International Journal on
"Technical and Physical Problems of Engineering"
(IJTPE)
Published by International Organization of IOTPE

ISSN 2077-3528
IJTPE Journal
www.iotpe.com
ijtpe@iotpe.com

March 2012	Issue 10	Volume 4	Number 1	Pages 52-58

AN EFFECTIVE FUZZY BASED ALGORITHM TO DETECT FAULTY READINGS IN LONG THIN WIRELESS SENSOR NETWORKS

A. Barati [1] S.J. Dastgheib [2] A. Movaghar [3] I. Attarzadeh [2]

1. Department of Computer Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran, abarati@iaud.ac.ir
2. Department of Computer Engineering, Dezful Branch, Islamic Azad University, Dezful, Iran, dastgheib@gmail.com, attarzadeh@iaud.ac.ir
3. Department of Computer Engineering, Sharif University of Technology, Tehran, Iran, movaghar@sharif.edu

Abstract- Wireless sensor networks (WSNs) consist of hundreds or thousands tiny nodes called sensor that work together and connected to each other to do some special tasks. It is expected that WSNs will be used widely in many applications in the near future. Localisation and detecting node with faulty readings are the two challenging issues in WSNs, as all the sensors communicate to each other to process and share their data. Existing algorithms use correlation of two nodes to detect faulty readings, which resulted in high computational complexity. Also, algorithms that use inverse of distance as scale of similarity have weaknesses in accuracy of calculations. This paper proposed a new fuzzy based algorithm to overcome the localisation problem and detecting faulty readings in WSNs. Using an effective fuzzy inference system can improve the decision-making algorithm, which use for localisation and detecting faulty readings in WSNs. The experimental results show that the proposed algorithm reduces computational complexity of correlation of two nodes, which resulted in reducing energy consumption of each network's node. Also, the accuracy of the results is improved when compared to the other algorithms.

Keywords: Wireless Sensor Networks (WSNs), Long Thin Network (LTN), Faulty Readings, Localisation, Network Energy Consumption.

I. INTRODUCTION

Wireless sensor networks consist of hundreds or thousands tiny nodes called sensor that work together to do some special tasks. Each node consists of sensor, processor, antenna, a tiny memory and one energy supply. Because of the limited energy resources in WSN, it is required to use algorithms with less energy consumption. Localisation and detecting faulty readings are the two fundamental challenges in WSNs. Localisation is determining location of nodes in the network [1].

One method is to use a Global Positioning System (GPS) device in each node, which would be very expensive in terms of energy consumption and cost. The network sensors are prone to produce incorrect data in noisy and unreliable environments. Faulty sensors produce arbitrary readings, which do not reflect the true state of environmental phenomenon or events under monitoring. Furthermore, sensors may report incorrect readings, which come from wrong inference and decision-making. Both arbitrary and noisy readings are caused faulty readings. These failures may even prevent the benefit of WSNs. Thus, it is important to identify and filter the faulty readings. There are few research works on the accurate localisation algorithms and detecting faulty readings [2].

The Long Thin Network (LTN) is a type of network topologies, which widely use in wireless sensor network applications. Some application of LTN are in surveillance application, ranges from leakage detection of fuel pipes, monitoring tunnels, stage measurements in sewer, street lights monitoring in highway systems, flood protection of rivers, vibration detection of bridges, roadside networks, pedestrian detection systems and etc. In LTN, nodes may form several long backbones and these backbones extend the network to intend coverage areas. A backbone is a linear path which may contain tens or hundreds of routers [3, 4].

Many algorithms have been proposed for detecting faulty readings of a sensor in WSN. These algorithms usually use a voting method to detect faulty readings. Voting methods can be classified to majority voting and weighted voting methods. In WSNs, reading data from neighboring sensors can be done using spatial correlation, that this information can be used to detect faulty readings. In this case, if a sensor (i.e. S_j) receives an unusual reading, this sensor asks neighbors if the reading is faulty or not. It can be done through sending an acknowledgment to them, then using majority voting technique to confirm the correct result.

In the classic majority voting, each node among a set of witnesses (e.g. node S_i) makes a judgment by comparing its reading with an abnormal reading submitted by the suspect node. If the difference between these two readings exceeds from a predefined threshold, S_i considers the reading posted by S_j as faulty and gives a negative vote to the S_j. Otherwise, S_i claims that S_j is normal and a positive vote returns to S_j. After collecting votes from the neighbors, S_j decides if the reading is faulty or not. If the number of negative votes is more than positive votes, reading reports of unusual S_j considers as a faulty reading [5].

The main aim of this paper is to propose a novel algorithm to detect faulty readings, which resulted in low energy consuming and better efficiency. Due to the structure of WSN, a node senses values from neighboring nodes, the difference between these sensed values can be used as a parameter to determine faulty nodes. The difference value can be represented by a linguistic variable in fuzzy logic. For example, the difference sensed values of two nodes can be shown as low, medium, or high in fuzzy logic as listen read phonetically. This paper uses an adaptive fuzzy inference system to detect faulty readings in WSN. Also, to reduce the effect of faulty readings, a confidence number is used, which obtained from Debraj De algorithm. Using fuzzy rule-based system can efficiently help to detect the faulty nodes in WSN. Furthermore, it can reduce the complexity of voting algorithms based on correlation, which resulted in increasing the accuracy of faulty nodes detection.

The rest of this paper organises as follows. In section II, fault tolerant deployment techniques for long thin wireless sensor networks are discussed. Section III shows error detection techniques of localisation, which includes basic concepts of detection of localisation error. In section IV, the characteristic of Debraj De algorithm for localisation error detection is discussed. The fuzzy logic concept and its characteristics and applications describe in section V. Section VI, proposes an effective fuzzy based algorithm to detect faulty readings in WSN. Experimental results, analysis, and comparison the obtained results with the other models are discussed win section VII. Finally, section VIII concludes this paper.

II. FAULT TOLERANT DEPLOYMENT TECHNIQUE FOR LONG THIN WIRELESS SENSOR NETWORKS

The structure of nodes in Long Thin Network (LTN) can produce errors in the network, as each node has less number of neighbors. This paper also presents an optimised structure for deployment of nodes in Long Thin network. This can be useful for a wide range of practical applications. The proposed optimised structure of LTN is shown in Figure 1. It shows the optimal sub-structure pattern of nodes that is repeated along the entire network. In this sub-structure, there are two parallel linear distributions with multiple lines. Therefore, each node in this distribution has at least four neighbors [6, 7]. In Figure 1, node N_2 is at a distance away from its closest neighbors on the same line, and distance D away from the closest neighbors on the other line, and distance B away from two lines.

According to the definition of rectangular triangle:

$$d^2 = b^2 + \left(\frac{a}{b}\right)^2 \tag{1}$$

Therefore, the number of neighbors of a node can also increase by increasing the number of parallel lines and changing the value of parameters a, b.

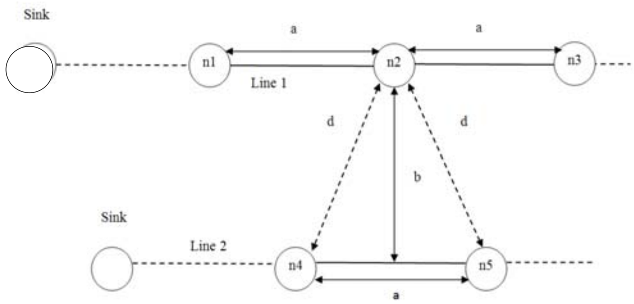


Figure 1. Proposed fault tolerant sub-structure for Long Thin Network

III. BASIC CONCEPT FOR LOCALISATION ERROR DETECTION

In localisation error detection technique, it is assumed that all the sensor nodes are aligned in X-Y plane. As shown in Figure 2, node N_i and N_j observes each other at angles P_{ij} and P_{ji}, respectively [7].

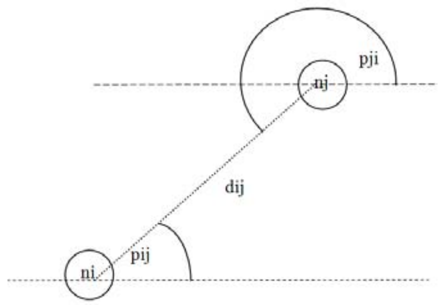


Figure 2. Basic concept behind localisation error detection

Also, suppose that they are at distance D_{ij}. Then coordinate of node N_j with respect to N_i is:

$$X_{ij} = d_{ij}.\cos p_{ij} \ , \ Y_{ij} = d_{ij}.\sin p_{ij} \tag{2}$$

Similarly, coordinate of node N_i with respect to N_j is:

$$X_{ji} = d_{ji}.\cos p_{ji} \ , \ Y_{ji} = d_{ji}.\sin p_{ji} \tag{3}$$

If there is no error in the calculation of distance and angle, then clearly $(X_{ji}+X_{ij})=0$ and $(Y_{ji}+Y_{ij})=0$. Therefore, each node can detect any error in localisation by checking the summation of $(X_{ji}+X_{ij})$ and $(Y_{ji}+Y_{ij})$, that should be zero.

IV. DEBRAJ DE ALGORITHM FOR LOCALISATION ERROR DETECTION

Because of static localisation in the most long thin network deployments, the localisation is prone to various kinds of errors. Therefore, a localisation algorithm with less complexity, full distribution, self-organising, and secure with error correction would be useful [8, 9]. The localisation algorithm is described with respect to node N_2 in the optimal sub-structure of LTN in Figure 3.

Each node in the network runs the same algorithm as in node N_2. The Debraj De algorithm uses localisation error detection, which can be summarised as follows [10].

- Node N_2 distributes a Hello message M_1, which is received by all of its neighbors, N_1, N_3, N_4 and N_5
- Then N_2 receives message M_1 from each of its neighbors
- From these messages, N_2 calculates relative position for each neighbor using Angle of Arrival (AOA) and distance information. Suppose that N_2 calculates its neighbor N_j*'s* relative position by (X_{2j}, Y_{2j}) from the angle of arrival A_{2j} and distance D_{2j} information. Therefore:

$$X_{2j} = D_{2j}.\cos A_{2j} \quad (4)$$

$$Y_{2j} = D_{2j}.\sin A_{2j} \quad (5)$$

- N_2 sends feedback message M_2 with calculated position information (X_{2j}, Y_{2j}) to all neighbors N_j.
- Node N_2 receives feedback M_2 with information (X_{j2}, Y_{j2}) from each of its neighbor N_j.
- Then the value of (X_{j2}+X_{2j}) and (Y_{j2}+Y_{2j}) should be zero. Due to different source of errors, it would not always be zero. Interestingly, these values capture all the possible errors that may affect accuracy of localisation.
- Faulty detection in the network: if N_2 gets only one message M_2 from N_j, then it saves the relative position information (X_{2j}, Y_{2j}). N_2 calculates (X_{j2}+X_{2j}) and (Y_{j2}+Y_{2j}).

If $(X_{j2}+X_{2j}) \le X_{error-threshold}$ and $(Y_{j2}+Y_{2j}) \le Y_{error-threshold}$, then N_2 can rely node N_j, and set a confidence level of node N_j, $Confidence_{2j}$=*trust_value*, Otherwise N_2 cannot absolutely rely on accuracy of N_j. Thus, Confidence level can be set to $Confidence_{2j}$=*non_trust_value*. Typically *trust_value* and *non_trust_value* can be represented by 3 and 1, respectively. Therefore, every node decides whether it can rely its neighbors information or not.

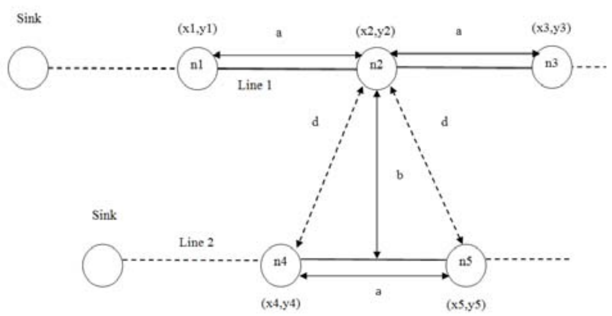


Figure 3. Localisation error detection on long thin topology

V. FUZZY LOGIC

Fuzzy logic (FL) is defined as the logic of human thought, which is much less rigid than the calculations computers generally perform. Fuzzy Logic offers several unique features that make it a particularly good alternative for many control problems. It is inherently robust since it does not require precise, noise-free inputs and can be programmed to fail safely [11-14]. The output control is a smooth control function despite a wide range of input variations.

Since, the FL controller processes user defined rules governing the target control system, it can be modified and tweaked easily to improve or drastically alter system performance. Fuzzy Logic deals with the analysis of information by using fuzzy sets, each of which may represent a linguistic term like "Warm", "High", etc. Fuzzy sets are described by the range of real values over which the set is mapped, called domain, and the membership function. A membership function assigns a truth value between 0 and 1 to each point in the fuzzy set's domain. Depending upon the shape of the membership function, various types of fuzzy sets can be used such as triangular, beta, PI, Gaussian, sigmoid, etc [15].

A Fuzzy system basically consists of three parts: fuzzifier, inference engine, and defuzzifier. The fuzzifier maps each crisp input value to the corresponding fuzzy sets and thus assigns it a truth value or degree of membership for each fuzzy set. The fuzzified values are processed by the inference engine, which consists of a rule base and various methods for inferring the rules. The rule base is simply a series of IF-THEN rules that relate the input fuzzy variables with the output fuzzy variables using linguistic variables, each of which is described by a fuzzy set, and fuzzy implication operators AND, OR, etc. The part of a fuzzy rule before THEN is called predicate or antecedent, while the part following THEN is referred to as consequent. The combined truth of the predicate is determined by implication rules such as MIN-MAX and bounded arithmetic sums.

All the rules in the rule-base are processed in a parallel manner by the fuzzy inference engine. Any rule that fires contributes to the final fuzzy solution space. The inference rules govern the manner in which the consequent fuzzy sets are copied to the final fuzzy solution space.

For example, techniques are MIN-MAX and fuzzy adaptive method. The defuzzifier performs defuzzification on the fuzzy solution space. That is, it finds a single crisp output value from the solution fuzzy space. Common defuzzification techniques are centroid, composite maximum, composite mass, etc.

VI. PROPOSING AN EFFECTIVE FUZZY BASED ALGORITHM TO DETECT FAULTY READINGS IN WSN

A combination of fuzzy logic, fuzzy inference system, the confidence number obtained from Debraj De detection faulty readings algorithm, and voting algorithm are used to propose an effective algorithm for detecting faulty reading in WSN. In the proposed algorithm, each weight is defined as difference value between the two nodes in WSN.

Each difference value can be represented by linguistic variables in fuzzy logic. The difference values can be shown by "Low", "Medium", and "High". The structure of adaptive fuzzy system, which applied into the proposed algorithm, is shown in Figure 4.

In the first step of fuzzification, a function needs to convert obtained values from the difference between two nodes, crisp values, into the fuzzy linguistic values: Less, Medium or High, as shown in Figure 5.

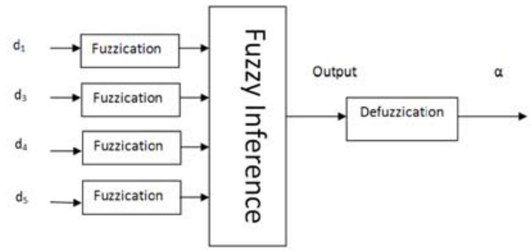


Figure 4. The structure of proposed fuzzy system

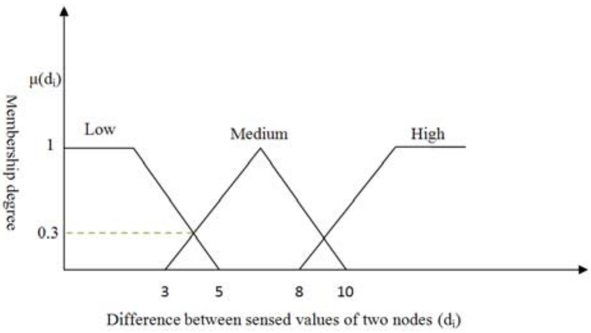


Figure 5. Membership function for the difference between sensed values of two nodes

Decision making on the D value, depends on environments and network topology. Based on the proposed fuzzy logic model for detecting faulty reading in WSN, steps of voting algorithm are described as follows.

- Assume that node N_2, in Figure 3, is a suspect node. To detect faulty readings, N_2 send information to its neighbors as follows:
  - Obtaining the difference between sensed values by node N_2 and the other nodes based on the following formula.

$$D_i = \left| S_i - S_2 \right|$$

In this formula D_i is difference between sensed values of two sensors. Also S is sensed value by a sensor.

  - Fuzzification: each node based on the following membership function, choose one of the Low, Medium, or High values for D_i.
  - After fuzzification phase, each node sends the obtained fuzzy value (Low, Medium, or High) to the node N_2.
- A faulty node in neighbors of the decision maker node, can destroy the result of decision-making algorithm. To overcome this problem, a confidence number that obtained from localisation error detection technique was used. To control the faulty neighbor node the following calculation is used, before sending the output of fuzzy system to the node N_2.

If $Confidence_{i,j}$=*non_trust_value* and D_j=*High* then D_j=*Medium*
Else if $Confidence_{i,j}$=*non_trust_value* and D_j=*Medium* then D_j=*Low*

- The level of faulty readings can be calculated by using fuzzy rule-based technique in the node N_2. The fault can be defined in five levels:

Very Low (VL), Low (L), Medium (M), High (H), Very High (VH)

There are 4*4*4*4 rules in the proposed fuzzy system, as some of them shown as follows and in Table 1.
If D_1=Low and D_3=Low and D_4=Low and D_5=Low then *Output*=Very Low Faulty

Table 1. Example of fuzzy rules

Rule #	D_1	D_3	D_4	D_5	*Output*
1	L	L	H	H	M
2	M	L	M	H	M
3	H	H	M	L	H
4	H	H	M	H	VH
5	H	H	H	H	VH
6	L	L	L	M	VL
7	L	L	M	H	L

- Defuzzification: in this step, the sensor reading values are used to make decision on each event in WSN. Sensed values by nodes are used to make decision on faulty readings. To avoid wrong decision making and ensure obtaining correct results, the following solution is considered. A coefficient value is assigned to faulty value of each node (Low, Very Low, etc.). In this case, if there is a faulty node with a coefficient less than one, e.g. α=0.8), the effect of this faulty node on the decision making is Low, as shown as follows.

If *Output*=Very Low Faulty then α=1
If *Output*=Low Faulty then α=0.88
If *Output*=Medium Faulty then α=0.7

According to statistics, the other two levels, High and Very High, have come to the conclusion that they consider 100% faulty. In these two levels readings of the node N_2 is not valid at all. In this case, the node returns average values of its neighbors as sensed value. Given that the neighbor nodes may read the faulty values, therefore, they send sensed values with coefficient (α) to the node N_2. The proposed fuzzy based algorithm to detect faulty reading in WSN is as follows.

```
DetectFaultyReadingByFuzzyLogic (Node_j) {
For (i=1; i<=4;i++) {
Send S_j to nigborors And Request LMH, S_i From They // Low , Medium or High (LMH)
S[i]=S_i ; LMH[i]=LMH ; Level=Call Procedure decision (LMH) ;
If Level=VH Or Level=H then {
  Report("This Node is damaged") ; S_j=Call Procedure Repair(S) }
if Level=M {
  Report("This Node is Medium Faulty But Still Useful"); S_j=0.7*S_j }
if Level=L {
  Report("This Node is Low Faulty And Still Useful") ; S_j=0.88*S_j }
If Level=VL
```

Report("This Node is Correct") / / End of Voting in $Node_j$
}

ProcessInNigborNode(S_j) { //$Node_i$
d=ABS(S_j-S_i) // $d=|S_j-S_i|$; if $d<\alpha$ then LMH=Low // α and β are constant value
elseif $\alpha<=d<\beta$ then LMH=Medium ; elseif $d<=\beta$ then LMH=High ;
if $Confidence_{ij}$=1 And LMH=High then LMH=Medium
if $Confidence_{ij}$=1 And LMH=Medium then LMH=Low
Return S_i, LMH
}

VII. RESULTS, ANALYSIS AND DISCUSSION

This section compares existing algorithms with the proposed fuzzy based algorithm. The existing algorithms can be classified into the two categories. The first category is established based on inverse of distance and the second category is established based on the correlation between sensors readings. The algorithms based on distance inverse are very vulnerable to faulty nodes. Besides, the algorithms based on correlation are very complex and costly. Debraj De faulty detection algorithm belongs to the first category. The complexity rate of these algorithm is high and it is overall about $O(n^3)$. Assume the LTN in Figure 6.

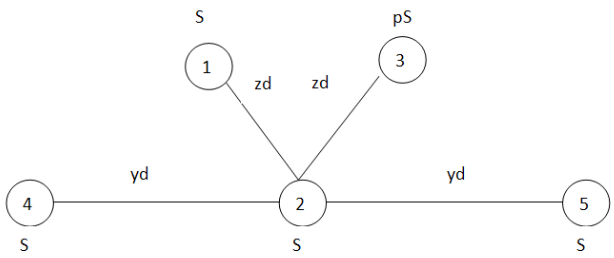


Figure 6. LTN Network

In Figure 5, S is sensed value by the nodes and D indicated the distance. Z, Y, P are fixed numbers. According to Debraj De voting algorithm, the following relations can be obtained [17-19]. These relationships are used to detect the faulty readings of node 2. Node 3 is a faulty node that is going to destroy the voting result.

$$w_{12}=\frac{3}{zd}\quad w_{23}=\frac{1}{zd}\quad w_{24}=w_{25}=\frac{3}{yd} \tag{6}$$

$$Vote_i=\frac{3}{zd}*S-\frac{1}{zd}*pS+\frac{6}{yd}*S \tag{7}$$

$$Vote_i=\frac{S}{d}\left(\frac{3}{z}-\frac{p}{z}+\frac{6}{y}\right) \tag{8}$$

Node 3 can destroy voting, if:

$$\frac{3}{z}-\frac{p}{z}+\frac{6}{y}<0 \xrightarrow{\text{Multiply xy in both sides}} 3y+py+6z<0 \tag{9}$$

wz can be written instead of y, then:

$$3wz-wpz+6z<0 \xrightarrow{\text{Discard } z} 3w+6<pw \tag{10}$$

In this case, if unequal value is produced, then voting algorithm produces wrong value. e.g., if $w = 2$ then

$$3\times 2z-2pz+6z<0 \longrightarrow 6<p \tag{11}$$

According to this example, if P is greater than 6 voting is failed. It shows the inaccuracy of results of this algorithm. Also, by decreasing the distance between faulty node and voting node (increasing w), the effect of faulty node on voting highly increases, as shown in Figure 7.

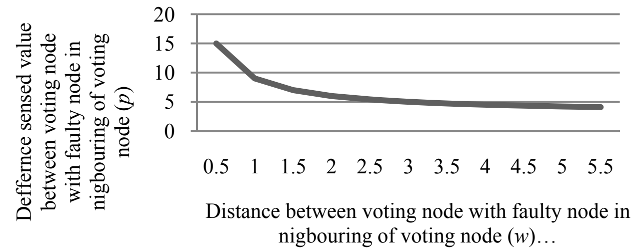


Figure 7. Effect of faulty nodes on voting

If the proposed fuzzy based fault detection algorithm applied on the same data, the results are:
$cd_1=|S-S|=0$ → LMH[1]=Low
$cd_3=|S-ZS|$ → LMH[3]=High because $confidence_{23}$=1 → LMH[3]=Medium
$cd_4=d_5=0$ → LMH[4]=LMH[5]=Low
because LMH[1]=Low and LMH[4]=Low and LMH[5]=Low and LMH[3]=Medium → $Output$=VL

The results show that the voting is absolutely correct, which shows the high accuracy in fault detection. Also, the complexity rate of the proposed algorithm is $O(n)$, which is much better that $O(n^3)$ [19]. Table 2 shows the results of comparison between the proposed fuzzy based faulty readings detection algorithm and the other algorithms.

Table 2. Comparing the results between the proposed fuzzy based faulty readings detection algorithm and the other algorithms

Criteria	Proposed Fuzzy Based Algorithm	Weighted based on correlation of nodes sensed values	Debraj De	Weighted based on inverse distance
Precision	Very High	Very High	Medium	Less
Complexity	Less	Very High	Less	Less
Energy Consumption	Less	Very High	Less	Less

Table 2 shows that the result of comparison between the fuzzy based faulty readings detection algorithm and the other algorithms based on precision, complexity, and energy consumption criteria. The comparison results shows that the proposed model shows more accurate results when compared with the other models.

VIII. CONCLUSIONS

The main aim of this research was to propose an effective fuzzy based faulty readings detection model in wireless sensor networks. Besides, localisation error detection, the proposed model can detect the faulty readings and overcome the faulty nodes. In the proposed algorithm, after localisation error detection, confidence value of each node was obtained, which used to detect faulty readings. Also, the fuzzy inference system was used to detect faulty readings in WSN. The experiment results showed that the fuzzy based faulty readings detection algorithm significantly reduces computational complexity of the calculation. Besides, it showed very high accuracy when compared with the other existing algorithms.

REFERENCES

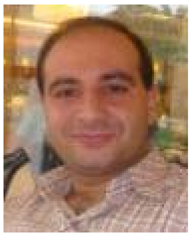
[1] R. Stoleru, J.A. Stankovic, S. Son, "Robust Node Localisation for Wireless Sensor Networks", Proc. of EmNets, 2007.
[2] A. Terzis, A. Anandarajah, K. Moore, "Slip Surface Localisation in Wireless Sensor Networks for Landslide Prediction", Proc. of 5th IEEE/ACM Int'l Conference on Information Processing in Sensor Networks (IPSN '06), pp. 109-116, April 2006.
[3] A. Gopakumar, J. Lillykutty, "Distributed Wireless Sensor Network Localisation Using Stochastic Proximity Embedding", Computer Communications, pp. 745-755, Vol. 33, No. 6, 2010.
[4] M. Vecchio, R. Lopez Valcarce, F. Marcelloni, "A Two-Objective Evolutionary Approach Based on Topological Constraints for Node Localisation in Wireless Sensor Networks", Applied Soft Computing, 2011.
[5] N. Patwari, A.O. Hero Ill, "Using Proximity and Quantized RSS for Sensor Localisation in Wireless Networks", Proc. of WSNA, 2003.
[6] A. Ali, T. Collier, L. Girod, K. Yao, D.T. Blumstein, C.E. Taylor, "An Empirical Study of Collaborative Acoustic Source Localisation", Proc. of IPSN, 2007.
[7] A. Srinivasan, J. Wu, "A Survey on Secure Localisation in Wireless Sensor Networks", Encyclopedia of Wireless and Mobile Communications, B. Furht (ed.), CRC Press, Taylor and Francis Group, 2007.
[8] H. Chen, W. Lou, J. Ma, Z. Wang, "TSCD: A Novel Secure Localisation Approach for Wireless Sensor Networks", Proc. of 2nd International Conference on Sensor Technologies and Applications (SensorComm), pp. 661-666, August 2008.
[9] F. Caballero, L. Merino, P. Gil, I. Maza, A. Ollero, "A Probabilistic Framework for Entire WSN Localisation Using A Mobile Robot", Robotics and Autonomous Systems, pp. 798-806, 2008.
[10] H. Aksu, D. Aksoy, I. Korpeoglu, "A Study of Localisation Metrics: Evaluation Of Position Errors in Wireless Sensor Networks", Computer Networks, in Press, Corrected Proof, July 2011.
[11] G.J. Jordt, R.O. Baldwin, J.F. Raquet, B.E. Mullins, "Energy Cost and Error Performance of Range-Aware, Anchor-Free localisation Algorithms", Ad Hoc Networks, Vol. 6, pp. 539-559, 2008.
[12] E. Elnahrawy, B. Nath, "Poster Abstract: Online Data Cleaning in Wireless Sensor Networks", Proc. of 1st International Conference on Embedded Networked Sensor Systems, pp. 294-295, 2003.
[13] A.A. Allahverdiyev, "Cargo Transportation Routing Under Fuzzy Conditions", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 6, Vol. 3, No. 1, pp. 45-48, March 2011.
[14] A.A. Allahverdiyev, "Application of Fuzzy-Genetic Algorithm for Solving an Open Transportation", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 7, Vol. 3, No. 2, pp. 119-123, June 2011.
[15] D. De, "A Distributed Algorithm for localisation Error Detection-Correction, Use in In-Network Faulty Reading Detection: Applicability in Long Thin Wireless Sensor Networks", Proc. of the IEEE Wireless Communications and Networking Conference, pp. 1-6, April 2009.
[16] B. Krishnamachari, 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.
[17] T. Sun, L.J. Chen, C.C. Han, M. Gerla, "Reliable Sensor Networks for Planet Exploration", Proc. of the IEEE International Conference on Networking, Sensing and Control (ICNSC), pp. 816-821, 2005.
[18] X. Xiao, W. Peng, C. Hung, W. Lee, "Using Sensor Ranks for In-Network Detection of Faulty Readings in Wireless Sensor Networks", Proc. of MobiDE, pp. 714-721, 2007.
[19] A. Strehl, J. Ghosh, R. Mooney, "Impact of Similarity Measures on Web-Page Clustering", Proc. 7th National Conference on Artificial Intelligence: Workshop of Artificial Intelligence for Web Search (AAAI), pp. 58-64. July 2000.

BIOGRAPHIES

Ali Barati was born in Dezful, Iran, in February 1981. He received the B.Sc. degree in Computer Hardware Engineering from Dezful Branch, Islamic Azad University, Dezful, Iran, in 2001. He also received the M.Sc. degree in Computer Software Engineering from Arak Branch, Islamic Azad University, Arak, Iran, in 2004. He is currently a member of Department of Computer Engineering, Dezful Branch, Islamic Azad University, Dezful, Iran. He is currently working toward the Ph.D. degree in Computer Software Engineering at Qazvin Branch, Islamic Azad University, Qazvin, Iran. His research interests include AD-HOC networks, wireless sensor networks, high speed networks and fault tolerant systems.

Seyyed Jalaleddin Dastgheib was born in Shiraz, Iran, in 1986. He received the B.Sc. degree in Computer Software Engineering from Tabriz University, Tabriz, Iran, in 2008. He also received the M.Sc. degree in Computer Hardware Engineering from Dezful Branch, Islamic Azad University, Dezful, Iran, in 2011. His research interests include computer arithmetic, wireless sensor network, fuzzy systems, and fault tolerant systems.

Ali Movaghar received his B.Sc. in Electrical Engineering from University of Tehran, Tehran, Iran in 1977 and both M.Sc. and Ph.D. in Computer Information and Control Engineering from University of Michigan at Ann Arbor in 1979 and 1985, respectively. He visited INRIA in France in 1984, work in AT&T laboratories during 1985-1986, taught in University of Michigan during 1987-1989. He is currently a Professor at Department of Computer Engineering of Sharif University of Technology, Tehran, Iran where he joined first an Assistant Professor in 1993. His research interests include performance and dependability modeling, verification and validation, computer networks and distributed real time systems. He is a senior member of the IEEE.



Iman Attarzadeh was born in Dezful, Iran, in 1980. He received his Bachelor in Computer Science (Computer Architecture) with certificate of appreciation of excellent graduate student in 2002 from Dezful Branch, Islamic Azad University, Dezful, Iran, in 2002, and his M.Sc. in Computer Architecture from Science and Research Branch, Islamic Azad University, Tehran, Iran, in 2005. Between 2002 and 2007, he was a member of Faculty of Computer Science and Information Technology, Dezful Branch, Islamic Azad University, Dezful, Iran. Between 2005 and 2007 he was appointed as the Head of Department of Computer Science at the University. His current research interests include software engineering (project management, formal method, algorithm design, and embedded systems), soft computing theories and techniques, image processing, and computer Architecture.