A Survey Paper on Fault Analysis and Management in Wireless Sensor Networks

Sangam Sahai, *Department of Computer Science, Texas Tech University.*

***Abstract*—A 'Wireless Sensor Network' (WSN) is a network of autonomous sensors spread out in any environment, which is used to monitor environment's physical condition like pressure, temperature etc. WSN's are evolving as key growth areas in the field of ubiquitous computing. Usually, WSN's are deployed in hostile and remote areas like forests, deserts, war fields etc.The bad effects of such harsh environmental conditions can take a toll on WSN, which includes the sensor nodes and the network. Also, environmental hazards cause the geographical topology of the area to change. This also may lead to network partitioning and communication outage. All this makes the WSN's so vulnerable and leads to the dissemination of imprecise or faulty data. Hence, given these harsh environmental conditions, it is challenge to maintain the quality of WSN and sustain the correctness of data it produces. Therefore, robust Fault Detection and Recovery mechanisms are the need of the hour. With the growing use of WSN's, we need smarter detection and recovery techniques, which minimize the vulnerability of WSN's and make them produce as correct and accurate data as possible. In this paper, firstly we will classify fault management approaches. Then this paper presents a survey of the existing techniques. We will discuss those techniques and finally discuss the scope of future work on this topic.**

I. INTRODUCTION

Internet of Things (IOT) is all about smart objects communicating with each other. IOT is a scenario which has the ability to exchange data over a network without human intervention. Needless to say, Wireless Sensor Networks form an integral part of IOT. IOT and WSN go in tandem with each other and cover a variety of topics like protocols, applications, domains etc.

As per the IOT point of view, in the future, all the physical objects will be impregnated with smart objects having computational and communication capabilities. And the capabilities will be utilized in reporting monitoring their actions .

A very simple example of this could be like, a new age library could have books which are equipped wita RFID tags. And hence , each book could be located through a WSN system which has been set in the library. And later , this system can also be integrated into the library's book search engine , which would provide the users , the most recent and most accurate data.

Inherently, the primitive fields of WSN , Embedded , Control Systems, and Automation , all contribute to the existence of IOT. Hence we see that IOT and WSN are very closely coupled with each other , and in this paper , we are going to cover a very major aspect of WSN's i.e. Fault Management.

To get a good grasp on fault management in WNS's, we first need to understand its importance. Sensors are small electronic devices with extremely high energy constraints. With limited battery power, they are expected to operate independently for years. Since any number of nodes can die out at any time, faults are very likely to occur. Also, since energy conservation is of prime importance, the fault management techniques should be potent yet energy efficient. Hence we have to strike a careful balance between the potency of the fault management technique and the energy which it consumes. Fault management approach for a WSN can be classified into three phases - Fault Detection, Fault Diagnosis and Fault recovery.

II. RESEARCH MOTIVATION

Usually, the efficiency of a good fault management approach is gauged in terms of accuracy, the number of faulty nodes tolerated, energy efficiency and the messaging overhead caused. Hence, the process of designing a good approach is very intricate and entails a lot of research work. A lot of intelligent tradeoffs have to be made. Also, since human intervention is very difficult in case of remote areas, we need to build smart sensor nodes, which take fault management more responsibly and are equipped with decision making, in order to make a self-managed network. Another major roadblock which Wireless Sensor Networks face is the vulnerability to message loss. We all know that messages passing in WSN's is not reliable. Hence, guarantee of a Hundred Percent accurate fault detection system is very difficult. Also, a lot of existing approaches are very efficient but they focus only on small network areas. Their efficiency degrades as network grows. So currently, we need solutions, which are far more scalable than the existing ones.

III. RELATED WORK AND ITS COMPARISON AND ANALYSIS

We classify the already existing fault detection approaches which are being used currently into two categories - 'Centralized and Distributed'[3].In the centralized approach, one particular central node (central controller) takes the responsibility of fault detection in the network. This central controller may be the geographical or logical center of the network.

This central node works on 'Active Detection Model' periodically disseminating 'keep-alive' messages or queries, into the network to retrieve the state of all nodes. However, in 'Distributed Approach' the sensor nodes have the power to make a few decisions. This involves failure detection via neighborhood coordination.

In the Distributed Approach, the nodes themselves coordinate with the neighboring nodes to sniff suspicious nodes and do not burden the central controller repeatedly. Suspicious nodes can be identified by comparing their sensor readings with the median of the reading of its neighbors. This method is very energy efficient as the central controller is not bothered every time and a lot of messaging overhead is reduced .This method can also be used to fortify the accuracy of the failure detection. Instead of disseminating the alarm directly after a failure detection, the node consults with its neighbors to reach a more accurate decision.

The two phase neighboring scheme is one such approach as in Hsin et al [7].However, the downside of the algorithm is that if a lot of nodes in close vicinity are faulty, then this algorithm will not detect failed nodes properly and will produce faulty results.

Another novel fault diagnosis mechanism was proposed by Kuo-Feng [2]. Source nodes send copies of same data to the sink using 2 different paths. If the data through both nodes is not same, that means there is a fault. Then, to find out the faulty node, the sink sends data packet back to the source node through both the paths and monitors the status after every hop and hence easily detects the faulty node.

Another fault detection mechanism which exploits the power of parallel computation is '*Cluster Based Approach'*[5]. In this approach, clusters (group of nodes) execute in parallel to detect failures.By definition, a cluster is a area in the network which is a disk with radius equal to the transmission range of the cluster head.Within any cluster , any node can talk to any other node in the cluster either directly or through the cluster head.

There can be two types of communication in this model.

Intra-Cluster – Where the nodes in the same cluster communicate with each other or the cluster head.

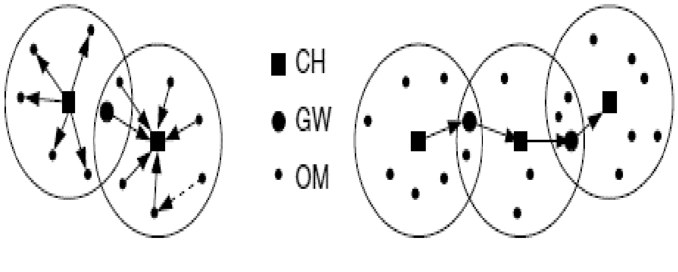
Inter-Cluster – Where the cluster heads of different cluster communicate with each other.

Following is a figure to explain this[5].

In this figure, CH stands for Cluster Head.

GW stands for 'gateway node'. Gateway node is any node which is a one hop neighbor of more than one cluster heads.

OM stands for 'ordinary members', which are neither cluster heads nor the gateway members.



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

The central node in the cluster is called 'Cluster Head' (CH).The clustering algorithm begins by first choosing the cluster head. There could be many ways of doing this. Through a special message that serves as a one hop message probing, a node identifies itself as a cluster head if it realizes that it has the minimum node ID within all its one hop neighbors.

In this model, the 'failure detection service' is executed locally in the clusters and if any failure is detected , then the cluster head spreads this information throughout the network by communicating to other cluster heads. The CH and the nodes exchange 'heart-beat' messages. Based on these messages, the CH detects failed nodes based on a failure detection algorithm.

Now we will delve into the detailed mechanism of this algorithm.

Step one - the nodes (OM) send heart beat diffusion messages to the cluster head.

Step two - In response to this, the cluster head also broadcasts back the heart beat diffusion message.

Step three- Now , in step one , when a OM sends out the diffusion message to the cluster head, other nodes also overhear this message. Based on this , every node prepare a list of all the nodes , who sent the diffusion message , based on overhearing the messages from other nodes. Then each OM sends this digest message to the cluster head. Basically this the OM's view of all the alive OM's present in the cluster head.

Failure Detection – After the above steps are performed, based on the following conditions , the cluster head computed if any not id dead or not. A node is dead if

First Condition – If the cluster head does not get any heart beat message from that node

Second condition – No digest message tells about the awareness or existence of that node.

If the above two conditions are satisfied , then the node is deemed dead by the cluster head.

This method is quite energy efficient, as it uses just one acknowledgement in place of two. It uses 'implicit acknowledgement' which is, sender overhearing the same message back from receiver, when receiver broadcasts it.

In case of densely populated WSN with large number of faulty nodes, a lot of false detections are produced. Hence we need an approach to address this. One of such approaches is *'Bayesian Algorithm for Data Fault Detection'*[4].Bayesian network is introduced to calculate fault probability of all sensor nodes and later the fault probability will be rectified by analyzing the neighboring nodes to enhance the accuracy of fault probability. The sensor nodes mutually exchange readings with their neighboring nodes to calculate the probability of fault. And finally, if sensor node's probability of fault is greater than a threshold value, it is considered as a faulty node.

The detailed algorithms is as follows –

Step one – In the beginning , the fault probability of all the nodes is set to 0.1 (but actually , it has been experimentally verified that the results of this algorithm do not depend on this value)

Step two – Nodes exchange their reading and fault probability with their neighboring nodes.

Step three – Assuming S(i) and S(j) are neighboring sensor nodes and S(j) is receiving the readings from S(i) , then S(j) will calculate V(ij) which is the *Variation* in the reading.

Step four – There is a pre set *Reading Threshold* value R.

If V(ij) is less than R, then f(ij) will be set to one.

F(ij) is the node's status flag. If F(ij) is one that means both the nodes – S(i) and S(j) share the same status. They both are either good or , they both are faulty. And if the F(ij) is zero , then their statuses are opposite. That means if one is good , then the other will be faulty or vice versa.

Step five – Each node has the readings and the fault probability (p) of it’s neighbors. Based on this, each node will calculate its ‘Posterior Fault Probability ’.

Given a sensor node S and the set of its neighbors(S), the fault posterior fault probability of S is –

$$\boldsymbol{P(S/N(S)) = P(S,N(S))/P(N(S))}$$

Step six – Based on the status flag (f) , fault probability (p) and the posterior fault probability, ‘Confidence’ of each node is calculated. If the confidence is greater than zero , then node is deemed faulty .

This method has been proven to be very accurate if average number of neighboring nodes is large and fault rate is high.

Fault Diagnosis is also a crucial aspect of Fault Management. Many tools facilitate fault diagnosis in WSN’s. In most of the tools, there are 2 major components - Information Collection and Root Cause Deduction [6].But in most of the tools, the information collection process is not coupled with root cause deduction. This results in overburdening the network and increased amount of redundant information.

A new approach called ‘*Directional Diagnosis’*(DID) addresses this issue in which the information gathering for diagnosis is driven by the fault inference process. This DID is mainly composed of four components – a node tracing module , a trace collecting module , a probabilistic inference model and an incremental probing module[6].

The basic algorithm of this model is as follows -

Step one - In this approach, each node caches its communication pattern. A small trace sniffer is concealed in all the nodes for this, and the communication pattern is periodically collected by the sink for diagnosis purposes.

Step two - The network exceptions are pre loaded in the sink.

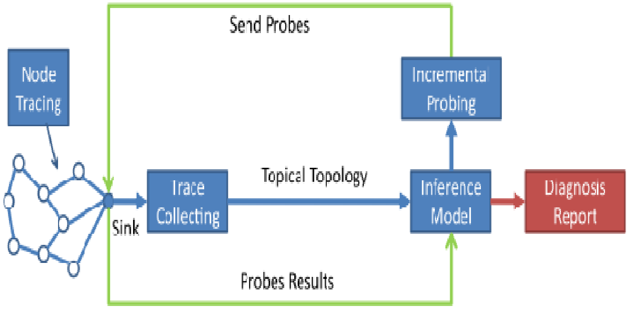
Step three -At the sink , a trace collecting process is fired in case some network exception is detected (by matching the communication pattern and the pre loaded exception pattern).In this case , the sink analyses the rows of the suspicious nodes as per the data it has , and then bundles this information as a ‘*collecting probe*’ and broadcasts this again in the network.

Step four- Based on this collecting probe, it again gets back some information.

Step five - After gathering the demanded information, the sink re analyses the data.

Step six - This probing stops when the sink finds the most critical symptoms that cause the problem and then diagnosis report is generated.

Following figure shows the overall Directional Diagnosis system architecture [6].



IV. Discussion

Though we have discussed so many approaches for fault detection and management, but no approach is perfect and every approach has a cost attached to it. The centralized detection approach is very accurate and energy efficient but is not fit for large-scale networks. The central controller gets overloaded. There is high volume is traffic at and around the central node which results in instant drainage of energy.

On the other side, distributed approach is very energy efficient and increases network lifetime. Since every network is prone to message loss, the message of node failure detection can also be lost. Hence, it is a big challenge to come up with any Fault Detection Service(FDS) which is complete and accurate.

Cluster based approach is very energy efficient and robust. But it also has certain problems. In the case, when any node is sleeping during the time when heartbeat message is exchanged, a false ‘failed node detection’ will be perceived.

V. Conclusion

Many ongoing research projects around the world have started focusing on ‘self-healing’ networks which automatically detects faults in the network and rectifies those faults automatically with minimum human intervention.

Future work of self-managed WSN(s) is proposed in MANNA[1] architecture. This concept proposes that WSN should reconfigure them in case of any event which demands intervention (like failure of network or faulty node).

Also, when a node fails, the network loses the data which was sensed in its last active cycle. And unfortunately, there are not many techniques which focus on recovery of that data. A few techniques have suggested frequent backup of data by sending update messages to either sink or some other central node.

But this needs more memory and entails extra energy. So, we need better mechanisms for this.

A lot of work is being done to make the sensor nodes perpetual energy machines. Experiments are being conducted where solar cells have been mounted on nodes for energy harvesting. But this method has still not guaranteed that the sensor nodes will not die out forever. And hence scientists are trying to make the nodes ‘eternal’, energy wise.

In any network , security is something which is of prime importance in today's world. This also applies to wireless sensor networks. There can be places where security breach can cost thousands of lives. For example, in case of a war front , many ad-hoc wireless networks are installed to alert the soldiers against enemies. In imagine , in this case , if some malicious agent intercepts the messages and meddles the messages , hundreds of soldiers can get wrong information and loose their lives. Sadly, not much work has been done in the field of security for wireless networks. This research area needs a lot of attention and all around the world , people have started working on this area now.

Another roadblock for crafting a fine failure detection scheme is mobility. These days , mobility is the hot topic. Networks are going mobile and in this case , the nodes are not static. They keep changing their location and some times , this change of location is at a fast rate. This throws in a lot of computational complexity in the fault detection systems. The normal fault detection cannot work equally well in the case when the nodes become mobile. In this case , a lot of extra measures have to be taken and in a few cases , the complete Algorithms have to be re designed in order to make them fit for mobile networks. A lot of work is being done these days to make the failure detection techniques fit to work in mobile wireless sensor networks too.

So as we saw, combining all the features and bundling them into a nice failure detection is not an easy job. Factors like design of energy efficiency , scalable and fully autonomous failure detection solution , fault recovery systems have to included for a fully functional and robust fault detection scheme.

The final thought which needs special attention is that Wireless sensor networks is a new emerging scientific field and hence it is still not very crystal clear as to , what could be the best way to address a any problem in the network. Hence , it is very difficult to predict the best strategies to strike out faults in the wireless sensor network system.

Also , the technology and applications are also changing really rapidly these days. For example , the latest approaches are focusing more and more on energy conservation. But the catch is the techniques remove the messaging overhead to reduce the energy conservation , but in this case , if the computation required by the system is more , the it is of no use. As the energy which is being saved in the communication will be depleted in the computation And this case , the whole algorithm will have to be structured again. Hence during the designing of fault management solutions both communication overhead and computational overhead should be evaluated and a careful trade off should be made if required.

In wireless sensor networks , each node is a small computational unit , with a communication and storage sub unit and battery system. So besides managing fault management the system through software , a careful thought should be given to make the hardware as robust as possible. The nodes could be make smarter by making the hardware more robust and smart.

Nodes can be made which can automatically fix hardware level problems if detected. If the problem cannot be self fixed , then the nodes could send out automated report of the detailed description of the problem , so that minimum manual intervention is required to fix the problem. For example , image a scenario where the sink or the central controller would get the detailed explanation of the issue a node is facing and also the hardware problem which needs to be fixed or repaired. In this case , humans would not have to go and manually debug the problems. They would exactly know what is required and hence the will be able to fix the problem very quickly. This would end up in saving a lot of time. And imagine a wireless sensor network where there are thousands of nodes . In that case , this feature would prove to be a blessing in disguise for the network engineers.

In this paper, we discussed and analyzed a few fault detection and management approaches. However, still there are so many research challenges and roadblocks which need to be addressed.

VI. References

[1] L.B. Ruiz at al. “Fault management in Event-Driven Wireless Sensor Networks”,ACM Press, Italy , 2004.

[2] K. Ssu et al. “Detection and Diagnosis of Data Inconsistency Failures in Wireless Sensor Networks, IEEE.

[3] Mengjie Yu; Mokhtar, H.; Merabti, M., "Fault management in wireless sensor networks," Wireless Communications, IEEE , vol.14, no.6, pp.13,19, December 2007

[4] Hao Yuan; Xiaoxia Zhao; Liyang Yu, "A Distributed Bayesian Algorithm for data fault detection in wireless sensor networks," Information Networking (ICOIN), 2015 International Conference on , vol., no., pp.63,68, 12-14 Jan. 2015

[5] Tai, A.T.; Tso, K.S.; Sanders, W.H., "Cluster-based failure detection service for large-scale ad hoc wireless network applications," Dependable Systems and Networks, 2004 International Conference on , vol., no., pp.805,814, 28 June-1 July 2004

[6] Wei Gong; Kebin Liu; Yunhao Liu, "Directional Diagnosis for Wireless Sensor Networks," Parallel and Distributed Systems, IEEE Transactions on , vol.26, no.5, pp.1290,1300, May 2015

[7] C. Hsin and M.Liu, “Self monitoring of Wireless sensor networks”, Comp. Commun, Vol 29,2005 , pp 462-78