Self-Diagnosis and Healing of System Failures in Immensely Colossal Wireless Sensor Networks

Adithya Raj, Sherin Koshy

Abstract— **Typically sensor nodes in sizably voluminous wireless sensor networks are analysed at the sink node where all nodes are sending data. Self diagnosis is the method which plays a vital role in the efficiency of the network by incorporating all the nodes in a WSN to participate in the fault analysis process. By attention to the node's circumscribed energy and most of their energy is consumed for communicating to other nodes or sink, the best way to use energy optimally in nodes is clustering. This makes the sensor node locate a faulty node in a more expeditious way. Sensor nodes customarily transmit the sensed data by IEEE 802.15.4 standard in which the frequency band is divided into different slots. So whenever the link quality is found weak, the sensor node can shift the operating channel to another one and it will improve the performance of wireless link. For this, the sensor node should be able to shift to different predefined channels depending upon the link quality. And this can be achieved by a technique called software defined radio (SDR). So by implementing this technique to the self diagnosing method the performance of WSN can be amended. The sensor nodes will act as self diagnosing as well as self healing.**

Index Terms— **Self-diagnosis, Wireless Sensor Network (WSN), SDR, Self-healing**

I. INTRODUCTION

Wireless sensor networks have been widely employed for enabling sundry applications such as environment surveillance, scientific observation, Traffic monitoring, indoor climate control, surveillance, precision agriculture etc. A sensor network typically consists of an immensely colossal number of resource constrained sensor nodes working in a self organized and distributed manner. Commonly considered sensor network is composed of an immensely colossal number of sensor nodes, which are densely deployed either inside the phenomenon or very proximate to it. The position of sensor nodes need not be engineered or pre-determined. This sanctions desultory deployment in inaccessible terrains or disaster assuagement operations. On the other hand, this additionally betokens that sensor network protocols and algorithms must possess
Self-organizing capabilities because deploying and maintaining the nodes must remain inexpensive – manually configuring astronomically immense networks of minuscule contrivances is impractical. The nodes are able to accumulate process, disseminate and store data. They perceive the environment, monitor different parameters and amass data according to the application purport. Another unique feature of sensor networks is the cooperative effort of sensor nodes. Sensor nodes are fitted with an on-board processor. In lieu of sending the raw data to the nodes responsible for the fusion, sensor nodes utilize their processing abilities to locally carry out simple computations and transmit only the required and partially processed data. The reason for this is that computation is much more frugal than communication in regard to the most critical resource, the energy. When remotely-deployed nodes become unresponsive, it is usually difficult to determine what caused some node to become silent, without sending a person to the field. If the cost of such field trips is large, remote damage assessment becomes highly desirable to assess the desideratum for intervention. For example, if the cause of the quandary is energy depletion, there may not be much that can be done about it until the energy source is renovated. On the other hand, if the cause is attributed to a transient error, power-cycling the system remotely may fine-tune the quandary. If the cause is attributed to a hardware malfunction, the exigency of rehabilitation may depend on whether or not the failure has affected the ability of the application to sample and store data.

The advantages of self-diagnosis are threefold. First, self-diagnosis can preserve a large amount of transmissions by applying local decision. Second, self-diagnosis evades information loss on the way to sink and thus amends precision of the system. Finally, it provides real time diagnosis results. Healing of such faults is equally important as its diagnosis [3]. The concept added into self-diagnosis is self-healing which is a method to discover, diagnose, and react to network disruptions. Through self-healing it is possible to detect system malfunctions or failures and commence corrective actions predicated on defined policies to recuperate the network or a node. The automatic recuperating from damages improves the accommodation availability.

In spite of all these benefits, employing a self-diagnosis strategy in an astronomically immense large scale WSN is arduous. First, it is well kenned that the computation and storage resources at each sensor are constrained, so the components injected to nodes have to be light-weight. Second, a sensor only has very narrow scope on the system state and in many cases it can remotely determine the root causes simply predicated on local evidences. The fault detector based on Finite State Model (FSM) satisfies the above conditions for the detector.

The rest of this paper is organized as follows. Section II presents the clustering architecture of the system. Section III explores the diagnosis technique through which the most commonly occurring faults are diagnosed. Section IV introduces the self healing mechanism. Section V compares the results obtains through the existing and proposed systems. The paper concludes with section VI.

Adithya Raj, Communication Engineering, Sree Buddha College of Engineering for Women, Pathanamthitta, India, 9633858009

Sherin Koshy, Communication Engineering, Sree Buddha College of Engineering for Women, Pathanamthitta, India

II. CLUSTERING ARCHITECTURE

In the architecture sensor nodes are grouped into clusters controlled by a single node. Every cluster has a gateway node which manages the working of nodes in a cluster. Clusters can be composed predicated on many criteria such as communication, number of nodes and its types. In this mode, the gateways collaboratively locate the deployed sensors and group them into clusters so that sensors transmission energy is minimized while balancing the load among the gateway as shown in Figure.1. In this paper, we postulate that each sensor node will make connections with its random neighbors and stop making connections after each node get connected with a particular number of nodes including itself. As the next step, cluster heads will be determined predicated on which nodes have higher traffic rate. After which these nodes get assigned under nearby cluster heads. Since total nodes are divided in to clusters each having a cluster head, the sink node can locate the faulty sensor node in a faster way. i.e. based on the head node from which data is received sink node can identify the cluster in which the faulty node is placed. This will improve the efficiency of diagnosis process.

A sensor network with different number of cluster heads are preferred to make the system more efficient otherwise cluster head may get overloaded with the incrimination in sensor density, system missions and detected targets/events. Such overload can cause delay in communication and inadequate tracking of targets or a sequence of events. To sanction the system to cope with supplemental load and to be able to cover an immensely colossal area of interest without degrading the accommodation, network clustering is customarily utilized by involving multiple gateways [10], [9].

A. Protocol

The protocol used for the cluster head selection is a clustering based protocol, LEACH. LEACH is a hierarchical protocol in which most of the nodes transmit to cluster heads, and the cluster heads aggregate and coordinate the data and forward it to the sink. It utilizes randomized rotation of local cluster-heads to evenly distribute the energy load among all the sensors in the network. Data aggregation reduces amount of information to be sent to sink. Each of the nodes uses a stochastic algorithm at each round to check whether it will become a cluster head in this round. LEACH postulates that each node has a radio powerful enough to directly reach the sink or the nearest cluster head. Nodes that have been cluster heads previously cannot become cluster heads again for *P* number of rounds, where *P* is the desired percentage of cluster heads. Thereafter, each node has a probability of $1/P$

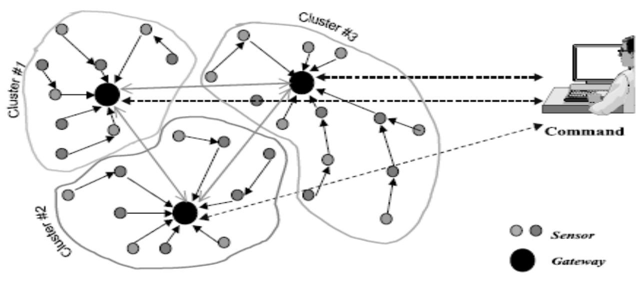

Fig.1. Clustering

becoming a cluster head in each of the round. At the end of each round, a node that is not a cluster head selects the closest cluster head and joins under that cluster. The cluster head then assigns a schedule for each node in its cluster to transmit its data. LEACH protocol is dynamic since the job of cluster head rotates among all the nodes.

$$T(n) = \frac{P}{1 - P \times \left(r \bmod \frac{1}{P} \right)} \qquad \forall n \in G$$

$$T(n) = 0 \qquad \forall n \notin G \qquad (1)$$

For making cluster head decision any node n will choose a random value between 0 and 1. If the value chosen by n is less than T(n), that node becomes the cluster head. Here T(n) is the threshold, P is the desired percentage of cluster head, r is the current round and G is the set of nodes that have never been cluster heads in the last 1/p rounds [11].

III. DIAGNOSIS TECHNIQUE

A. Fault Detector Design

The detectors used are based on the Finite State Machines (FSMs) model. A FSM model consists of a certain number of states and transitions between these states. A state change be enabled when specified condition is fulfilled and it is considered to be the transition in FSM. Current state of a node is determined by the historical states of the system, so it indicates the series of inputs to the system from the very beginning to present moment. The fault detectors generalize the FSM model and use the local evidences on each sensor node as inputs. Each state can be seen as an intermediate diagnosis decision and if the local evidences support certain conditions on current state, the detector state will be transited to the corresponding new state [5].

Since there can be various kinds of failure cases, single fault detector cannot cover all of them. So to control this issue, these faults into different categories based on its symptoms. We consider three classes of symptoms. The first category of symptoms is caused by local errors such as the low battery power or system reboot, which betokens we can pinpoint the root causes from the local evidences only. The second category relates to failures on other nodes, for example if current node detects that a neighbor has just been abstracted from its neighbor table, it will issue a fault detector to neighborhood to ascertain whether this neighbor is still alive. The third category of symptoms can be caused by local or external quandaries while multiple nodes interact with each other. For example, when two nodes are communicating with each other and the sender experiences a high retransmission ratio on its current link. The node, however, is unable to ken whether it is because of the poor link quality or the congestion at the receiver. To deal with unknown type of failures, our solution provides an open framework that can scale to incipient fault types by developing and disseminating incipient fault detectors to sensor nodes [1].

B. Message and Report Concepts

The message exchanged during the diagnosis process includes four major components, the source node ID that engenders the fault detector, the detector type, current state of the detector and other additional information. Upon receiving the state of an incipient fault detector, the sensor node will check whether it can contribute to this fault diagnosis task. If it has some cognizance, the sensor will transit the state of the corresponding fault detector and propagate the incipient state, otherwise it simply drops or broadcasts the state to other nodes according to the lifetime of this detector. Note that each fault detector has a circumscription homogeneous to TTL on the number of hops it is delivered. When the final diagnosis decision is made at some node, it will endeavor to report the decision to sink. If further information is required by the sink, the corresponding sensor nodes will commence the active information accumulation components [1].

C. Change Point Detection and Analysis

Change-points are the abrupt variations in the generative parameters of a time series and by apperceiving these variations we can know whether there are ostensible changes in the parameter values. There are many subsisting solutions for change-point detection and analysis. Considering the circumscribed computation and storage resources in sensor nodes, in this work we apply a light-weight approach which cumulates the cumulative sum charts (CUSUM) and bootstrapping to detect changes [1].

a) Cumulative Sum Chart

Some traffic data are taken to explain the diagnosis triggering mechanism from Figure.2. Postulate that the window size is 12 and thus a sensor node keeps 12 latest data points of its ingress traffic. Initially the cumulative sum charts of this data sequence needs to be calculated.

$$C_i = C_{j-1} + (X_{i\,-}\, X) \qquad (2)$$

Where $\{X_i\}$ i = 1, 2, . . . , 12 indicates the data points in the stream and X be the mean of all values. The cumulative sums are represented as $\{C_i\}$ i = 0, 1, . . . , 12. Here we define C_0 = 0 and then the other cumulative sums are calculated by adding the difference between current value and the mean value. The CUSUM reaches zero at the cessation. An incrimination of the CUSUM value betokens that the values in this period are above the overall average value and a descending curve denotes that values in the corresponding period are below the overall average. A straight line in the cumulative sum charts betokens that the pristine values are relatively stable. In contrast, bowed curves are because of the variations in the initial values. The CUSUM curve in Figure. 2 turns in direction around *C*6 and we can infer that there is a significant change. Besides making decision directly according to the CUSUM charts, we additionally suggest a confidence level to our tenaciousness by a bootstrap analysis [1].

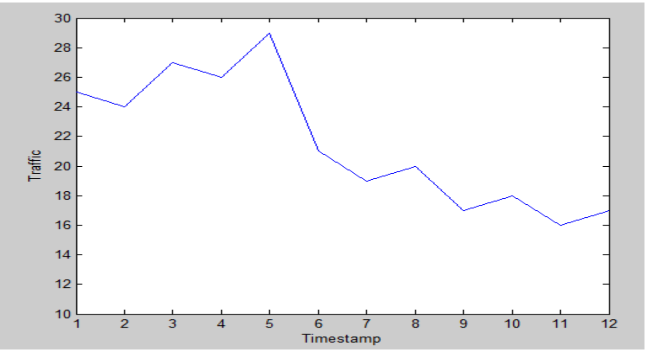

Fig.2. CUSUM

b) Bootstrap Strategy

Estimator (D_c) of the change can be calculated as follows.

$$D_c = \max(C_i) - \min(C_i) \qquad (3)$$

For calculating bootstrap each time the original data sequence needs to be reordered. The conception behind bootstrap is that arbitrarily reordered data sequences simulate the comportment of CUSUM if no vicissitude has occurred. With multiple bootstrap samples it is possible to estimate the distribution of *Dc* without value changes. We then derive the confidence level by comparing the *Dc* calculated from values in pristine order with that from the bootstrap samples. Where D_c is calculated from the pristine data sequence and D_c^j is derived from a bootstrap, *m* is the total number of bootstraps performed. If the confidence is above a pre-designated threshold for example 90%, we decide that there is an ostensible vicissitude in the parameter values [1]. Bootstraps performed are shown in Figure.

$$\text{Confidence} = \text{Number of } (Dc > Dc^j)/m \qquad (4)$$

IV. Healing Technique

Wireless sensor networks are now being considered for many critical applications, which are often largely unattended and need to operate reliably for years. However due to the authentic world communication, sensing and failure realities, node faults and system performance may degrade gradually with time. It is highly desirable that these natural deteriorations can be monitored perpetually and can be

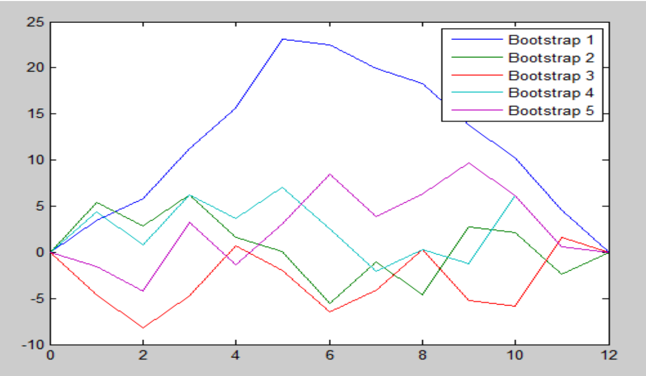

Fig.3. Bootstraps

redressed with self healing when obligatory. In this paper, we introduce a self healing scheme for wireless sensor networks.

Sensor nodes usually transmits the sensed data by IEEE 802.15.4 standard (zigbee) in which the frequency band is divided in to different slots i.e. zigbee device operating at 2.4GHz can operate at 16 different channels. So whenever the link quality is found weak if the sensor node can shift the operating channel to another one it will improve the performance of wireless link. For this, the sensor node should be able to shift to different predefined channels depending upon the link quality (This technique is called software defined radio technique). So by implementing this technique to this self diagnosing method the performance of WSN can be improved. The sensor nodes will act as self diagnosing as well as self healing. This method can also be used for diagnosing and healing several other faults like routing loops, congestion etc. Problems occurring in a wireless sensor network due to congestion is detected and controlled by assigning some threshold value to the nodes. This helps in avoiding congestion in upstream direction which is the direction from sink to sensor nodes. It considers an amalgamation of both present & past loading conditions of the current buffer occupancy in the receiving node. If the occupancy of a node exceeds the threshold value, then congestion scenario is inferred. The node which has detected the congestion will notify its upstream neighbors to decrease the flow by backpressure mechanism.

TABLE I
Comparison Results

Methods	Packet Loss	Throughput
Self-Diagnosis	High	Low
SDSH	Low	High

V. Result Analysis

The table I indicates the result comparison between existing and the proposed systems. Existing system deals with the diagnosis of faults only. So the packet loss is high and throughput is low when compared to the proposed system since it doesn't take steps to solve the problems. In the proposed system it heals the faults along with its diagnosis. It finds the alternate path using distance vector algorithm when a faulty node is diagnosed in between the source node and the destination node. Using the RTS/CTS mechanism it is able to reduce packet loss. And thus it increases the throughput.

VI. Conclusion

Sensor Networks (WSN) promise researchers a powerful mechanism for observing sizable phenomena with fine granularity over long periods. Since the precision of data is important to the whole system's performance, detecting nodes with faulty readings is an essential issue in network management. The goal of fault detection is to verify that the services being provided are functioning properly, and in some cases to predict if they will continue to function properly in the near future. And for recovering these problems human intervention is required. It can lead to errors, it has a high cost and it is not efficient.

In this paper, we introduced self healing along with self diagnosis. So by implementing this technique to self diagnosing method the performance of WSN can be improved. The sensor nodes will act as self diagnosing as well as self healing. This enable systems to continue operating according to their specifications even if faults of a certain type are present.

References

[1] Kebin Liu, Qiang Ma, Wei Gong, Xin Miao, and Yunhao Liu "Self-Dia gnosis for Detecting System Failures in Large-Scale Wireless Sensor Ne- tworks," IEEE Transactions On Wireless Communications,Vol. 13, No 10, October 2014

[2] P. Li and J. Regehr, "T-Check: Bug finding for sensor networks," in Pro- c.*IEEE/ACM IPSN*, 2010, pp. 174–185

[3] K. Liu *et al.*, "Passive diagnosis for wireless sensor networks," in Proc. ACM *SenSys*, 2008, pp. 113–126.

[4] Alec Woo, Terence Tong, David Culler"Taming the Underlying Challen ges of Reliable Multihop Routing in Sensor Networks", *SenSys'03,* November 5–7, 2003, Los Angeles, California, USA.

[5] Ram Rajagopal, XuanLong Nguyen, "Distributed Online Simultaneous Fault Detection for Multiple Sensors", Research supported by California Department of Transportation and AROMURI UCSC-W911NF-05-1-0246-VA-09/05

[6] Anas Abu Taleb, J. Mathew and D.K. Pradhan, "A Novel Fault Diagnosis Technique inWireless Sensor Networks", InternationalJournal on Advanc in Networks and Services, vol 2 no 4, year 2009

[7] Kebin Liu1, Qiang Ma, Xibin Zhao and Yunhao Liu, " Self-Diagnosis for Large Scale Wireless Sensor Networks", IEEE INFOCOM 2011

[8] Sushruta Mishra, Lambodar Jena, Aarti Pradhan , "Fault Tolerance in Wireless Sensor Networks", International Journal of Advanced Research in Computer Science and Software Engineering

[9] Yunhao Liu Kebin Liu, " Passive Diagnosis for Wireless Sensor Net- works", IEEE/ACM Transactions On Networking, Vol. 18, No. 4, August 2010

[10] http://www.wikipedia .com

[11] http://www.google.com

Adithya Raj is currently a PG student at the Department of Communication Engineering at the Mahatma Gandhi University, Kerala. She received bachelor's degree in Electronics and Communication Engineering (2013) from Anna University, Chennai. Her main research interests are in wireless sensor networks.

Sherin Koshy is currently an assistant professor at the Electronics and Communication Engineering Department in Sree Buddha College of Engineering for Women under Mahatma Gandhi University.