Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 160523, 8 pages
http://dx.doi.org/10.1155/2013/160523

Research Article

Distributed Fault-Tolerant Event Region Detection of Wireless Sensor Networks

Dyi-Rong Duh,[1] Ssu-Pei Li,[2] and Victor W. Cheng[2]

[1] *Department of Computer Science and Information Engineering, Hwa Hsia Institute of Technology, New Taipei City 23568, Taiwan*
[2] *Department of Computer Science and Information Engineering, National Chi Nan University, Nantou Hsien 54561, Taiwan*

Correspondence should be addressed to Dyi-Rong Duh; drduh@cc.hwh.edu.tw

Received 23 January 2013; Accepted 10 August 2013

Academic Editor: Hongju Cheng

This work provides a distributed fault-tolerant event region detection algorithm for wireless sensor networks. The proposed algorithm can identify faulty and fault-free sensors and ignore the abnormal readings to avoid false alarm. Moreover, every event region can also be detected and identified. Simulation results show that fault detection accuracy (FDA) is greater than 92%, false alarm rate (FAR) is near 0%, and event detection accuracy (EDA) is greater than 99% under uniform distribution. FDA is greater than 92%, FAR is less than 1.2%, and EDA is greater than 88% under random distribution when sensor fault probability is less than 0.3.

1. Introduction

The wireless sensor network (WSN) is a novel technology developed in recently years [1–9]. A wireless sensor network (WSN) consists of a large number of sensors. These sensors are used to monitor environmental variations such as temperature, humidity, pressure, and concentration of chemicals. Sensors in a WSN have limited computation, communication, and sensing capabilities because they are low cost and energy-constrained. Moreover, sensors are often deployed in an uncontrolled and harsh environment. They are prone to be faulty and hard to maintain. Therefore, a fault-tolerant and energy-efficient algorithm is required for network operation [4, 10–13].

The communication protocol of a WSN is very important in order to reduce power consumption. Three general-purpose protocol architectures include direct transmission protocol [15, 16], routing protocol [17–19], and clustering protocol [1, 3, 7, 11, 20–24]. Clustering is a popular energy-efficient protocol in WSNs [3, 11, 21, 23]. It divides the monitored area into several clusters, and each cluster has a cluster head. A cluster head can be regarded as a local fusion center. Each cluster member will send its physical value or decision to its cluster head, and the cluster head can thus make data aggregation or data fusion to eliminate invalid values and report this result to the base station (BS). Significantly, the challenge of designing a clustering WSN is that a cluster head is easy to exhaust its energy, and hence the network will be out of control. Heinzelman et al. proposed the LEACH algorithm [11]. In LEACH, cluster heads are randomly chosen, and a self-organization procedure is performed. This protocol balances the energy load between sensors. Heinzelman et al. later proposed the improved centralized algorithm as LEACH-C [21]. In LEACH-C, each sensor takes its turn as the cluster head according to the remaining energy. Therefore, the energy load is more evenly balanced than that in the LEACH algorithm. Nocetti et al. proposed a clustering algorithm based on connectivity [23]. The efficiency of this algorithm can be measured by the number of clusters formed and the number of border nodes produced.

Chen et al. proposed a distributed fault detection algorithm in [25]. Each sensor compares the observed data with its neighbors and makes a local decision by majority of votes. Krishnamachari and Iyengar proposed a distributed Bayesian algorithm [26]. In this algorithm, faults can be detected and corrected, but it also introduces new errors. Wu et al. [14] proposed an algorithm to solve the fault-event disambiguation problem. They focused on two typical cases, the event regions with ellipses or straight lines as boundaries.

This work provides a distributed and fault-tolerant algorithm to extend the uses of fault-event disambiguation. It can identify not only faulty and fault-free sensors but also the region where the event occurs. Simulation results demonstrate that the proposed algorithm has high accuracy and low false alarm in any shape of event regions.

The rest of this paper is organized as follows. Section 2 defines the network model and fault model. Section 3 describes the algorithm. Section 4 makes some simulations and shows their results. Conclusion and future work are drawn in Section 5.

2. Network Model and Fault Model

For ease of reference, frequently used notations and definitions in this paper are listed in Table 1. Each sensor s_i in a WSN has an observation x_i from its location. This observation is independent and identically distributed on phenomenon from sensor to sensor. θ_d is defined as a threshold for checking whether two readings are in the same situation. For example, to get the relationship and to decide the status of two sensors s_i and s_j, the measurement difference d_{ij} is compared with θ_d. When $d_{ij} > \theta_d$, at least one of s_i and s_j is abnormal. θ_{ev} is defined as the event threshold. When s_i is a fault-free sensor and x_i is greater than θ_{ev}, s_i is identified as an event sensor. θ_D and θ_f are defined as the thresholds of degree and faith, respectively. These two thresholds are used in the procedure of making the strong decision by a cluster head, which will be further described in the next section. Assume that each sensor has a unique ID and can use the power control mechanism to vary the transmitting power, and the transmission range in the WSN is independent from sensor to sensor. If a sensor is far away from all other sensors, it can enhance its transmitting power to increase the transmission range and communicate with other sensors.

Notably, the fault model of this work assumes that computation and communication capabilities of each sensor always work properly. Three types of sensing fault are defined as stuck-at-maximum fault, stuck-at-minimum fault, and random fault. When a sensor is on the stuck-at-maximum fault, on the stuck-at-minimum fault, or on the random fault, it reports the maximum physical value, the minimum physical value, or a random physical value uncorrelated to the environment, respectively. These faulty sensors make the network unreliable and affect the final decision of the WSN. Therefore, a fault-tolerant algorithm to identify and isolate faulty sensors is designed in this work, and hence the precision of the event region detection is improved.

3. Proposed Algorithm

Assumes that the BS is capable of broadcasting a global message to sensors everywhere and each sensor knows its own geographical location either through GPS or RF-based beacons [27]. Each cluster head collects data from its cluster members. The data may contain abnormal readings. Therefore, this work provides an algorithm to identify those sensors that get abnormal readings. The proposed algorithm adopts the correlative relationship to distinguish event sensors or faulty sensors. Faulty sensors are likely to be uncorrelated, and sensors in the event region are spatially correlated. The algorithm is divided into two phases, clustering phase and decision phase.

Table 1: Notations and definitions.

n	Total number of sensors
s_i	The sensor with ID i
$N(s_i)$	The set of neighbors of s_i
$M(s_i)$	The set of neighbors of s_i that have not joined any cluster
D_i	The degree of s_i
B_i	The set of cluster(s) that s_i belongs to
x_i	The measurement of s_i
d_{ij}	The measurement difference of s_i and s_j, $d_{ij} = \lvert x_i - x_j \rvert$
c_{ij}	The measurement status between s_i and s_j, $c_{ij} \in \{0, 1\}$, $c_{ij} = c_{ji}$
w_i	The weight of cluster head i
f_i	The faith of cluster head i, $f_i = w_i/D_i$
P_i	The status of s_i, $P_i \in$ {GD, FT, EV, UD}
θ_d	The measurement difference threshold
θ_D	The degree threshold
θ_f	The faith threshold
θ_{ev}	The predefined event threshold
T_{rcv}	The timer for receiving cluster invitation
T_{self}	The timer for the self-decision
T_{weak}	The timer for the weak decision

Some other definitions are described as follows. If $d_{ij} \le \theta_d$, where s_i is a cluster head and $s_j \in N(s_i)$, then $c_{ij} = c_{ji} = 0$ and w_i increases by one. If $d_{ij} > \theta_d$, then $c_{ij} = c_{ji} = 1$ and w_i does not increase. Weight w_i is used to calculate f_i. Four statuses of a sensor in a WSN are defined as Good (GD), Faulty (FT), Event (EV), or Undetermined (UD).

3.1. Clustering Phase. The clustering phase starts when the BS broadcasts a clustering message to all sensors. In this phase, the monitored environment is divided into several clusters by cluster heads. Each cluster head can be regarded as a local fusion center. It collects information from its members and reports the fusion to the remote BS. Therefore, a well-designed clustering rule is to avoid chaotic situations such that a cluster head has no members or two cluster heads exist in the same cluster. The clustering rule of this work uses the degree of a sensor as the primary key and the sensor ID as the secondary key. At the first stage, each sensor exchanges degrees with its neighbors. Every sensor s_j, which has $D_j = 0$ and does not belong to any cluster, is classified as an isolated sensor. Every sensor s_i with the highest degree among $M(s_i)$ and itself is chosen as the cluster head. The remaining sensors start a timer T_{rcv} for receiving invitations. If two or more sensors have the same maximum degree as s_i and $M(s_i)$, the sensor with the lowest sensor ID is chosen to be the cluster head. Each cluster head sends an invitation to its neighbors to include them as its cluster members and ends the clustering phase. When a sensor receives an invitation, it joins the cluster. Every sensor will check if it has joined any cluster

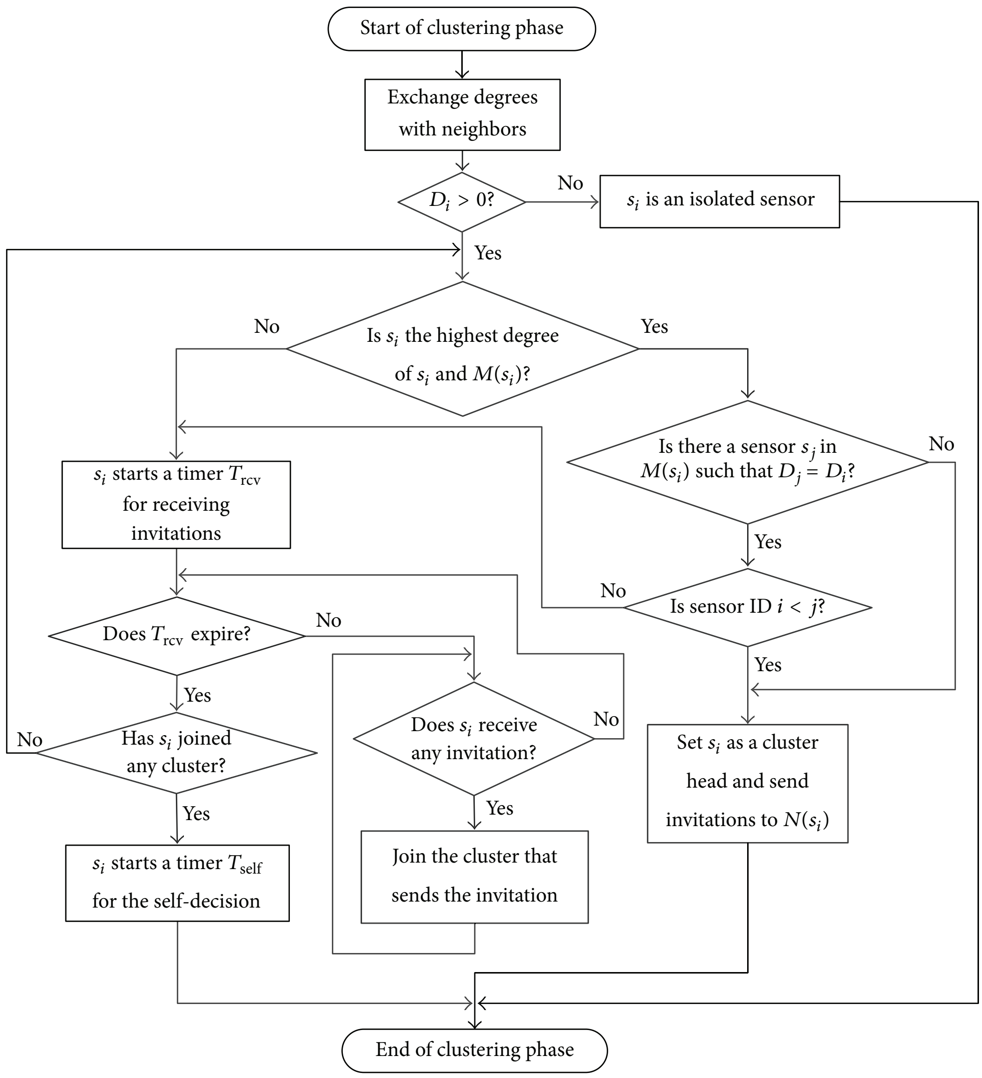

Figure 1: The flow chart of the clustering phase.

when T_{rcv} expires. If yes, it starts a timer T_{self} for the self-decision which will be described in the decision phase, and the clustering phase is completed. If not, it continues to check if it is the highest degree among $M(s_i)$ and itself as is in the previous step. In the clustering phase of the algorithm, each cluster head forms its own cluster and each member sensor may belong to more than one cluster. Because all neighbors of a cluster head are its cluster members, the clustering rule used in the algorithm guarantees that every cluster head is not adjacent to any other cluster head; that is, there is only one cluster head in each cluster. Figure 1 shows the flow chart of the clustering phase.

3.2. Decision Phase. Initially, the status of each sensor is set to UD. This phase is divided into three kinds of decision: strong decision, weak decision, and self-decision. An important mechanism in this phase is that when sensor s_j is set to GD or EV, it will cease to allow other sensors to reset its status. On the other hand, when s_j is set to FT, P_j may be switched to the same status as P_k by s_k if $s_k \in N(s_j)$ and P_k = GD or EV to reduce false alarms. The strong decision is only performed in the cluster heads whose degree and faith exceed the thresholds. The weak decision is performed in the cluster heads with status UD when T_{weak} expires. The self-decision is performed in the cluster members with status UD when T_{self} expires. Each decision is described in detail as follows.

3.3. Strong Decision. First, each member sensor s_j sends the observation x_j to its cluster head(s), and each cluster head s_i calculates d_{ij}, c_{ij}, w_i, D_i, and f_i where $s_j \in N(s_i)$. If $D_i \geq \theta_D$, $f_i \geq \theta_f$, and P_i = UD, cluster head s_i sets P_i to GD when $x_i < \theta_{\mathrm{ev}}$ or EV when $x_i \geq \theta_{\mathrm{ev}}$ and determines the statuses of $N(s_i)$. If $D_i \geq \theta_D$, $x_i \geq \theta_{\mathrm{ev}}$, $f_i \geq \theta_f/2$, and P_i = UD, cluster head s_i sets P_i to EV and also determines the statuses of $N(s_i)$. Otherwise, s_i starts a timer T_{weak} for the weak decision described in next paragraph and exits current decision. Second, during the status propagation, s_i checks c_{ij} to set P_j to UD or FT for all $s_j \in N(s_i)$. If $c_{ij} = 0$, then P_j is copied from P_i, and s_j continues to set the statuses of $N(s_j)$. If $c_{ij} = 1$ and P_i = GD, P_j is set to FT, and s_j terminates

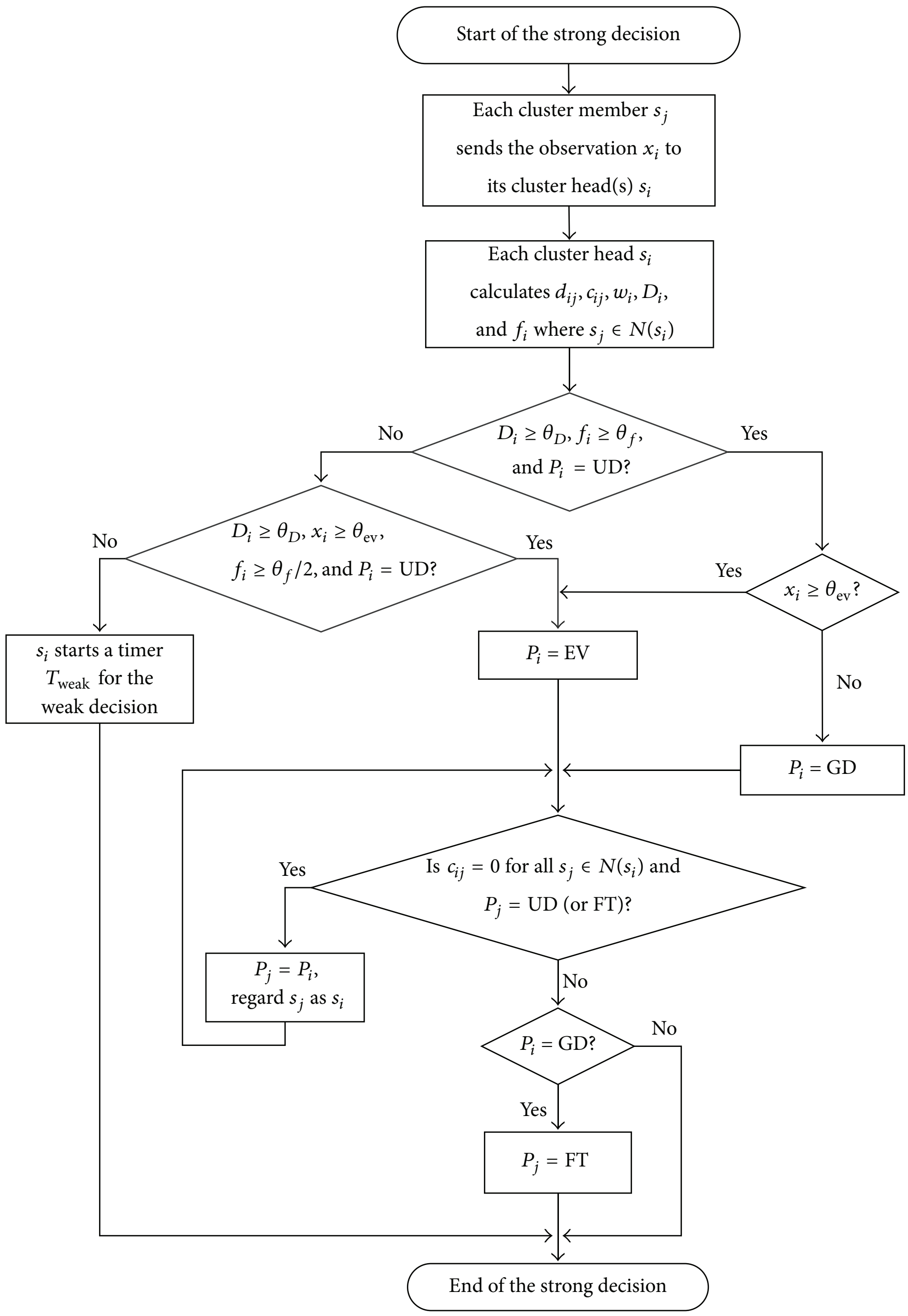

FIGURE 2: The flow chart of the strong decision.

the status propagation. Figure 2 illustrates the flow chart of the strong decision.

3.4. Weak Decision. When T_{weak} of cluster head s_i expires, s_i with P_i = UD starts the weak decision. Based on majority vote, if $f_i > 0.5$, s_i sets P_i to GD when $x_i < \theta_{ev}$ or EV when $x_i \geq \theta_{ev}$, and s_i checks c_{ij} to set P_j to UD or FT for all $s_j \in N(s_i)$. If $c_{ij} = 0$, P_j is copied from P_i. Moreover, if $c_{ij} = 1$ and P_i = GD, P_j is set to FT. If $f_i \leq 0.5$, s_i sets P_i to FT and terminates the weak decision. Figure 3 illustrates the flow chart of the weak decision.

3.5. Self-Decision. When T_{self} of member sensor s_j expires, s_j first starts reclustering described in next paragraph and then checks P_j. If P_j = UD, s_j searches for a GD sensor s_k in $N(s_j)$ and calculates c_{jk}. If $c_{jk} = 0$, P_j is set to GD. If $c_{jk} = 1$, P_j is set to FT. Notably, it is possible that no sensor with status GD can be found in $N(s_j)$. When that happens, s_j uses its

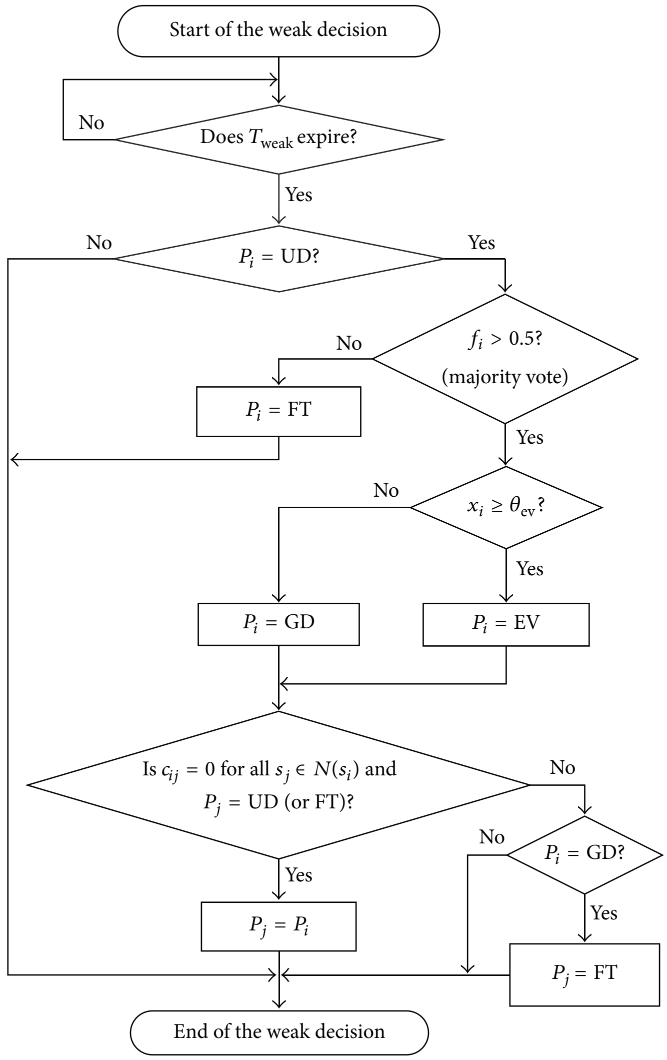

Figure 3: The flow chart of the weak decision.

maximum transmission range for searching a sensor with status GD. Nevertheless, if a sensor with status GD cannot be found within the range, P_j is set to FT. Figure 4 illustrates the flow chart of the self-decision.

Reclustering is an important approach in the self-decision. If a cluster head s_i is set to FT, its members who do not belong to any other cluster must be reclustered because a faulty cluster head cannot properly report data to the BS. The erroneous data may cause incorrect final decisions at the BS and thus the WSN may miss some event regions. For example, s_i and s_h are two cluster heads, and P_i is set to FT, and P_h is set to GD. There exist two cluster members s_j and s_k of cluster s_i such that $B_j = \{i, h\}$ and $B_k = \{i\}$. In this case, since s_i is faulty, s_j removes i from B_j and belongs to cluster h only. Therefore, s_j does not require reclustering. On the other hand, after removing i from B_k, s_k does not belong to any cluster and its observations will be ignored. Therefore, s_k requires reclustering. After reclustering, everything is accomplished by the new cluster head. Every sensor is included in some cluster after the decision phase, except for all isolated sensors.

4. Simulation Results

The simulation platform is a PC with Windows XPSP3, and the simulation program is developed in C#. The network parameters in this simulation are preset as follow. The transmission range of a sensor is between 2 m and 10 m. Thresholds θ_d, θ_{ev}, and θ_f are set to 5, 60, and 0.66, respectively, and θ_D is set to the average degree of the network. Notably, with the aid of experiments, we tuned every threshold carefully to get higher FDA and EDA and lower FAR. Each physical datum of any observation is between 0 and 100. The fault detection accuracy (FDA) is defined as the ratio of the number of faulty sensors detected to the total number of faulty sensors. The false alarm rate (FAR) is defined as the ratio of the number of fault-free sensors diagnosed as faulty to the total number of fault-free sensors. The event detection accuracy (EDA) is defined as the ratio of the number of event sensors detected to the total number of sensors located in the event region.

The result of the proposed algorithm will be compared with that of a recent algorithm for event boundary detection

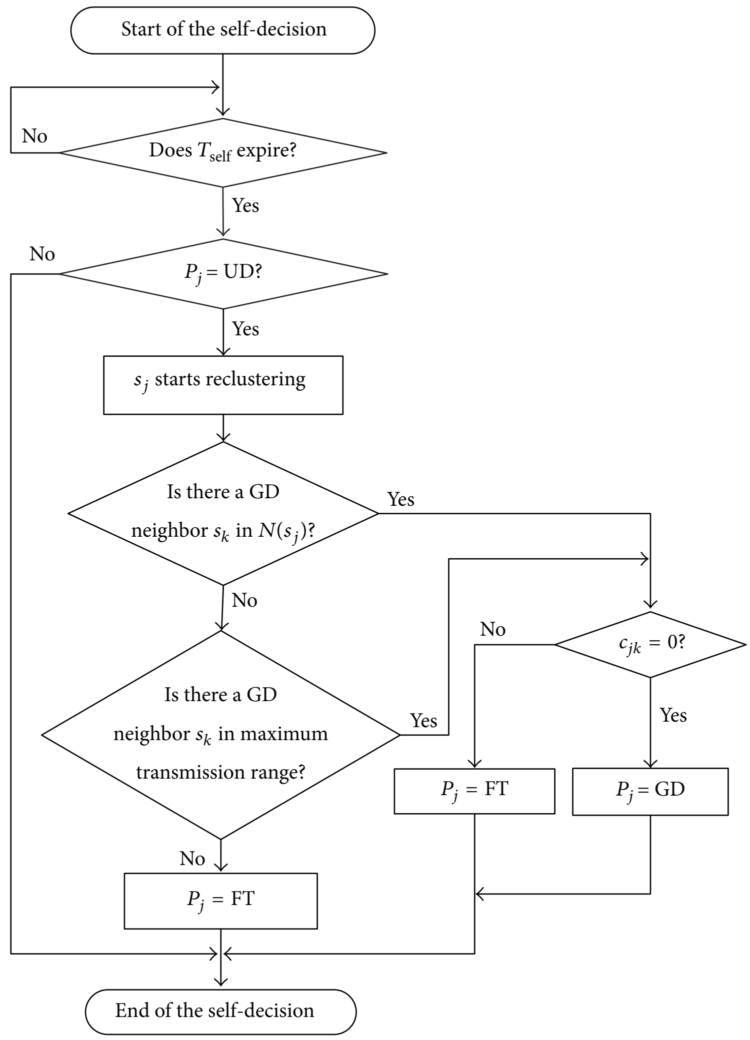

Figure 4: The flow chart of the self-decision.

with faulty sensors in a WSN by Wu et al. [14]. For ease of comparison, the network setup is the same as that of Wu et al. 4,096 sensors are uniformly distributed in a 64 m × 64 m square region and represent an averaged summary of 100 runs. Figure 5 illustrates the FDA of the algorithm provided by Wu et al. and the proposed algorithm. Figure 6 shows the FAR of Wu et al.'s algorithm and the proposed algorithm.

Referring to Figure 5, the FDA of Wu et al.'s algorithm is slightly better than that of the proposed algorithm when average degree is between 20 and 50. This is because our simulation includes random faults, and random faulty sensors may catch random physical data near normal range (or event range), which are difficult to be correctly detected. Nevertheless, the FDA of Wu et al.'s algorithm is very inaccurate, while average degree is 10. In contrast, the FDA of the proposed algorithm remains close to 93% under various average degrees. The FARs of Wu et al.'s algorithm and the FAR of the proposed algorithm are illustrated in Figure 6. Undoubtedly, higher FAR always gets higher FDA. Obviously, our algorithm thus makes a significant improvement in FAR since it is almost 0%. The key point is that when a fault-free sensor s_j has been diagnosed as FT, its status may be switched to GD as long as there exists a sensor with status GD in $N(s_j)$ running the status propagation.

Figure 7 demonstrates that the FDA of our algorithm decreases smoothly with the sensor fault probability from 0.1 to 0.4 and average degree 10. Actually, a high degree network demands high cost and is not practical for a large scale network. Therefore, the proposed algorithm is more suitable for WSNs.

Figure 8 shows the results of our algorithm that 250 sensors are randomly distributed in a 30 m × 30 m square region with average degree of 10, and the predefined event region is a square with side length 8 m located in the middle. The FDA is greater than 92%, the FAR is less than 1.2%, and EDA is greater than 88% when sensor fault probability is less than 0.3. Figure 9 illustrates the simulation result that the BS uses the convex hull algorithm to identify the event region. In Figure 9, red, black, and blue nodes indicate GD, FT, and EV sensors, respectively. The black circles are clusters, the green square is the predefined event region, and the purple polygon is the detected event region.

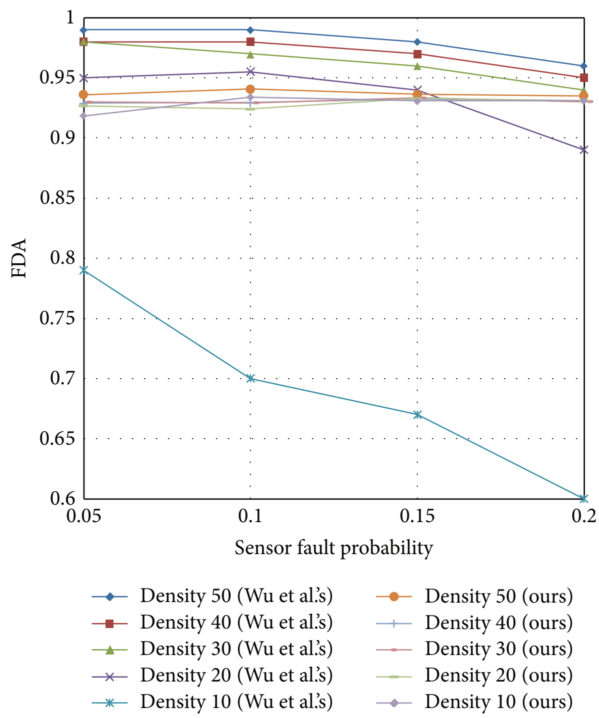

FIGURE 5: The FDA of Wu et al.'s algorithm [14] and the proposed algorithm.

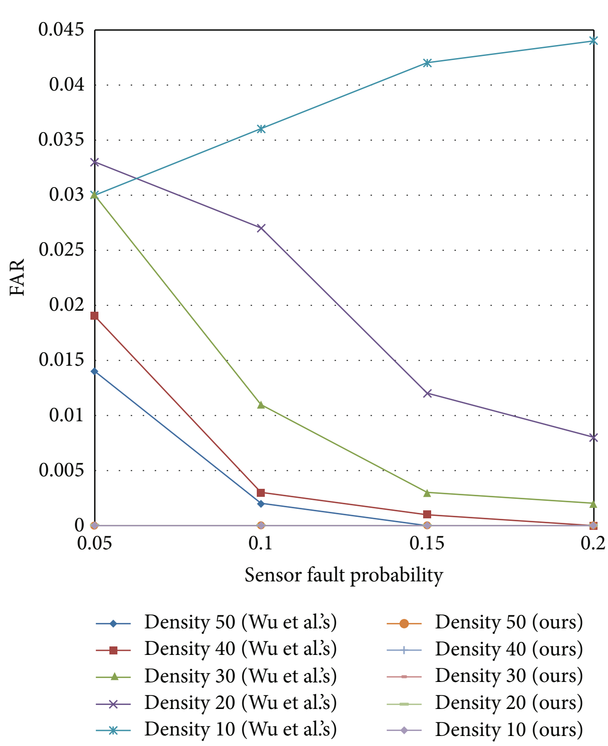

FIGURE 6: The FAR of Wu et al.'s algorithm [14] and the proposed algorithm.

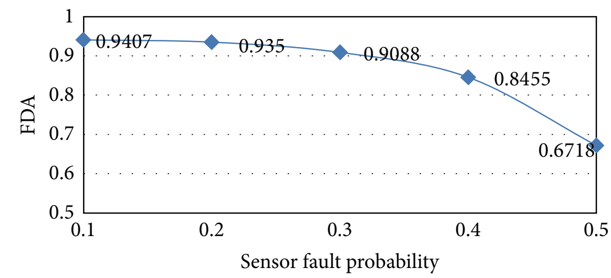

FIGURE 7: The FDA of our algorithm in a WSN with an average degree of 10.

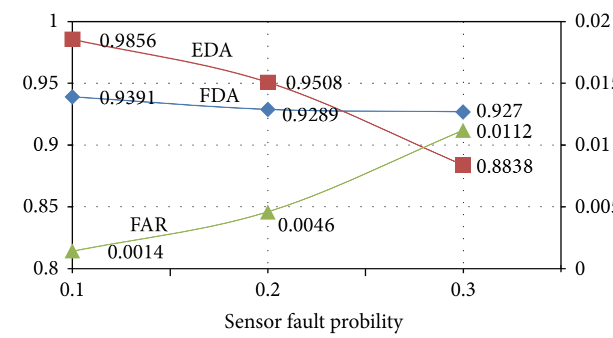

FIGURE 8: FDA, FAR, and EDA of our algorithm in a WSN when the sensors are randomly distributed with an average degree of 10.

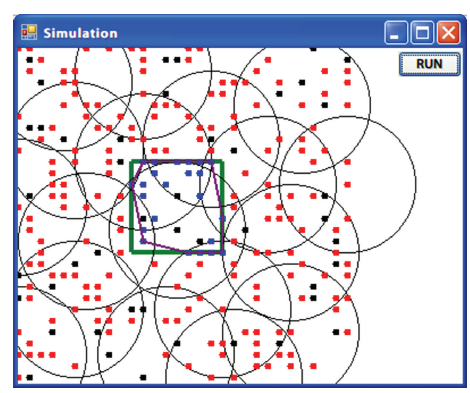

FIGURE 9: Convex hull algorithm is used to identify the event region at the BS. (Red: GD sensor; black: FT sensor; blue: EV sensor; green: the predefined event region; purple: the detected event region).

5. Conclusion

This work provides a distributed fault-tolerant algorithm for event region detection of WSNs to solve the fault-event disambiguation problem. It may be widely used in abnormal region detection. For example, sensors can be deployed in a forest by an airplane to discover promptly forest fires by monitoring the temperature around each sensor. The BS can identify the abnormal region based on the report of the cluster heads and send out a fire alarm signal. The proposed algorithm has high accuracy on event detection and low false alarm. Our future work will focus on power consumption and cluster head rotation to extend system life time. Moreover, to compute the best value of every threshold is our further research.

Acknowledgment

The authors would like to thank the National Science Council, Taiwan, for financially supporting this research under Contract NSC 100-2221-E-146-014.

References

[1] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Communications*, vol. 30, no. 14-15, pp. 2826–2841, 2007.

[2] J. Agre and L. Clare, "An integrated architecture for cooperative sensing networks," *IEEE Computer*, vol. 33, no. 5, pp. 106–108.

[3] N. Amini, A. Vahdatpour, W. Xu, M. Gerla, and M. Sarrafzadeh, "Cluster size optimization in sensor networks with decentralized cluster-based protocols," *Computer Communications*, vol. 35, no. 2, pp. 207–220, 2012.

[4] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[5] D. Culler, D. Estrin, and M. Srivastava, "Overview of sensor networks," *Computer*, vol. 37, no. 8, pp. 41–49, 2004.

[6] D. D. Geeta, N. Nalini, and R. C. Biradar, "Fault tolerance in wireless sensor network using hand-off and dynamic power adjustment approach," *Journal of Network and Computer Applications*, vol. 36, no. 4, pp. 1174–1185, 2013.

[7] J.-W. Lin and D.-R. Duh, "Distributed fault detection of wireless sensor networks," in *Proceedings of the 2010 International Conference on Wireless Networks (ICWN '10)*, pp. 472–477, Monte Carlo Resort, Las Vegas, Nev, USA, July 2010.

[8] J.-Y. Wu, D.-R. Duh, and T.-Y. Wang, "On-line fault-tolerant decision fusion scheme based on a record table in wireless sensor networks," in *Proceedings of the Conference on Information Technology and Applications in Outlying Islands (ITAOI '13)*, pp. 249–255, Jinning, Kinmen, Taiwan, May 2013.

[9] D. Xia and N. Vlajic, "Near-optimal node clustering in wireless sensor networks for environment monitoring," in *Proceedings of the International Conference on Electrical and Computer Engineering (CCECE '06)*, pp. 1825–1829, Ottawa, Ontario, Canada, May 2006.

[10] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: scalable coordination in sensor networks," in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM '99)*, pp. 263–270, Seattle, Wash, USA, August 1999.

[11] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Siences (HICSS-33 '00)*, pp. 1–10, January 2000.

[12] A. Manjeshwar and D. P. Agrawal, "TEEN: a routing protocol for enhanced efficiency in wireless sensor networks," in *Proceedings of the 15th International Symposium on Parallel and Distributed Processing (IPDPS '01)*, vol. 3, pp. 2009–2015, San Francisco, Calif, USA, April 2001.

[13] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the 21st Annual Joint Conference of the IEEE International Conference on Computer and Communications Societies (INFOCOM '02)*, pp. 1567–1576, New York, NY, USA, June 2002.

[14] W. Wu, X. Cheng, M. Ding, K. Xing, F. Liu, and P. Deng, "Localized outlying and boundary data detection in sensor networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 8, pp. 1145–1156, 2007.

[15] L.-C. Chen, D.-R. Duh, and T.-Y. Wang, "Using a saturation counter to improve the accuracy of final decisions in wireless sensor networks," in *Proceedings of the International Conference on Wireless Networks (ICWN '09)*, pp. 124–129, Monte Carlo Resort, Las Vegas, Nev, USA, July 2009.

[16] T.-Y. Wang, L.-Y. Chang, D.-R. Duh, and J.-Y. Wu, "Fault-tolerant decision fusion via collaborative sensor fault detection in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 2, pp. 756–768, 2008.

[17] M. Ettus, "System capacity, latency, and power consumption in multihop-routed SS-CDMA wireless networks," in *Proceedings of the IEEE Radio and Wireless Conference (RAWCON '98)*, pp. 55–58, Colorado Springs, Colo, USA, August 1988.

[18] T. J. Shepard, "Channel access scheme for large dense packet radio networks," in *Proceedings of the Annual Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '96)*, pp. 219–230, Stanford, Calif, USA, August 1996.

[19] S. Singh, M. Woo, and C. S. Raghavendra, "Power-aware routing in mobile and ad hoc networks," in *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM '98)*, pp. 181–190, Dallas, Tex, USA, 1998.

[20] D. J. Baker, A. Ephremides, and J. A. Flynn, "The design and simulation of a mobile radio network with distributed control," *IEEE Journal on Selected Areas in Communications*, vol. 2, no. 1, pp. 226–237, 1984.

[21] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.

[22] C. R. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 7, pp. 1265–1275, 1997.

[23] F. G. Nocetti, J. S. Gonzalez, and I. Stojmenovic, "Connectivity-based k-hop clustering in wireless networks," *Telecommunication Systems*, vol. 22, no. 1–4, pp. 205–220, 2003.

[24] R. Ruppe, S. Griswald, P. Walsh, and R. Martin, "Near Term Digital Radio (NTDR) system," in *Proceedings of the 1997 Military Communications Conference (MILCOM '97)*, vol. 3, pp. 1282–1287, Monterey, Calif, USA, November 1997.

[25] J. Chen, S. Kher, and A. Somani, "Distributed fault detection of wireless sensor networks," in *Proceedings of the Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks (DIWANS '06)*, pp. 65–71, Los Angeles, Calif, USA, September 2006.

[26] B. Krishnamachari and S. Iyengar, "Distributed Bayesian algorithms for fault-tolerant event region detection in wireless sensor networks," *IEEE Transactions on Computers*, vol. 53, no. 3, pp. 241–250, 2004.

[27] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *IEEE Personal Communications*, vol. 7, no. 5, pp. 28–34, 2000.