Faulty Node Detection in Wireless Sensor Networks Using Cluster

Srikanta Kumar Sahoo

Abstract- Since the accuracy of data is important to the whole system's performance, detecting nodes with faulty readings is an essential issue in network management. Removing nodes with faulty readings from a system or replacing them with good ones improve the whole system's performance and at the same time prolong the lifetime of the network. In general, wireless sensor nodes may experience two types of faults that would lead to the degradation of performance. One type is function fault, which typically results in the crash of individual nodes, packet loss, routing failure or network partition. The other type of error is data fault, in which a node behaves normally in all aspects except for its sensing results, leading to either significant biased or random errors.

Index Terms- Wireless Sensor Networks, Data Fault Detection, Functional Fault.

Introduction

Wireless Sensor Networks have emerged as an important new area in wireless technology. A wireless network consisting of tiny devices which monitor physical or environmental conditions such as temperature, pressure, motion or pollutants etc. at different areas. Such sensor networks are expected to be widely deployed in a vast variety of environments for commercial, civil, and military applications such as surveillance, vehicle tracking, climate and habitat monitoring, intelligence, medical, and acoustic data gathering. The key limitations of wireless sensor networks are the storage, power and processing. These limitations and the specific architecture of sensor nodes call for energy efficient and secure communication protocols. The key challenge in sensor network is to maximize the lifetime of sensor nodes due to the fact that it is not feasible to replace the batteries of thousands of sensor nodes. Therefore, computational operations of nodes and communication protocols must be made as energy efficient as possible. Since the accuracy of data is important to the whole system's performance, detecting nodes with faulty readings is an essential issue in network management. The accuracy of individual node's readings is crucial; the readings of sensor nodes must be accurate to avoid false alarms and missed detections. Some applications are designed to be fault tolerant to some extent, removing nodes with faulty readings from a system with some redundancy or replacing them with good ones can still significantly improve the whole system's performance and at the same time prolong the lifetime of the network. To conduct such after deployment maintenance (e.g., remove and replace), it is essential to investigate methods for detecting faulty nodes

Wireless Sensor Network

Wireless sensor networks are potentially one of the most important technologies of this century. Recent advancement in wireless communications and electronics has enabled the development of low-cost, low-power, multifunctional miniature devices for use in remote sensing applications. The combination of these factors has improved the viability of utilizing a sensor network consisting of a large number of intelligent sensors, enabling the collection, processing analysis and dissemination of valuable information gathered in a variety of environments. A sensor network is composed of a large number of sensor nodes which consist of sensing, data processing and communication capabilities.

The fundamental objectives for sensor networks are reliability, accuracy, flexibility, cost effectiveness and ease of deployment.

Sensor Faults

In this chapter, we will review some of the commonly observed sensor faults . Faults in sensor data can occur for many reasons. The first source stems from unpredictable environmental conditions, which can often cause sensors to behave erratically. Factors such as extreme temperatures or precipitation can affect sensor performance. During a deployment, sensors can be displaced or change orientation. This can be caused, for example, by animals or humans or wind or if they are deployed in a body of water that freezes. Another type of fault occurs when the environmental conditions travel outside the range of values the sensor is able to detect. In this case, the sensor becomes saturated and is unable to report the true readings.

In the following taxonomy, we discuss both the manifestation of faults in the data as well as specific reasons that different types of faults occur. After the fault types are introduced, we discuss the problem formulation and assumptions that will be made throughout this thesis.

Clustering in WSN

It is widely accepted that the energy consumed in one bit of data transfer can be used to perform a large number of arithmetic operations in the sensor processor. Moreover in a densely deployed sensor network the physical environment would produce very similar data in near-by sensor nodes and transmitting such data is more or less redundant. Therefore, all these facts encourage using some kind of grouping of nodes such that data from sensor nodes of a group can be combined or compressed together in an intelligent way and transmit only compact data. This can not only reduce the global data to be transmitted and localized most traffic to within each individual group, but reduces the traffic and hence contention in a wireless sensor network. This process of grouping of sensor nodes in a densely deployed large-scale sensor network is known as clustering. The intelligent way to combined and compress the data belonging to a single cluster is known as data aggregation.

There are some issues involved with the process of clustering in a wireless sensor network [12]. First issue is, how many clusters should be formed that could optimize some performance parameter [14]. Second could be how many nodes should be taken in to a single cluster. Third important issue is the selection procedure of cluster-head in a cluster. Another issue that has been focused in many research papers is to introduce heterogeneity in the network [17]. It means that user can put some more powerful nodes, in terms of energy, in the network which can act as a cluster-head and other simple node work as cluster-member only. Considering the above issues, many protocols have been proposed which deals with each individual issue.

Motivation of the work

Sensor nodes are widely used in surveillance, vehicle tracking, climate and habitat monitoring, intelligence, medical, and acoustic data gathering. The accuracy of data is important to the whole system's performance, detecting nodes with faulty readings is an essential issue in network management. The work in this dissertation is motivated by the problem of detecting the faulty sensors nodes in the WSN (Wireless Sensor Network).

Objective of the Work

Motivated by the need of a fault detection algorithm for WSN (Wireless Sensor Network), the objective of this work is given as follows:

(a) To propose a frame work for fault detection in WSN.
(b) To propose a faulty node detection mechanism.
(c) To propose a clustering approach to devise an efficient fault detection algorithm.
(d) To study and validate the performance of the proposed fault detection algorithm through simulation using MATLAB.
(e) To compare the proposed approach with the existing approach.

LITERATURE SURVEY

Review of the literature on fault detection in sensor networks

The approach presented in [13] describes faulty sensor nodes are identified based on comparisons between neighboring nodes and dissemination of the decision made at each node. Nodes with malfunctioning sensors are allowed to act as a communication node for routing, but they are logically isolated from the network as far as fault detection is concerned. It employs local comparisons of sensed data between neighbors and dissemination of the test results to enhance the accuracy of diagnosis. Transient faults in communication and sensor reading are tolerated by using time redundancy. Faulty nodes are isolated by correctly identifying fault-free nodes. Both the network connectivity and accuracy of diagnosis are taken into account since fault-free nodes isolated might be of little or no use even if they are determined to be fault-free, unless they can participate in the network via intermediate communication nodes with faulty sensors.

Sensor faults have been studied extensively in [21] in the context of fault diagnosis for systems. The problem of tolerating and modeling sensor failures was also studied in [Mar90]. However, studying faults in wireless sensing systems differs from the issues studied in these references in several ways that make the problem more difficult. The first issue is that sensor networks may involve many more sensors over larger areas. Second, the phenomena being observed by sensor networks are not well controlled or measured in most cases. Instead, they can be very heterogeneous, resulting in higher uncertainty when modeling sensor behavior and sensor faults.

The distributed fault detection scheme have been studied in [12], it checks out the failed nodes by exchanging data and mutually testing among neighbor nodes in the network, but the fault detection accuracy of a DFD (Distributed Fault Detection) scheme would decrease rapidly when the number of neighbor nodes to be diagnosed is small and the node's failure ratio is high.

The author of this[15] paper uses a heartbeat based testing mechanism to detect failure in each cluster and take the advantage of cluster based architecture to forward the failure report to other cluster and their respective members.

The work in [14] use a cluster-based communication architecture to permit the FDS (Failure Detection Service) to be implemented in a distributed manner via intra-cluster heartbeat diffusion and to allow a failure report to be forwarded across clusters through the upper layer of the communication hierarchy. In doing so, we extensively exploit the message redundancy that is inherent in ad hoc wireless settings to mitigate the effects of message loss on the accuracy and completeness properties of failure detection.

The work in [22] identifies a small number of sensor faults observed in real deployments. The faults are briefly defined, and different methods of offline detection of these faults are examined. The performance of each of these fault detection methods are analyzed using three real deployments.

PROPOSED SYSTEM

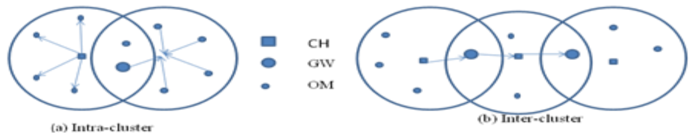
System and Fault Model

We assume that the wireless ad hoc network is a large connected network in which there are totally N sensor nodes denoted by 1, 2, 3… N. The nodes are distributed randomly in some physical domain and become stationary after deployment. The transmission range for each node is fixed and link between two hosts is bi-directional. If host u is in the transmission range of another host v, then there must be a link between the two. The system can be modeled as a communication graph G = {V, E}, where V= {1, 2….N}, and E= {(v1, v2): v1 is in transmission range of v2 and vice versa}.

A cluster is a unit disk with a radius equal to the center node's transmission range. As a result, any non-center nodes in a cluster are one-hop neighbors of the center node. The center node is called the cluster head (CH), while a node that is a one hop neighbor of the CHs of two different clusters can become the gateway (GW) node (see Figure 1) . After the autonomous cluster formation, only CH and GW node, which are elected in a fully distributed fashion, participate in the inter-cluster communication (see Figure 1(b)), while ordinary members (OMs) in each cluster talk only to their CHs (and to other members when necessary).

```
For any unselected node v
{
    If ((node v is an indispensable node) || (node v is the only node with highest quality Qv among unselected neighbor) || (among unselected neighbor with same quality node v is with the smallest ID)
    {
        Update status to selected;
        Regard itself as a CH;
        Send an invite packet, invite (v) to all neighbors ;
    }
    On receiving an invite packet from neighboring node v
    If (node u is an indispensable node)
        Discard this packet;
    Else
    {
        Regards itself as an ordinary node;
        Updates status to selected;
        Sends a join packet, join (u,v) to join the cluster constructed by v;
        If (more than one such packets are received)
            Join the one with smallest ID;
        Else
            Joins sender with largest logical degree;
        Regards itself as a gateway node;
    }
    On receiving a join packet sent from neighboring node u decreases the logical degree by 1;
}
}
```

The proposed system is not fully distributed. The total number of nodes is equally divided into a number of clusters. Each cluster has a CH and there is a GW node between two clusters to forward the message from one cluster to another. The cluster is controlled by the CH. The fault is detected by the CH in each cluster and the message is forwarded to all nodes of the cluster and also forwarded to other CH . All the clusters are operating simultaneously.



(a) Intra-cluster (b) Inter-cluster

1: Intra-Cluster and Inter-Cluster Communication

In the fault detection of wireless sensor networks, we assume that all the sensor nodes have the same transmission range. Sensor nodes can be randomly deployed or placed in predetermined locations. Nodes with faulty sensors and permanent communication faults are to be identified. Sensor nodes which generate incorrect sensing data or fail in communication intermittently are treated as usable nodes, and thus are diagnosed as fault-free. Sensor nodes with malfunctioning sensors could participate in the network operation since they are still capable of routing information. Only those sensor nodes with a permanent fault in communication (including lack of power) are detected and this information is disseminated throughout the network and removed from the network.

Algorithm for cluster formation

This section describes the algorithm for cluster formation in the proposed system model. The algorithm is given in a table.

Table 1: Algorithm for Cluster formation

The system model uses an existing method FIND (Faulty Node Detection) to detect nodes with data faults [11]. After the nodes in a network detect a natural event, FIND ranks the nodes based on their sensing readings as well as their physical distances from the event. A node is considered faulty if there is a significant mismatch between the sensor data rank and the distance rank.

There are three stages in the proposed system

(1) Map Division
(2) Detection Sequence Mapping
(3) Fault Detection

Map Division: The first stage of our proposed system is **map division**, in which the map is divided into a number of subareas named faces based on the topology of the network. Each face is uniquely identified by a distance sequence, which is denoted as a sequence of sensor node IDs (e.g., 2-1-3-4-5). Within a distance sequence, the IDs are sorted in order of nodes' distances from an arbitrary point within this face. Map division can be pre-computed before detecting faulty nodes such that a number of distance sequences can be obtained.

For two sensor nodes 1 and 2, the perpendicular bisector Div (1,2) divides the area into two subareas. For any position point below Div (1,2), node 1 is closer than node 2, so the distance sequence is 1-2. For any position point above Div (1,2), node 2 is closer than node 1, so the distance sequence is 2-1.

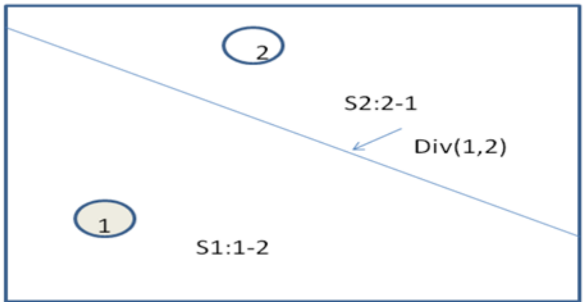


Figure 2: Map Division

Detection Sequence Mapping: The second stage is **detection sequence mapping**, a number of events appear in the monitored region and are detected by the sensor nodes. For a single event, the sensing result of each node (e.g., the received signal strength or time-of-arrival) varies depending on how far the node is away from the event. A detected sequence is then obtained by ordering the sensing results of all the sensor nodes. Without knowing the location of the event, this detected sequence is mapped with one of the distance sequences corresponding to the face in which the event most likely takes place, yielding an estimated sequence. The same mapping process is repeated for all the events such that an estimated sequence is obtained for every detected sequence after this stage.

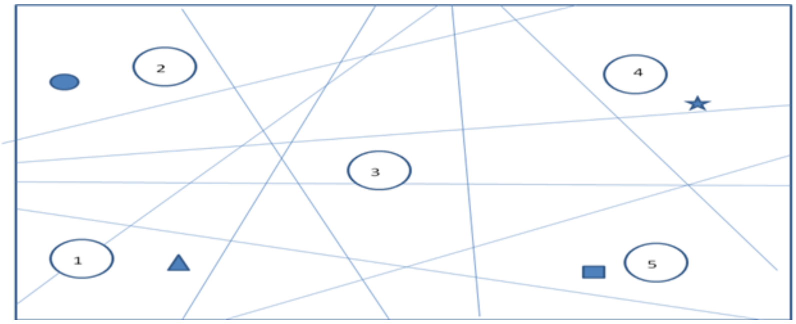


Figure 3: Map Division with Events and Nodes

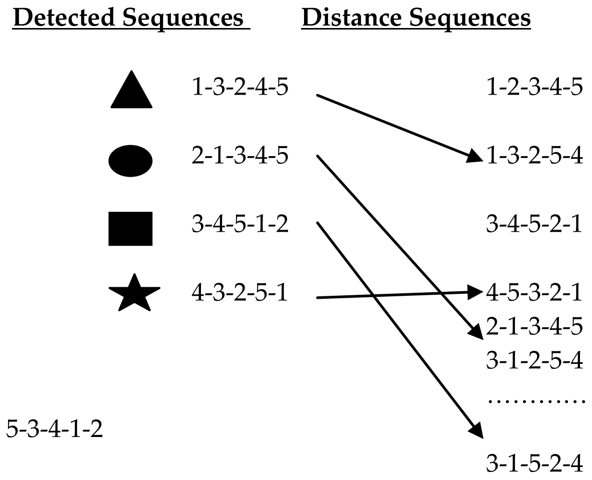


Figure 4: Detection Sequence Mapping

Let the N sensor nodes divide the map into M faces, identified by a set of distance sequences
$V = \{S_1, S_2, \cdots, S_M\}$.

Suppose S′ is a detected sequence from a single event and S is the distance sequence corresponding to the face where the event takes place.

$\{A_1, A_2, \cdots, A_M\}$ denotes the size of the M faces, the probability that an event takes places within the i^{th} face can be computed as the following:

$$\Pr(S = S_i) = \Pr(S_i) = A_i / \textstyle\sum_{j=1}^{M} A_j \quad , 1 \le i \le M \qquad (1)$$

S can be estimated by the method of Maximum A Posteriori (MAP) estimation as

$$S''_{MAP}(S') = \text{args}\ \max \Pr(S \mid S') \quad S_i = S$$

$$= \arg\max \quad \Pr(S' \mid S_i)\Pr(S_i) / \textstyle\sum_{k=1}^{M} \Pr(S' \mid S_k)\Pr(S_k) \quad \text{where} \quad S_i \in V$$

$$= \arg\max \quad \Pr(S' \mid S_i)\Pr(S_i) \qquad \text{where} \quad S_i \in V \qquad (2)$$

Fault Detection: The third stage is **fault detection using Ranking Differences**, after a sufficient number of mappings becomes available; a blacklist is then obtained by analyzing the inconsistencies between the detected sequences and the estimated sequences.

Average Ranking Differences

We can identify faulty nodes by using the average ranking difference based on the following two cases.

Case 1. A node with a larger average ranking difference has a higher probability of being a faulty node.

The ranking difference $d_i(K)$ for node K in sample i is given in the following equation:

$d_i(K) = |R(S_{i1}, K) - R(S_{i2}, K)| \quad 1 \leq K \leq N$

Where R (*, K) denotes the ranking of node K in sequence *,S_{i1} is estimated sequence & S_{i2} is the detected sequence. Then, the average ranking difference of node K in the n samples (denoted as D(K)) is computed by averaging $d_i(K)$:

$$D(K) = 1/n \sum_{i=1}^{n} di(K) \qquad (3)$$

D (K) is also known as the sample mean of the ranking differences.

Case 2. The majority of faulty nodes can be obtained by selecting nodes whose ranking differences are above a lower bound. A node q is faulty if its average ranking difference D (q) is greater than a bound B given by

$$B = \frac{Ne}{N-1}(\mu_e + N_e)$$

Where N_e is the number of faulty nodes in an N-node network, μ_e is the arithmetic mean of the average ranking difference of faulty nodes, and D (q) is calculated under a sufficient large sample size.

Fault Detection Algorithm

This section describes the algorithm for faulty sensor node detection in the proposed system model. The different notations used in this algorithm are presented in a table.

μ_e :	arithmetic mean of average ranking difference of faulty nodes
N_e :	number of faulty nodes
B :	bounded value
J :	variable used in for……end loop
N :	total number of nodes
$D(n_j)$:	mean of ranking difference

Table 2: Meaning of Different Terms

Initially the blacklist is empty (Line 1) and N_e, μ_e and B are set to 0 (Line 2). Starting from n_1 (which is the node with the largest average ranking difference) and check the nodes one by one (Line 3) until the ranking difference of a node is no greater than B (Line 4 and 5). Specifically, a node can be added into the blacklist if and only if its ranking difference is greater than B. After adding a new node into the blacklist, N_e, μ_e and *B* are updated (Line 7 to 9). Then, an estimation of faulty nodes $\{n_1, \cdots, n_K\}$ can be obtained if this loop breaks at node n_{K+1}.

```
Input: Sorted node sequence $n_1, \cdots, n_N$, average ranking differences D
Output: The Blacklist $\{n_1, \cdots, n_K\}$

1: Blacklist ← $\emptyset$ // Initialization
2: $\mu_e \leftarrow 0$, $N_e \leftarrow 0$, $B \leftarrow 0$
3: for j ←1 to N do
4: if $D(n_j) \leq B$ then
5: Break
6: else
7: $\mu_e \leftarrow \frac{(\mu e Ne + D(nj))}{Ne+1}$, $Ne \leftarrow Ne + 1$
8: $B \leftarrow \frac{Ne}{N-1}(\mu_e + N_e)$
9: Blacklist ←Blacklist + $\{n_j\}$
10: end if
11: end for
12: Return Blacklist
```

Table 3: Algorithm for Fault Detection

Example:

In the following example we consider five nodes 1, 2,3,4,5 and four events. The detected sequences and estimated sequences for different events are calculated.

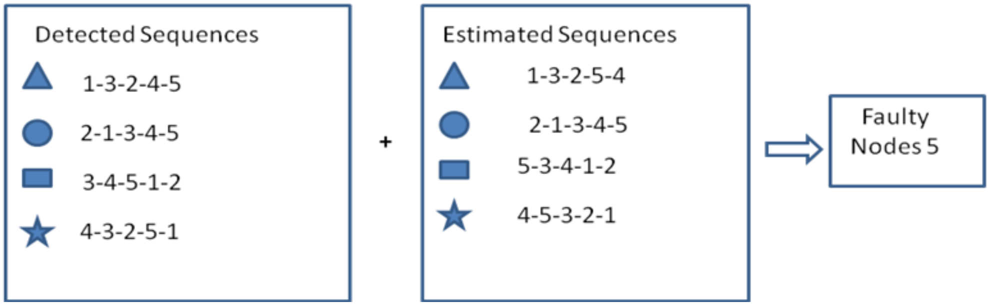
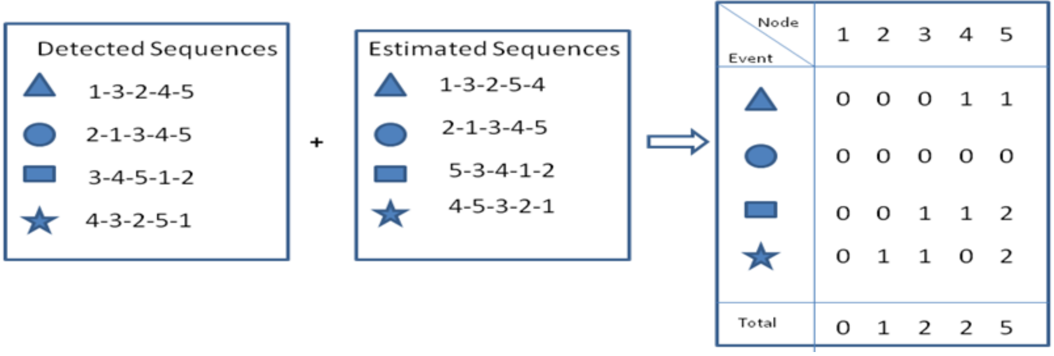


Figure 5: Fault Detection



Node / Event	1	2	3	4	5
▲	0	0	0	1	1
●	0	0	0	0	0
■	0	0	1	1	2
★	0	1	1	0	2
Total	0	1	2	2	5

Figure 6: Fault Detection Using Ranking Differences

In the above Figure 6, four detected sequences are mapped with their estimated sequences and the ranking differences can be computed by comparing the rankings of each node in the two sequences. Let us take the square event as an example. For the square event, the detected sequence is 3-4-5-1-2 and the estimated sequence is 5-3-4-1-2, where nodes 1 and 2 have the same rankings such that their ranking differences are 0. Node 3 and 4 both shift by one and their ranking differences are 1. Node 5 ranks the first in estimated sequence and the third in detected sequence, and thus its ranking difference is 2. The ranking differences of all the nodes in all events are shown in the above figure. In this example, node 5 is a faulty node, with all its readings lower than expected. First, as a faulty node, node 5 has a non-zero ranking difference in most events due to its faulty reading. Second, node 5 also changes some normal nodes' rankings. However, a faulty node at most changes normal nodes' rankings by 1. Third, for different events, the sets of normal nodes whose rankings are changed by node 5 are different. The total (or average) ranking difference of node 5 is the largest and is the faulty node.

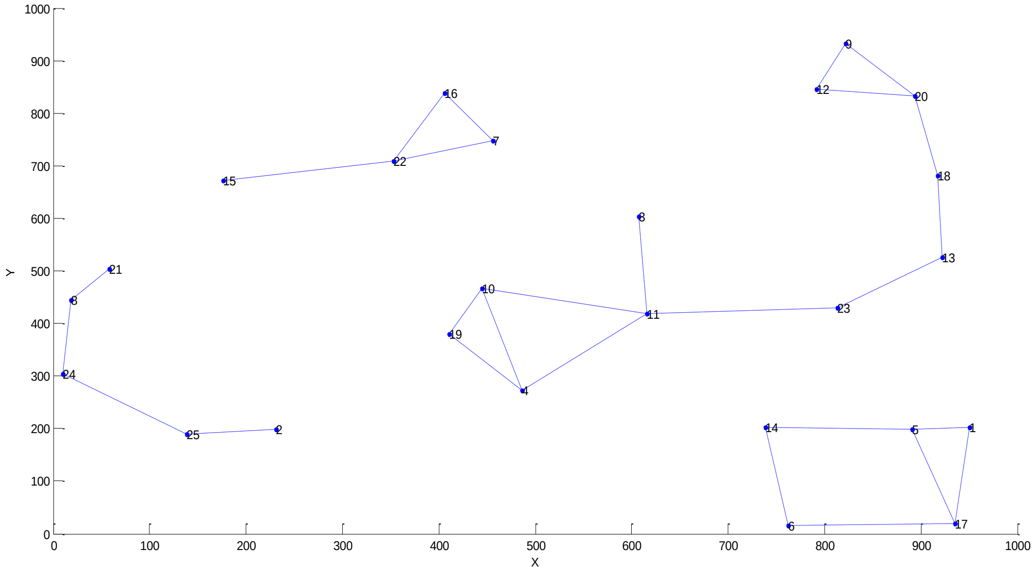
SIMULATION & RESULT ANALYSIS

Simulation Model and Parameters

We implemented the proposed system using MATLAB. We setup a network having 25 sensor nodes and different numbers of events. All the nodes are randomly distributed throughout the terrain and divided into clusters. Each cluster having equal number of nodes, a CH and events. Broadcasting events, identified by their unique event ids. After receiving a broadcasting packet, nodes measure the RSS and record it together with event id. The number of events generated varies from 19 to 49. After recording the RSS of all the events, we randomly select 1 to 5 nodes and inject errors into their readings, corresponding to a 4% to 20% defective rate.

We evaluate the performance of proposed system in two scenarios. In the first scenario, an accurate A (defective rate) is assumed so that the first $A*N$ nodes with the largest ranking differences are selected as faulty nodes. This scenario is known as **A-based detection** (or A −detection for short). A is the defective rate i.e

A = (no. of faulty nodes / total no. of nodes) x100.



In the second scenario, detection algorithm is used for selecting faulty nodes whose ranking differences are greater than B (bounded value). This scenario is known as ***B*-based detection** (or *B*-detection for short).

Two metrics are used for evaluating the proposed system**: false negative rate and false positive rate**. The former is defined as the proportion of faulty nodes that are reported as normal, which is also known as miss detection rate. The latter is defined as the proportion of normal nodes that are reported as faulty, which is also known as the false alarm rate.

from the so-called cell arrays to the definition of classes in object oriented programming.

MATLAB syntax remains very simple and MATLAB programs can be written far more easily than programs in other high level languages or computer algebra programs. A command interface created for interactive management without much ado, plus a simple integration of particular functions, programs, and libraries supports the operation of this software tool. This also makes it possible to learn MATLAB rapidly.

MATLAB is not just a numerical tool for evaluation of formulas, but is also an independent programming language capable of treating complex problems and is equipped with all the essential constructs of a higher programming language. Since the MATLAB command interface involves a so-called interpreter and MATLAB is an interpreter language, all commands can be carried out directly. This makes the testing of particular programs much easier.

MATLAB 7 is equipped with a very well conceived editor with debugging functionality, which makes the writing and error analysis of large MATLAB programs even easier.

The last major advantage is the interaction with the special toolbox Simulink, This is a tool for constructing simulation programs based on a graphical interface in a way similar to block diagrams. The simulation runs under MATLAB and an easy interconnection between MATLAB and Simulink is ensured.

Figure 7 and 8 depicts simulation scenario for 25 nodes with 1000x1000 area and cluster formation.

Figure 7: Randomly Placed Sensor Nodes

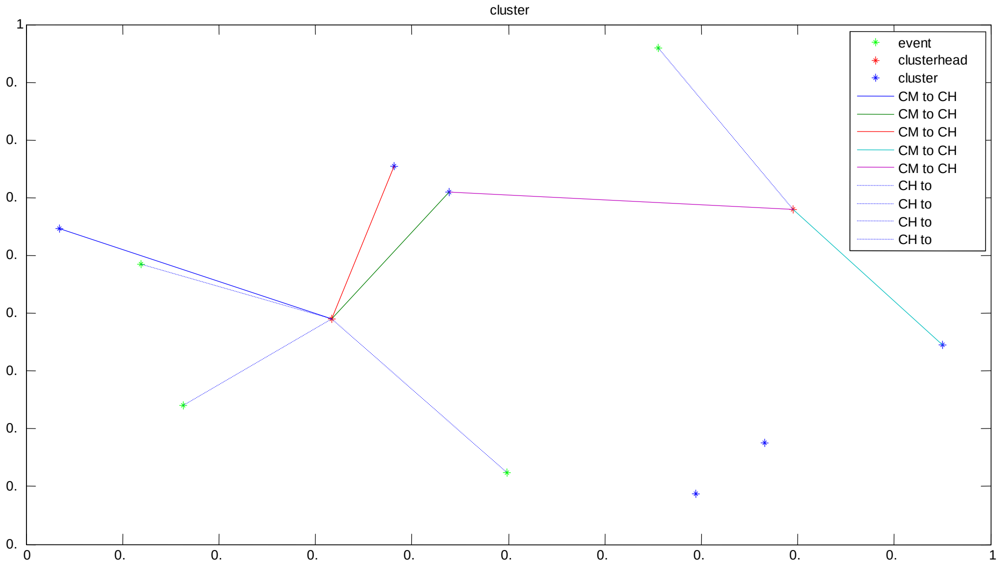


Figure 7: Cluster Formation

Figure 8 shows the simulation result of false negative rate for 19 events based on A detection .

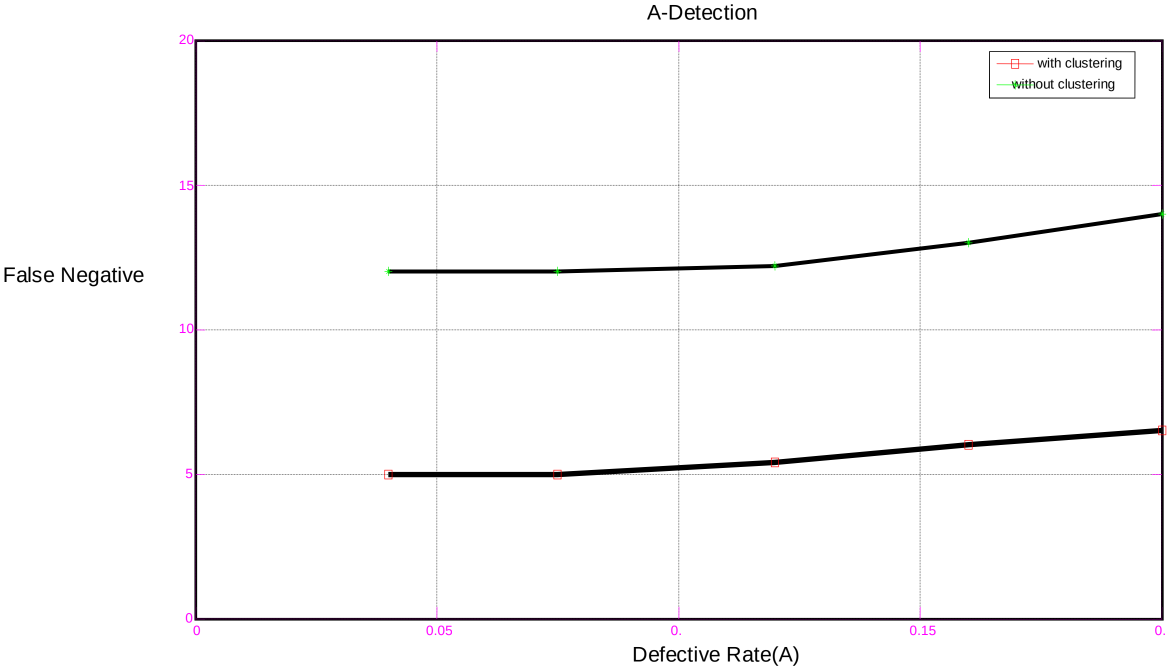


Figure 8: False Negative Rate (19 Events, A Detection)

Figure 9 shows the simulation result of false negative rate for 29 events based on A detection.

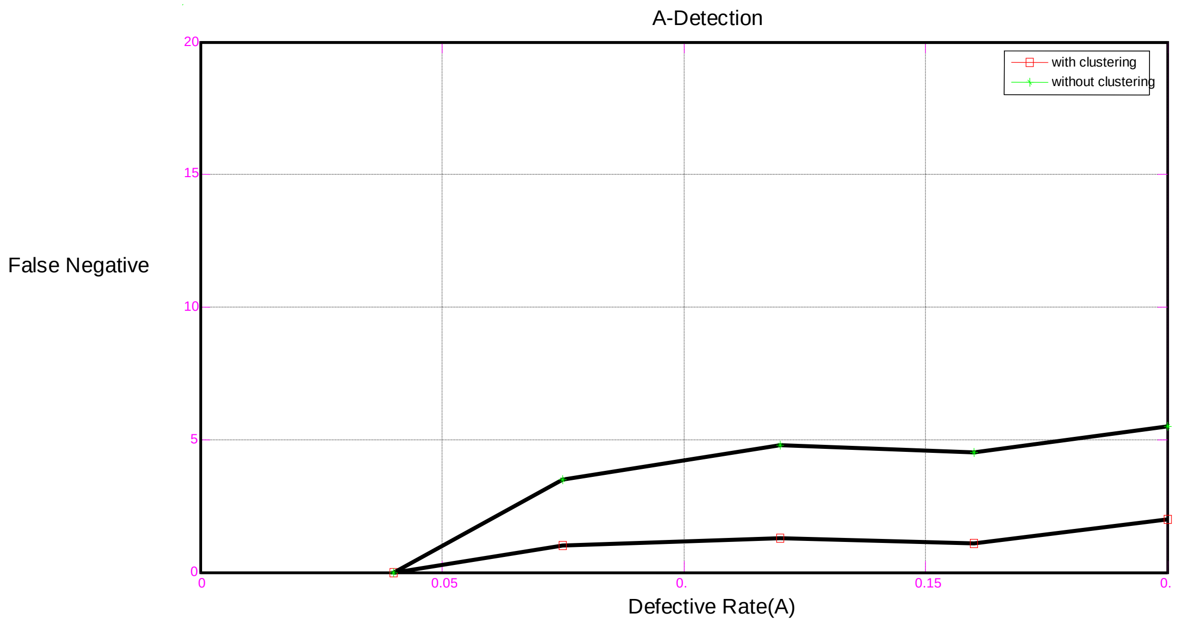


Figure 8 False Negative Rate (29 Events, A Detection)

Figure 9 shows the simulation result of false negative rate for 39 events based on A detection .

Events, B Detection)

tection(19,29,39,49 events respectively).

CONCLUSION & FUTURE WORK

Future Work

It can be possible extension of present work. The work in this dissertation is motivated by the problem of fault detection for WSN. We have presented a promising framework for fault detection using FIND to model faults and monitor the data to detect the occurrence of these faults. It is worth emphasizing that fault detection is not the end goal in sensor networks. The primary goal is to obtain clean data which can be analyzed to achieve the scientific objectives. Fault detection methods are used to make this goal possible. In this context, the future work is that persistent faults will be diagnosed as quickly as possible so that large amounts of data are not wasted.

False Negative Rate

REFERENCES

[1] R. Szewczyk, A. Mainwaring, J. Anderson, and D. Culler. An Analysis of a Large Scale Habit Monitoring Application. In SenSys'04, 2004.

[2] G. Tolle, J. Polastre, R. Szewczyk, N. Turner, K. Tu, S. Burgess, D. Gay, P. Buonadonna, W. Hong, T. Dawson, and D. Culler. A Macroscope in the Redwoods. In SenSys'05, November 2005.

[3] W. Su Y. Sankarasubramaniam E. Cayirci Akyildiz, I.F. A survey on sensor networks. IEEE Communications Magazine, pages 102{114, 2002}.

[4] Kumar.S.P. Chee-Yee Chong. Sensor networks: Evolution, opportunities, and challenges. Proc IEEE, August 2003.

[5] Ismail H. Kasimoglui Ian .F. Akyildiz. Wireless sensor and actor :research challenges. (Elsevier) Journal, 2(38):351{367, 2004.

[6] Sarika Agarwal Leszek Lilien Maleq Khan, Bharat Bhargava and Pankaj. Self-configuring node clusters, data aggregation, and security in microsensor networks. Department of Management Information Systems Krannert Graduate School of Management Purdue University, West Lafayette, (IN 47907), 2007. pankaj@mgmt.purdue.edu.

[7] Sungha Pete Kim Bo-Cheng Charles Lai, David D. Hwang. Reducing radio energy consumption of key management protocols for wireless sensor networks. ACM 1-58113-929-2/04/0008, pages 9{11, August 2004.

[8] Ismail H. Kasimoglui Ian .F. Akyildiz. Wireless sensor and actor :research challenges. (Elsevier) Journal, 2(38):351{367, 2004.

[9] K. Delin, S. Jackson, and R. Some, “Sensor Webs,” NASA Tech Briefs Journal, vol. 23, p. 80, Oct. 1999.

[10] J. Rabaey et al., “PicoRadios for wireless sensor networks — the next challenge in ultra- low power design,” in Proc. IEEE Int. Solid-State Circuits Conference (ISSCC), (San Francisco, CA), Feb. 2002.

[11] Shuo Guo, ZiguoZhong, Tian He “FIND: Faulty Node detection for Wireless Sensor Networks”, in SenSys’09, November 4-6, 2009, Berkeley, CA, USA.

[12] Peng Jiang “A New Method for Node Fault Detection in Wireless Sensor Networks” ,Sensors 2009.

[13] Myeong-Hyeon Lee, Yoon-Hwa Choi “Fault detection of wireless sensor networks” ,(Elsevier) Computer Communications 31 (2008) 3469–3475.

[14] Ann T. Tai Kam S. Tso, William H. Sanders “Cluster-Based Failure Detection Service for Large-Scale Ad HocWireless Network Applications” ,Proceedings of the 2004,IEEE.

[15] M.Yu, H.Mokhtar, and M.Merabti “A Survey on Fault Management in Wireless Sensor Networks”.

[16] Sumit Gupta, Rong Zheng, Albert M. K. Cheng “ANDES: an Anomaly Detection System for Wireless Sensor Networks”, 2007 IEEE.

[17] Abolfazl Akbari and Neda Beikmahdavi and Bahman Shahamat Naderi “Cluster-based and cellular approach to fault detection and recovery in wireless sensor network”, IJWMN February 2010.

[18] Sheela D. Nair “Finding Fault: Anomaly Detection for Embedded Networked Sensing ”.

[19] Pabitra Mohan Khilar, Jitendra Kumar Singh, Sudipta Mahapatra “Design and Evaluation of a Failure Detection Algorithm for Large Scale Ad Hoc Networks Using Cluster Based Approach “,2008 IEEE.

[20] F.L.Lewis “ Wireless Sensor Networks “, To appear in Smart Environments: Technologies, Protocols, and Applications ed. D.J. Cook and S.K. Das, John Wiley, New York, 2004.

[21] R. Isermann. Fault-Diagnosis Sytems: An Introduction from Fault Detection to Fault Tolerance. Springer, 2005.

[22] A. Sharma, L. Golubchik, and R. Govindan. \On the Prevalence of Sensor Faults in Real World Deployments." In IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON).

AUTHORS

- ***Srikanta Kumar Sahoo,*** *is currently pursuing masters degree program in computer science and engineering department in* ***Biju Patnaik University of Technolog,****Odisha,India,Mobile No-+919776293140,Email-srikant.sahoo@live.com.*