and *E* respectively). By observing the approximate span of areas by its two children and its own reported readings, *C* it can be concluded that *D* lies in the same region as with it. Therefore, it regenerates the readings from node *D* and combines those readings with its own reported readings to generate a new P_{flag}. *C* then sends this P_{flag} and x_{Nmin}, y_{Nmin}, $x_{Nmax,}$ y_{Nmax} (denoted by the bold rectangle comprising of the area given by the dotted rectangles spanned individually by nodes *C* and *D*) and unchanged *P* and x_{min},y_{min},x_{max}, y_{max} to *A*. x_{Nmin}, y_{Nmin}, $x_{Nmax,}$ y_{Nmax} gives the range of the area and this range keeps getting modified as it is passed along the AT reflecting the area involving multiple tree nodes lying in R_{event}. When the root node receives x_{Nmin}, y_{Nmin}, $x_{Nmax,}$ y_{Nmax} from a child node also sending P_{flag}, it can roughly get an estimated value according to event P_{event} denoted by the boundary between *R* and R_{event} in Fig.1 and the span of its area.

C. Multiple Events Detection

In contrast to single-event, the exchange of data packets and computational complexity between tree nodes will increase when multiple events occur. With the increase of the event sources, the effect of each event will reduce. As the packets are sent up to the tree, if one of the child nodes of a sub-tree approximates the data range as that of the parent, then a new polynomial is formed by combining both the data sets. However, if the dataset of none of the child nodes is the same as that of the parent, only their data range and range of area span are sent up to the tree. In this case, their areas are combined to form a larger area if both of two child nodes approximate the same data value.

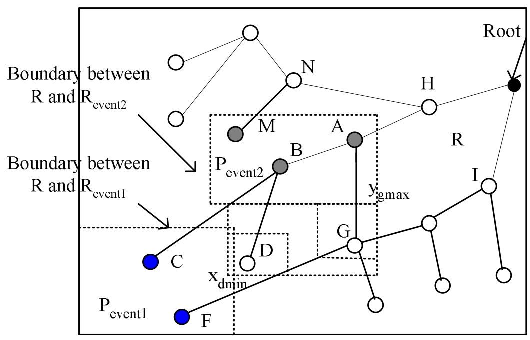

Fig. 4. Occurrence of two new events P_{event1} and P_{event2}.

Suppose in a rectangular area, there are two unusual events and a normal phenomenon. As shown in Fig. 4, two events P_{event1} and P_{event2} occur in region *R*.*B* detects event P_{event2}, while its children nodes *C* and *D* detect P_{event1} and a normal phenomenon separately. In this case, C and *D* transmit respective Polynomial and the coordinate scope {x_{cmin}, y_{cmin}, x_{cmax}, y_{cmax} } as well as { x_{dmin}, y_{dmin}, x_{dmax},y_{dmax} } to *B*. *B* calculates the maximum and minimum data value of the two nodes received, getting the scope of their data. B will transform S_{cmax},S_{cmin},, *C*, as well as the coordinates scope of S_{dmin},S_{dmax}, *D* to *A*. If using single event detection, we need six packets, three of which are corresponding to three polynomials and the other three to the range of areas. Furthermore, *A*'s other child *G* approximates the same phenomenon as *D*. Therefore, *A* combines their ranges together and senses the same event P_{event2}. Then, *A* recreates a new polynomial and sends its polynomial, its range and it's maximum and minimum to its parent *H*. At the same time, M's estimated value is the same as that sent by *A* and sends its coefficients to the root after combining N's regenerated data. As both *H* and *I* lie in *R*, the event region of P_{event2} is not modified further by *H* and is sent to the root unchanged giving the approximated event region and boundary. In this way, both the events P_{event1} and P_{event2} are detected with limited computation overhead as only one polynomial regression is performed at each tree.

IV. SIMULATION RESULTS AND DISCUSSION

This paper accesses simulation results using discrete event simulation platform NS2,which assumes a collision and contention free MAC protocol with simulation parameters shown in table 1, which focus the data aggregation algorithm for performance evaluation.

Table 1 Values of simulation parameters used

Parameter	Variation
Area A	800×800
Communicate radius R	40m
Nodes total D	1630
Node density A/D	0.0025
Event area A_s	400×400
Father node choose probability P "	0.33
Tree depth p	4
Average of reports n_s	12

Place randomly about 250 sensor nodes in the square of 800× 800 units, changes in the temperature 30℃-35℃ for normal and consider the temperature 39℃-49℃ for abnormal as it may be on fire in the region. The depth of the tree is set to 4. From the data set it can be validated that the values of temperature are distributed according to a normal distribution (i.e., $\varphi(i)$) in case of no event. The C_{min},C_{max}, and E(i) are 30, 35, and 32.5 respectively and the variance of $\varphi(i)$ is 1.2. After validation of the assumption, a random location is selected inside *R* and a single event P_{event1} is generated at the centre with a normal distribution of temperature data in the range 55-65℃ $\tau_1 = \tau_{th}$ is set to 2.5, since E(di)- $C_{\min} = C_{\max}$ -E(di). = 2.5 The event area is called R_{event} as we described. τ_1 is set to 2.5, since $E(\sigma(i,t)) - C_{\min} = C_{\max} - E(\sigma(i,t)) = 2.5$ and τ_2 = τ $_{threshold}$ is set to 4.5. The following performance metrics are obtained for multiple rounds of sensing when the event(s) has occurred and the readings have been reported by the *NT* nodes to the nearest tree nodes:

(1) Event boundary: we compare the actual event boundary (of different shapes) and the detected event boundary. (2) Percentage error: Using he final polynomial, the root can obtain readings at the sensor locations where approximation of the actual reading. Therefore, the event area detected based on these readings differ slightly from the actual boundary and this deviation can be expressed as a percentage. This, Percentage Error is defined as the absolute deviation of the approximated reading (i.e., $\overline{z}$) from the true sensor reading (say, z) taken over all the sensors present on the boundary of the event:

$$E = \left(\sum_{k=1}^{n_b} \left(\frac{| z_k - \overline{z}_k}{z_k} \times 100\right) / n_b\right) \le \varepsilon_{th} \qquad (18)$$

Where ε_{th} = 10% is the error threshold given n_b (number of sensors present on the boundary).(3) Event recognition delay: This is defined as the period between the time of occurrence of an event and the final event.

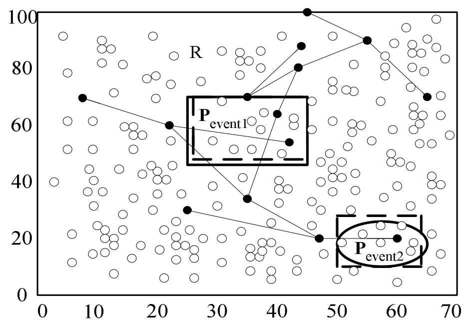

Fig 5. Event boundary approximated by STERD

A. *Detection of an Event Boundary*

We implemented STERD according to the above configuration and obtained an approximated boundary of the event. To study the performance of STERD on occurrence of multiple events with boundaries of different shapes, a second event P_{event2} is generated at a corner of *R* with the temperature range from 5-10℃. Fig. 5 depicts the boundaries of these two events P_{event1} (i.e., the bold rectangle) and P_{event2} (i.e., the bold oval shape). The detected boundaries are given by the dashed rectangles. The shaded dots in the figure represent the tree nodes. The experimental results from our polynomial functions (i.e., P_{flag}) show the accuracy of approximation: 7.8% error and 8.5% error for P_{event1} and P_{event2} respectively. From these approximated boundaries, we observe that the outlines of the reconstructed event regions (i.e., the dash rectangles) match the actual event boundary (i.e., bold rectangle and bold oval), thus confirming the accuracy of STERD. However, since the area approximated by STERD will always be a rectangle (as the event region is reconstructed iteratively by utilizing the ranges of the smaller approximated areas which themselves are rectangles), the event which itself is rectangular is approximated better as compared to other shapes (e.g., oval).

B. Error Rate

Without changing the network size and the number of nodes, we employ STERD to detect a single event and the Extended STERD to detect two events. Fig.6 depicts the actual percentage error in event detection with variation of communication range. The observed percentage errors for different communication ranges are within 10% for both cases of STERD as shown in Fig.6 However, the percentage error of using Multi-STERD (error range from 6.8-8.9) to detect event is 6% higher than using single STERD(error range from 6.4-7.6). For single event detection with STERD, the range is slightly smaller because the approximate boundary can be more accurately pinpointed by regenerating data values from the final P_{flag} which is not available with the Multi-STERD scheme. As shown in Fig. 6 that the error level for the event detection does not significantly vary with communication range, unlike[3]where error increases substantially with the communication range. With increase in the communication range, more sensors with readings from region R_{event} may report to a tree. In STERD, since the tree nodes generate different polynomials corresponding to readings indicating different events, the overall error rate is independent of the communication ranges of each sensor. It can be observed from the result in Fig.6 that with an increase in the density of *NT* nodes, the error percentage increases slightly. With increase in the node density, the probability of a tree node lying at the border of the two events increases, and therefore, the fraction of *NT* nodes lying at the boundary between events in the network increases. Again, with an increase in the node density, accuracy of the approximated polynomial increases, as more readings are considered by tree nodes from reported sensors, relatively larger area is covered and a better approximation is provided of the sensed parameter over the region [18].The latter positive effect on accuracy controls a massive increase in the percentage error due to increase in the node density.

C. Delay Incurred in Event Detection

Fig. 7 shows that the delay in the event detection remains almost constant with increase in the node density. This is because the size of the tree is fixed, irrespective of the number of nodes in the network. Event detection delay mainly results from three parts: computational delay for event recognition (called event recognition delay), computational delay for polynomial (called computation delay), and the delay for event report by packets (called transmission delay). It can be explained from Fig.7 that the event recognition and the computational delay for the polynomial are much smaller than the communication delay. It suggests that communication delay is almost 77% of the total delay. Once the aggregation tree is fixed, the communication delay remains almost constant and is independent of the density of the *NT* nodes. When the node density increases, the number of *NT* nodes around each tree node also increases. However, the result also shows that the computational overhead for event recognition and the construction of the polynomial remain almost constant when the number of nodes (n_{nt}) increases. It is due to the increase of node numbers (n) in the entire network does not cause any significant increase of n_s. STERD reduces the complexity of event recognition and event report because of the following reasons. Firstly, event recognition and polynomial-based data aggregation do not involve any complicated computation in STERD. Secondly, the tree-based network architecture makes the event recognition localized. Thirdly, when the network size (n) increases, the number of tree nodes increases accordingly so that STERD is made scalable.

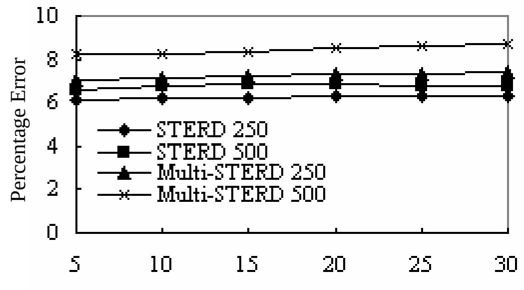

Fig. 6. Percentage error vs. communication range and node density.

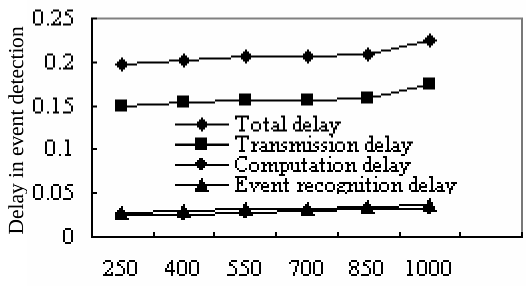

Fig. 7. Delay distribution in event detection vs node density

V. CONCLUSION

This paper has studied high energy efficiency fault-tolerant detection of the wireless sensor network event region. Firstly, we proposed a novel data aggregation algorithm through the construction of the splay tree and this algorithm can also be able to detect the event attribute value lacking of the sensor node position, using the spatio-temporal correlation of the detected event and the error rate in the range of acceptable. In addition, on this basis, an event region detection fault-tolerant algorithm based on the splay tree was proposed, the algorithm can detect a number of events and identify event that occurred in the boundary region and faulty sensor nodes, keeping the error ratio of the overall aggregated data reported to the *BS* under control, quickly conveying of this information to the *BS*, thereby reducing the energy consumption and the delay in data transmission. Results show that with the generation of a large number of packets in the network, error reading detection has nothing to do with the accuracy of the node density; and a faulty sensor can be detected with an average accuracy of 94% and it increases with the increase in the node density, which plays a very important role in the application of sensor networks.

ACKNOWLEDGMENT

Supported by the the National Natural Science oundation of China (60673092,60873116,60873047); Natural Science Foundation of Jiangsu Province of China(BK2008161, BK2008154, BK2009116); The Key Programs of Ministry of Education of China (207040);Jiangsu Provincial Major Program of Science Technique support and independent innovation Foundation (BE2008044); Funded by Preresearch Project of Soochow University; The Opening Project of JiangSu Province Support Software Engineering R&D Center for Modern Information Technology Application in Enterprise (SX200903);The Higher Education Graduate Research Innovation Program of Jiangsu Province in 2009.

REFERENCES

[1] F. Aurenhammer, R. Klein, Handbook of Computational Geometry.Amsterdam, Netherlands, 2000.201-290.

[2] M. Berthold, Intelligent Data Analysis, Springer Press,2nd edition, Feb 2007

[3] J. Chen, S. Kher, A. Somani, Distributed fault detection of wireless sensor networks, In Proc of the 2006 Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks (DIWANS 2006).2006.65-72.

[4] Q. Chen, K.Y. Lam, P. Fan , distributed Bayesian algorithms for fault tolerant event region detection in wireless sensor networks, IEEE Transactions on Computers ,2005,54(9):1182-1183

[5] K.K. Chintalapudi, R. Govindan, Localized edge detection in sensor fields[C],In Proc. IEEE Workshop on Sensor Network Protocols and Applications, Los Angeles May 2003.59-70

[6] M. Ding, D. Chen, et al.Localized fault tolerant event boundary detection in sensor networks, In Proceedings of the Annual IEEE Conference on Computer Communications (INFOCOM),Miami,2005,2:902-913

[7] C. Guestrin, P. Bodi, R. Thibau et al,Distributed Regression: an Efficient Framework for Modeling Sensor Network Data, IPSN. New York, ACM Press,2004

[8] F, Koushanfar. M, Potkonjak, et al, On-Line fault detection of sensor measurements, In Proc of the IEEE Sensors. 2003,2:974-979.

[9] F. Koushanfar, M. Potkonjak, et al, Fault-Tolerance in sensor networks. In: Handbook of Sensor Networks. CRC Press, 2004.

[10] F. Koushanfar, M. Potkonjak, et al. Error models for light sensors by statistical analysis of raw sensor measurements, In Proc of the IEEE Sensors, 2004,3,1472-1475.

[11] B. Krishnamachari, S Iyengar, Distributed Bayesian algorithms for fault-tolerant event region detection in wireless sensor networks, IEEE Trans. on Computers, 2004,53(3):241-250.

[12] T. Le, N. Ahmed, N. Parameswaran, S.Jha, Fault repair framework for mobile sensor networks, in Proc of IEEE International Conference on Communication System Software and Middleware, COMSWARE, 2006,1-8

[13] X. Luo, M. Dong, Y, Huang, On distributed fault-tolerant detection in wireless sensor networks, IEEE Trans. on Computers, 2006, 55(1):58-70.

[14] J.Z. Li, J.B. Li, S.F. Shi, Concepts, issues and advance of sensor networks and data management of sensor networks. Journal of Software, 2003,14(10):1717-1727

[15] R. Nowak, U. Mitra, Boundary estimation in sensor networks: Theory and methods, In Proc. 2nd International Workshop on IPSN, April 2003,2634,86-95

[16] F.Y. Ren, H.N. Huang, C. Lin, Wireless sensor networks. Journal of Software, 2003,14(7):1282-1291

[17] K. Ren, K. Zeng, W. Lou, Secure and Fault-Tolerant Event Boundary Detection in Wireless Sensor Networks, IEEE transactions on wireless communications,2008,7(1):354-363

[18] A. Sharaf, J. Beaver, A. Labrinidis, et al. Balancing energy efficiency and quality of aggregate data in sensor networks, The VLDB Journal,2004,13(4):384-403

ZHANG Shu-Kui is currently an associate professor in the Institute of Computer Science and technology at Soochow University, China. His research areas include ad-hoc and wireless sensor networks, mobile computing, distributing computing, intelligent information processing, parallel and distributed systems etc.

GONG Sheng-Rong is a professor of Institute of Computer Science and technology in Soochow University, China. He received his the PhD in Computer Science from Beijing University of Aeronautics & Astronautics of China in 2001.His research areas are video processing and communications and computer vision.

CUI Zhi-Ming He is a professor and doctoral supervisor at the Institute of Computer Science and technology, Soochow University and a CCF senior member. His researches areas are intelligent information processing, distributing computing, Deep Web data mining etc.

LIU Quan is a professor and doctoral supervisor at the Institute of Computer Science and technology, Soochow University and a CCF senior member. His researches areas are intelligent information processing, distributing computing, automated reasoning and GIS etc.

FAN Jian-Xi received the BS, MS, and PhD degrees in computer science from Shandong Normal University, Shandong University,and City University of Hong Kong, China, in 1988, 1991, and 2006, respectively. He is currently a professor of computer science in the School of Computer Science and Technology at Soochow University, China.His research interests include parallel and distributed systems, interconnection architectures, design and analysis of algorithms.