An Optimised Algorithm to Detect Faulty Readings along the Substrate Access Wireless Long-Thin Sensor Networks

Ali Barati
Department of Computer Engineering
Dezful Branch, Islamic Azad University,
Dezful, Iran
abarati@iaud.ac.ir

S.Jalalledin Dastgheib
Department of Computer Engineering
Dezful Branch, Islamic Azad University,
Dezful, Iran
dastqeib@gmail.com

Ali Movaghar
Department of Computer Engineering
Sharif University of Technology
Tehran, Iran
movaghar@sharif.edu

Iman Attarzadeh
Department of Computer Engineering
Dezful Branch, Islamic Azad University,
Dezful, Iran
attarzadeh@iaud.ac.ir

Abstract— **Wireless sensor networks (WSNs) are composed of hundreds or thousands of small nodes, which work together and associate with a specific task or tasks to do. It is expected that wireless sensor networks will be used widely in many applications in the near future. One of the most important issues in WSNs is localisation. There are crucial problems over network localisation such as security attacks (internal or external), energy efficiency, and accuracy, which impact performance and energy-consuming of wireless sensor networks. The main source of these problems is network topology. A long-thin network topology (LTNT) in wireless sensor can produce errors in network localisation due to special deployment of nodes; also it can cause detecting nodes with faulty readings. This paper proposes an optimised algorithm based on Debraj De algorithm to determine faulty readings in WSNs. This algorithm uses a correlation parameter of two nodes to detect nodes with faulty readings. The proposed algorithm reduces computational complexity of the correlation algorithm, which causes network energy consumption becomes significantly low when compared with the original algorithm**.

Keywords- Wireless Sensor Networks; Long-thin Networks; Faulty Readings; Network Energy Consumption

I. INTRODUCTION

Wireless sensor network (WSN) consists of hundreds of small nodes that work together to accomplish a network task. Each node includes: a sensor, a processor, communication components (antenna), a small memory, and a source of energy [1]. Due to WSN energy resources limitation, it is required to use algorithm that consumes much less energy in the WSN. Network localisation is one of the important issues in WSNs, which determines the location of a node in the network.

One of the methods in network localisation is that all the network nodes are equipped with devices (i.e. Global Positioning System (GPS)) then the nodes are divided into the two groups: Range-free and Range-base nodes [2, 3]. This method would be very expensive in terms of energy consumption and implementation cost. There have been plenty of researches conducted to establish correct localisation algorithms for different application scenarios [4].

Energy efficiency, algorithm accuracy, and security attacks are considered as the main metrics in network localisation [5]. In terms of the first two metrics, several researches have been conducted. However, the network security metric has just been considered in the last few years [6]. Security attacks in WSNs are divided into the two categories: Internal and External attacks. In the internal attacks, a network node sends wrong information to other nodes in the network, because of being faulty. However, in the external attacks a malicious node sends information to damage network localisation [7-9]. Another important issue that needs to be considered is nodes with faulty readings. In this case, network sensors in specific environments are tending to failure.

Long-Thin Network (LTN) is a specific type of network topology that widely used in wireless sensor applications. The applications 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 the LTN, sensors may form several long backbones, which extend the network to intended coverage areas. A backbone is a linear path which may contain tens or hundreds of network routers [10].

In this paper a new method to control fault tolerance in the LTN is introduced, which is based on Debraj De's localisation error detection algorithm [11-15].

This paper also shows that the distance between two sensors does not completely indicate solidarity between the readings of two sensors. Moreover, if the nearest sensor is faulty, the result of voting by the suspected node is hierarchically corrupted. This problem named Domination Problem (DP). Figure 1 shows the form of a sensor network that is connected to neighbour sensors.

Each link is labelled with a weight that is used in the voting process. Assume that the weight of nodes S2, S3, and S4 are 0.3, 0.4 and 0.9, respectively, and sensor S4 is a faulty sensor. Obviously, reading of node S1 is introduced as a faulty reading, when the method of weighted voting is done. For example: 0.3 * 0.9 * 0.4 * 1 (-1) = -0.2. The positive and negative votes are shown respectively as 1 and -1, as well. As it is stated above, weighted voting method based on distance has several major defects [16, 17]. Based on the above observations, an innovative network voting algorithm is suggested to identify faulty reading of a node by taking the

978-0-7695-4619-3/11 $26.00 © 2011 IEEE
DOI 10.1109/EMS.2011.42

correlation of readings of two nodes and their confidence number.

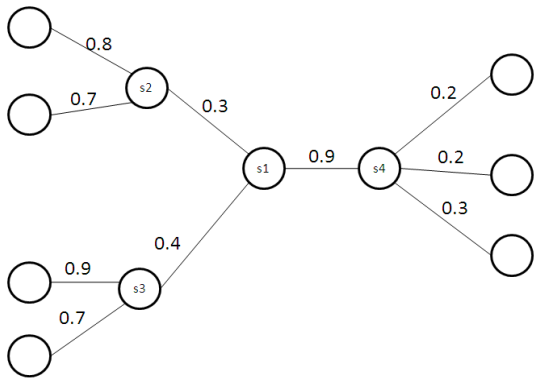

Fig. 1. An illustrative topology of a wireless sensor network

In the proposed algorithm, a weight is given to each node that includes total correlation of the adjacent nodes with confidence coefficient and also sensed value of current time of the sensor. Although this algorithm is complex than the Debraj De algorithm, it is greatly more affordable compared to other algorithms. The Debraj De algorithm uses space between nodes and causes less distance to corrupt voting [18]. In the proposed algorithm, the effect of corrupted nodes in voting is much more reduced to achieve better results. Since, the proposed algorithm is not only based on the calculated correlation, the range of correlation is reduced. Therefore, it reduces calculation time and cost, and improves energy consumption in the network.

The rest of this paper organises as follows. In section II, a new algorithm for fault tolerable deployment is proposed for long-thin wireless sensor networks. Section III shows error detection techniques of localisation, which includes basic concepts of detection of localisation error. In section IV, a new algorithm for localisation error detection in long-thin network is proposed. The proposed algorithm based on correlation to determine faulty readings is discussed in section V. Section VI compares the results of the proposed algorithm with other related work. Finally, section VII concludes this paper.

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

The form of nodes distribution in the long-thin network causes each node to have fewer neighbors. Few neighbors help to have faults in network. Number of neighbors should not be very few, which resulted in compromising the health of network. Long-thin structures are usually used in environments that are included in the restrictions. These restrictions limit the number of neighbors. In this structure, failure of some close together nodes may pull some parts of network into isolation, or in a worse case the entire network may stop working. The proposed structure for LTN is an optimal deployment for the sensor nodes within the LT network, and is useful in most practical applications as shown in Figure 2.

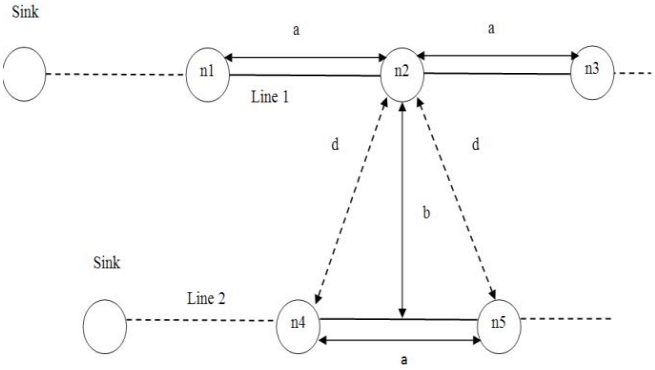

Fig. 2. Proposed fault tolerance substructure for long-thin network

In this infrastructure increasing in number of lines depends on space limitations and other issues. In this case, the number of neighbour nodes is four. This infrastructure is repeated throughout the network. The distance between a node and adjacent nodes on the same line is *a* and distance between two lines is *b* and the distance of node with node in the next line is *d*. By rectangular triangle definition, the following equation is established:

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

The number of neighbours of a node can also increase by increasing the number of parallel lines and changing the value of parameters *a*, *b*.

III. BASIC CONCEPT OF LOCALISATION ERROR DETECTION

The concept of localisation error detection technique is shown in Fig. 3. It is assumed that the sensors are on a XY page.

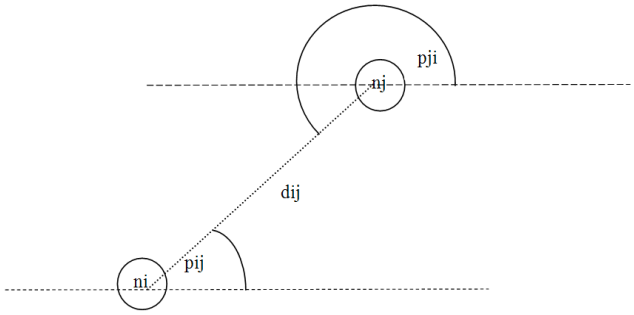

Fig. 3. Basic concept behind localization error detection

Fig. 3 shows that nodes n_i and nj see each other with the angles P_{ij}, P_{ji}. These two nodes have a distance of d_{ij} from each other. Then coordination of n_j to ni is obtained from:

$$X_{ij} = d_{ij} \cdot \cos p_{ij} \text{ and } y_{ij} = d_{ij} \cdot \sin p_{ij} \qquad (2)$$

Similarly, the node coordinates ni to nj comes:

$X_{ji} = d_{ji} . \cos p_{ji}$ and $y_{ji} = d_{ji} . \sin p_{ji}$ (3)

If there is no error in calculation of distance and angles size, then:

$(x_{ji} + x_{ij}) = 0$ and $(y_{ji} + y_{ij}) = 0$ (4)

Therefore, by comparing these values, error in the localisation can be detected.

IV. Basic Concept of Localisation Error

Using static localisation in most long-thin network configurations causes dispose to some types of errors. Fig. 4 shows the Debraj De's algorithm for localisation error detection.

In Fig. 4, the localisation algorithm is illustrated with respect to node N_2. Each node in the network runs the same algorithm as node N_2. The proposed algorithm, which is established based on optimised Debraj De algorithm is described as follows:

- Node N_2 broadcasts a HELLO message or a dummy 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.
- Based on these messages, N_2 calculates relative position of each of its neighbors, using Angle of Arrival (AOA) technique and distance information. Suppose that N_2 calculates its neighbor N_j's relative position as (X_{2j} ,Y_{2j}) from Angle of arrival A_{2j} and distance D_{2j} information. Therefore:

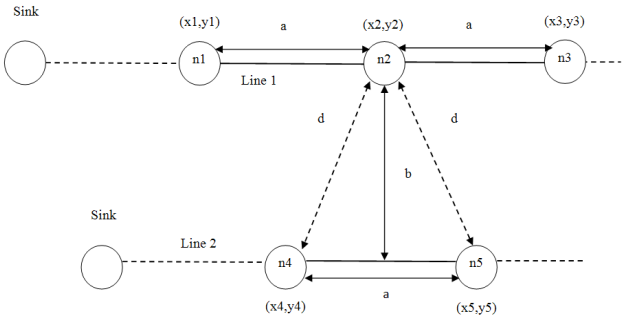

Fig. 4. Localisation error correction on long-thin topology

$X_{2j} = D_{2j} . \cos A_{2j}$

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

- N_2 sends feedback message M_2 with calculated position information (X_{2j}, Y_{2j}) to each specific neighbour N_j. In the other word, every node plays a game with each of its neighbour by exchanging their positions, at which, one can observe the other.
- Node N_2, receives feedback M_2 with information (X_{j2}, Y_{j2}) from each of its neighbour N_j.
- The value of (X_{j2} +X_{2j}) and (Y_{j2} +Y_{2j}) should be zero. But due to different kinds of errors, it would not be zero. Obviously, these values capture all the possible errors that may affect accuracy of localisation.

Regarding detecting faulty or malicious node, if N_2 gets only one message M_2 from N_j, then it saves the relative position information (X_{2j} ,Y_{2j}). Node N_2 calculates (X_{j2} + X_{2j}) and (Y_{j2} + Y_{2j}). If (X_{j2} + X_{2j}) <= $X_{error\text{-}threshold}$ and (Y_{j2} + Y_{2j}) <= $Y_{error\text{-}threshold}$, then N_2 can rely node N_j, and set a confidence level of node N_j to confidence N_j = trust_value, otherwise, N_j can't absolutely rely on accuracy of N_j, so set confidence level confidence N_j= non_trust_value. Typically, trust_value and non_trust_value can be selected as 3 and 1, respectively. In this case, every node decides whether to rely on its neighbour information or not [14]. Also it sets the confidence levels for the neighbours, which is utilised in network detection of faulty readings described in the following section.

V. Proposed Algorithm

A. Correlation

As it stated above, in the most of previous works, the distance between two sensor nodes is considered when the sensor reads correlation. Furthermore, it is possible that the close reading of two sensors would be very different in geometric terms. Therefore, it is important to achieve more accurate correlation between sensor readings for the distance between them.

Suppose that all readings of a S_i sensor include a sequence of readings inside the sliding window Δt. This sequence reading is called Read Vector. Si readings can be expressed as follows:

$b_i(t) = \{x_i(t - \Delta t + 1), x_i(t - \Delta t + 2), \ldots, x_i(t)\}$ (6)

Where $X_i(t)$ is sensed value by S_i in time t. Therefore, similarities between two sensor nodes in terms of reading vectors can be defined due to faulty reading is very different from other ordinary readings in terms of direction and amount. In this paper, the extended version of Jakard algorithm [18] was used as the similarity function. Based on the Jakard algorithm, similarity function for calculating similarity of two sensors S_i, S_j is shown by $Corr_{ij}$ and it is defined as follows:

$$corr_{ij} = \frac{b_i(t).b_j(t)}{||b_i(t)||_2^2 + ||b_j(t)||_2^2 - b_i(t).b_j(t)} \quad (7)$$

$$\text{where} \| b_i(t) \|_2^2 = | x_i(t - \Delta t1) |^2 + \ldots + | x_i(t) |^2 \quad (8)$$

If readings of two sensors are not similar, the value of $Corr_{ij}$ would be close to zero. On the other hand, the value of

$Corr_{ij}$ would be one when the vectors of two-sensor readings are exactly the same.

B. Proposed Algorithm

As a common method to detect faulty reading in the network, voting methods can be used. In the common voting methods correlation parameter is used but this parameter is not enough alone to vote and it requires additional parameters take part in voting. These voting methods are really costly [18].

In Debraj De algorithm, distributed localisation error detection for each node is calculated through the level of confidence of the node. In the Debraj De algorithm, to explore the faulty readings the following formulas are used:

$$w_{ij} = \frac{\text{confidence}_{ij}}{d_{ij}} \tag{9}$$

$$\text{Vote}_i = \sum\nolimits_j (w_{ij} . s_j) \tag{10}$$

In the Debraj De voting algorithm, the space parameter has a source that sometimes causes an error in the voting algorithm. To overcome this problem, in the proposed algorithm, each node weight is shown as follows:

$$W_{ij} = corr_{ij} . confidence_{ij} \tag{11}$$

And voting is done based on:

$$\text{Vote}_i = \sum\nolimits_j (w_{ij} . s_j) \tag{12}$$

In the proposed method, voting on node i is considered. Confidence of I and j is achieved in phase of localisation error detection. $Corr_{ij}$ is also obtains according to the alogorithm initialisation. To prevent faulty nodes from interfering with the voting, confidence and $Corr_{ij}$ are calculated. Suppose that faulty node value is very different from the rest of the node values. In the proposed algorithm for voting node is described as follows:

```
VotingOnNode (Nodei)
{
  for j=1 to 4 {
  Request Node_j to send sensed data within predefined
previous time
  Request Vote from Node_j
  Save vote as V_j
  Save those data to an array by name b_j
  Corr_ij= Call Calculate Correlation(b_i,a)     // a is array of
sensed data by node i
  Vote=V_j*corr_ij*confidence_ij
  }
  If  Vote>0 then
        Return Node is Good
  else
    Return Node is faulty
}
```

Since the $Corr_{ij}$ calculation may consume energy, it simply decreases the range of sampling to acceptable threshold. This algorithm has more computational costs than the Debraj De algorithm and also solves and optimises the problems of Debraj De algorithm. This algorithm has much less computational cost than other voting algorithms which use correlation.

VI. Result Analysis and Comparing

This section compares the results obtained from the proposed algorithm with the other existing algorithms. The existing algorithms can be classified into the two categories. The first batch is weighted algorithms that using of distance inverse as weight. These algorithms have less complexity but they are very vulnerable.

One of the weaknesses of these algorithms is that the high effect of faulty nodes is near to the voting node on the obtained result. The Debraj De algorithm, decreases the vulnerable down to the acceptable level, but the stated problem still exist. The Debraj De algorithm uses the following voting method:

$$w_{ij} = \frac{\text{confidence}_{ij}}{d_{ij}} \tag{13}$$

$$\text{Vote}_i = \sum_j (w_{ij}, s_j) \tag{14}$$

Where D_{ij} is the distance between two nodes. The second category includes algorithms that use the correlation between two nodes as weight. The complexity of these algorithm is high and equals $O(n^3)$. It is easily prove that the proposed algorithm is much more accurate than the algorithms in the first category as shown in Fig. 5.

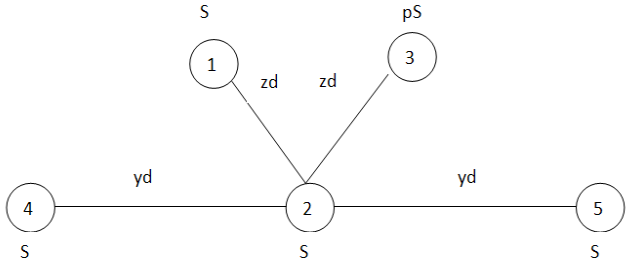

Fig. 5. Evaluation of faulty node effect on the voting

In Fig. 5 S is the value sensed by the nodes and D indicated the distance. Z, Y, and P are constant numbers. According to Debraj De voting algorithm which belongs to the first category, the following relations can be obtained. These relationships are used to detect the faulty readings of node 2. Assume that node 3 is a faulty node that is going to destroy the voting result.

$$w_{12} = \frac{3}{zd} \qquad w_{23} = \frac{1}{zd} \qquad w_{24} = w_{25} = \frac{3}{yd} \tag{15}$$

$$\text{Vote}_i = \frac{3}{zd} * S - \frac{1}{zd} * pS + \frac{6}{yd} * S \tag{16}$$

$$\text{Vote}_i = \frac{S}{d}\left(\frac{3}{z} - \frac{p}{z} + \frac{6}{y}\right) \quad (17)$$

If node 3 try to destroy voting, then:

$$\frac{3}{z} - \frac{p}{z} + \frac{6}{y} < 0$$

$$\xrightarrow{\text{Multiply xy in both sides}}$$

$$3y + py + 6z < 0 \quad (18)$$

WZ can be written instead of y, then:

$$3wz - wpz + 6z < 0 \xrightarrow{\text{Discard z}} 3w + 6 < pw \quad (19)$$

In this case if unequal value is generated, voting is wrong. For example, if w=2 then:

$$3*2z - 2pz + 6z < 0 \longrightarrow 6 < p \quad (20)$$

According to this example, if P is greater than 6 voting is failed. Obviously, it shows a very low accuracy and very high risk of inaccuracy. By decreasing the distance between faulty node and voting node (increasing w), the effect of the faulty node on voting is increased, as shown in the Fig. 6.

To justify the finding and to identify corrupted reading of node N2, the following test case is considered as shown in Fig. 7. After stated calculations and voting procedures, the obtained results show that N2 is faulty. Obviously, this result is wrong and N2 acts correctly. Wrong decision is made due to involvement of the faulty node N3. This problem occurs because of distance parameter selection in the weight of nodes. To overcome this problem, in the proposed algorithm the value of node reading similarity or correlation were used instead of the distance between two nodes due to using correlation parameter alone is not a good criterion for voting.

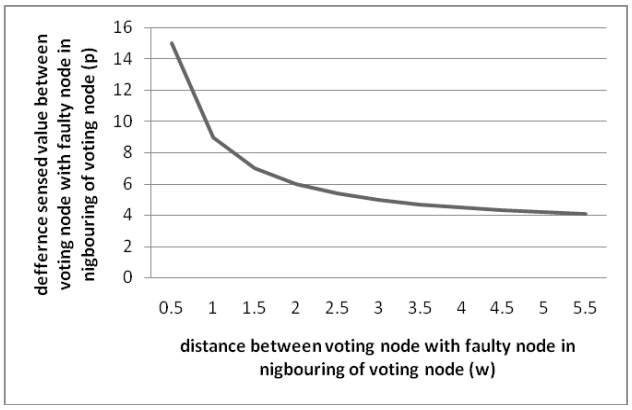

Fig. 6. Effect of faulty nodes on Debraj De voting algorithm

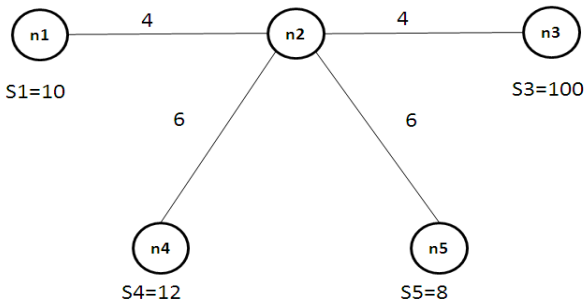

Fig. 7. Wrong decision making due to faulty node N3

It is assumed that:

$$w_{ij} = \frac{\text{confidence}_{ij}}{d_{ij}} \quad (21)$$

$$w_{12} = \frac{3}{4} = 0.75 \qquad w_{23} = \frac{1}{4} = 0.25$$

$$w_{24} = w_{25} = \frac{3}{6} = 0.5 \quad (22)$$

$$\text{Vote}_i = \sum_j (w_{ij}, s_j) \quad (23)$$

$$\text{Vote2} = 0.75 * 10 + 0.5 * 12 + 0.5 * 8 - 100 * 0.25 = 7.5 + 6 + 4 - 25 = -7.5 \quad (24)$$

This algorithm belongs to the second category of voting algorithms. The formula of voting in the proposed algorithm is shown as follows:

$$\text{Vote}_i = \sum_j (\text{corr}_{ij}.\text{confidence}_{ij}.s_j) \quad (25)$$

According to the Fig. 5, node 3 tries to destroy voting result. The confidence number between node 2 and node 3, is 1. Therefore, the following relationships can be inferred. This relation is not true with each value of P. Therefore, it shows that the proposed algorithm can also solve the problem of voting algorithms in the first category, as shown in Fig. 8.

$$\text{corr}_{2,3} = \frac{ps*s}{(pS)^2 + S^2 - pS*S}$$
$$\text{corr}_{2,1} = \text{corr}_{2,4} = \text{corr}_{2,5} = 1 \quad (26)$$

$$\text{vote}_2 = -\frac{p}{p^2+1-p} * pS + 9S < 0 \Rightarrow 9 < \frac{p^2}{p^2+1-p} \quad (27)$$

Assume that the observation period of each node rages t to t+2 as it shown below, and node N1 is less reliable (confidence12 =1), then the voting process is:

N1=101, 100, 99
N2=1.6, 1.5, 1.4
N3=2, 2, 2
N4=1, 1, 1
N5=1.1, 1, 0.9 (28)

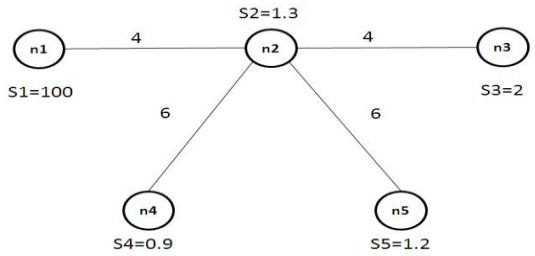

Fig. 8. Solving the problem of voting algorithms in the first category

$$corr_{21} = \frac{161.6+150+138.6}{6.77+10201+10000+9801+450.2} = 0.0152308$$

$$corr_{23} = \frac{9}{9.77} = 0.92119 \qquad corr_{24} = \frac{3}{6.77} = 0.44313$$

$$corr_{25} = \frac{4.52}{7.67} = 0.5893 \qquad (29)$$

$$\text{Vote2} = -0.152308*100*1 + 0.92119*2*3 + 0.44313*3*0.9 + 0.5893*1.2* = -1.52308 + 5.52714 + 1.196451 + 2.12148 >= 0 \qquad (30)$$

According to the obtained result, it shows that the node N2 produces correct data. Also, the proposed algorithm has less complexity than the Debraj De algorithm, thus the energy consumption for calculating faulty reading is improved, as shown in Table 1.

Table 1. Comparing between algorithms using common parameters

Algorithm / Parameter	Weighted Voting with Distance	Debrajde's Algorithm	Weighted Voting with Correlation	Proposed Algorithm
Precision	Less	Medium	Very High	High
Complexity	Less	Less	Very High	Less
Energy consumption	Less	Less	Very High	Less

VII. CONCLUSION

The main aim of this research was to improve detection algorithm for nodes with faulty readings in the network. Besides detection localisation errors, the proposed algorithm can detect the faulty readings. After detecting localisation errors, confidence number of each node was calculated and was used to detect faulty readings using algorithm in the network. Voting can be done through weight based on correlation or distance. The use of voting based on the value of correlation of two nodes is often costly. The Debraj De voting algorithm uses distance parameter in voting; therefore it does not always produce accurate results. The proposed algorithm can overcome the problems of Debraj De voting algorithm. Also it reduces computational complexity of the voting algorithm using correlation between sensed values of nodes.

REFERENCES

[1] R. Stoleru, J. A. Stankovic, and S. Son, "Robust node localzation for wireless sensor networks," In Proc. of EmNets, 2007.

[2] A. Terzis, and A. Anandarajah, and K. Moore, "Slip Surface Localzation in Wireless Sensor Networks for Landslide Prediction," In 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 localzation using stochastic proximity embedding," Computer Communications, pp. 745-755, 33 (6) , 2010.

[4] M. Vecchio, R. López Valcarce and F. Marcelloni, "A two-objective evolutionary approach based on topological constraints for node localzation in wireless sensor networks," Applied Soft Computing, 2011.

[5] N. Patwari and A. O. H. III, "Using proximity and quantized rss for sensor localzation in wireless networks," In Proc. of WSNA, 2003.

[6] A. Ali, T. Collier, L. Girod, K. Yao, D. T. Blumstein, and C. E. Taylor, "An empirical study of collaborative acoustic source localzation," In Proc. of IPSN, 2007.

[7] A. Srinivasan and Wu J., "A Survey on Secure Localzation in Wireless Sensor Networks,"

[8] Encyclopedia of Wireless and Mobile Communications, CRC Press, Taylor and Francis Group, 2007.

[9] Chen H., Lou W., Ma J., and Wang Z., "TSCD: A Novel Secure Localzation Approach for Wireless Sensor Networks," In Proc. of 2nd International Conference on Sensor Technologies and Applications (SensorComm), pp. 661-666, August, 2008.

[10] F. Caballero, L. Merino, P. Gil, I. Maza, A. Ollero, "A probabilistic framework for entire WSN localzation using a mobile robot," Robotics and Autonomous Systems, pp. 798-806, 2008.

[11] H. Aksu, D. Aksoy, I. Korpeoglu, "A study of localzation metrics: Evaluation of position errors in wireless sensor networks," Computer Networks, In Press, Corrected Proof, July 2011.

[12] Gustav J. Jordt, Rusty O. Baldwin, and John F. Raquet, "Barry E. Mullins, Energy cost and error performance of range-aware, anchor-free localzation algorithms," Ad Hoc Networks, vol. 6, pp. 539–559, 2008.

[13] E. Elnahrawy, and B. Nath, "Poster Abstract: Online Data Cleaning in Wireless Sensor Networks," In Proc. of 1st International conference on Embedded networked sensor systems, pp. 294-295, 2003.

[14] D. De, "A distributed algorithm for localzation error detection-correction, use in in-network faulty reading detection: applicability in long-thin wireless sensor networks," In Proc. of the IEEE Wireless Communications and Networking Conference, pp. 1-6, April 2009.

[15] 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.

[16] T. Sun, L.J. Chen, C.C. Han, and M. Gerla, "Reliable Sensor Networks for Planet Exploration," In Proc. of the IEEE International Conference On Networking, Sensing and Control (ICNSC), pp. 816-821, 2005.

[17] X. Xiao, W. Peng, C. Hung, and W. Lee, "Using SensorRanks for In-Network Detection of Faulty Readings in Wireless Sensor Networks," In Proc. of MobiDE, pp. 714-721, 2007.

[18] A. Strehl, and J. Ghosh, and R. Mooney, "Impact of similarity measures on web-page clustering," In Proc. 7th National Conference on Artificial Intelligence: Workshop of Artificial Intelligence for Web Search (AAAI), pp. 58-64. July 2000.