Comparative Study of Fault Management Algorithms in Wireless Sensor Networks

Author: **Charu Virmani**

Assistant Professor,

Manav Rachna International University, Faridabad

Khushboo Garg

Student, Manav Rachna International University, Faridabad

Haryana (India)

Abstract

Wireless sensor network is an uprising low cost sensor network for sensing, integrating and processing data collected by sensors. It provides a wide area of research as they can be used in various applications (e.g. home, military, security etc.). Wireless sensor networks are the networks which are inherently prone to errors or faults due to sensitiveness of the sensor nodes. These sensor networks are usually deployed in remote and unattended areas hence the effect of harsh environment, limited resource or some other hardware configuration can make them faulty. In order to mitigate the effect of these failures, fault management of these WSNs becomes imperative. Various effective algorithms or adaptive fault tolerant mechanisms are designed till now to achieve a good fault management. In this paper we will discuss already implemented algorithms and existing approaches of network fault management and compare there features for an effective one.

Keywords: Wireless sensor network, fault tolerance, fault management.

1. Introduction

With rapid advancement in wireless communications and networking made possible the deployment of sensitive wireless sensor networks. These WSN consists of spatially distributed sensor nodes which have limited processing power, storage and communication abilities and finite energy supply. The WSN is built from nodes which amount from few to several hundreds or even thousands, and each node is connected to sensor. These sensor devices are armed with a radio transceiver, an antenna, a microcontroller and battery with finite energy supply. This sensor node senses the environmental conditions and reacts on the physical phenomenon. In *computer science* and *telecommunications*, wireless sensor network is an active research area with numerous workshops and conferences arranged each year. Wireless sensor networks must be deployed in the remote and unattended areas, like faraway forests, military area for enemy intrusion, environmental sensing which includes volcanoes, oceans, glaciers etc. where the access or operations to these sensor nodes are rarely possible. Different types of faults like node failure, link failure, network congestion, energy loss etc. may likely to occur. Hence, to ensure the correctness and

scalability of the system *fault management* architecture is very much essential. In this paper we study all these type of faults, reason behind these faults and various algorithms implemented so far for the process of fault management.

Fault management is a key part of network management required to maintain the proper functionality of the system. The system is known as faulty when any error comes in the system or the system behaves arbitrarily or maliciously which is not expected from the system. In order to design efficient fault management different types of faults need to be considered like node failure due to some hardware or software failure, how we can detect a link failure between two nodes, what happen when network congestion occurs or if some adversaries could try to hack the control over the nodes because these nodes are deployed in remote or unattended areas so the adversaries could not only manipulate the environment but can gain physical access to the node. Moreover, as the sensor nodes are deployed in the open external environments so they are also susceptible to the failure due to environmental conditions like rain, fire and fall of trees etc.

This paper is organized as follows: section 2 comprises of related work. Section 3 explains the fault management Frameworks. Section 4 shows a comparative study of existing algorithms. Section 5 reflects conclusion and future work.

2. Related Work

Many techniques have been proposed till now for fault detection and recovery. Shahram and Ali [1] propose a decentralized cluster based method for fault detection and recovery which is energy efficient namely DFDM in order to avoid the faults occurring due to energy depletion. Nan Li [11] proposes a automatic fault management (AFM) to automate many of the fault management tasks by continuously monitoring, analyzing the fault when it is detected self-diagnosis, and taking adaptations actions for self recovery while there is not any algorithm for fault recovery has been mentioned. WinMS [2] provides an adaptive policy-based sensor network management system that provides self-management for maintaining the performance of the network and achieving effective networked node operations without human intervention. WinMS adapts to changing network conditions by allowing the network to reconfigure itself according to current events as well as predicting future events. Tai [3] proposes cluster-based communication architecture to permit the Fault 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. It extensively exploits 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. Asim proposes a fault management architecture which partitions the network into virtual grid of cells to perform fault detection and recovery locally with minimum energy consumption. A cell manager and gateway nodes are chosen in each cell to perform management tasks. Chen [5] locates the faulty sensors in the wireless sensor networks and evaluates a localized fault detection algorithm to identify the faulty sensors. The implementation complexity of the algorithm is low and the probability of correct diagnosis is very high even in the existence of large fault sets. Antonio [7] presents architecture to support medical sensor networks to change the entire medical

environment in hospital on the wireless sensor networks on the basis of 6LoWPAN technology. This protocol carry out inter WSN mobility inside the architecture. Sympathy [9] is designed for data collection applications, which gather distributed data at a centralized sink location for analysis. (Most of today's deployed sensor networks fit this description, as will many networks deployed in future.) Nodes periodically send metrics back to a sink, which combines this information with passively-gathered metrics to detect failures and determine their causes. Sympathy gathers and analyzes general system metrics such as nodes' next hops and neighbors. Based on these metrics, it detects which nodes or components not delivered sufficient data to the sink and infers the causes of these failures. Ruiz and Loureiro [10] Proposes the MANNA management architecture for WSNs. In particular, it presents the functional, information, and physical management architectures that take into account specific characteristics of this type of network. Some of them are restrict physical resources such as energy and computing power, frequent reconfiguration and adaptation, and faults caused by nodes unavailable. The MANNA architecture considers three management dimensions: functional areas, management levels, and WSN functionalities. These dimensions are specified to the management of a WSN and are the basis for a list of management functions. A number of methods and techniques have been proposed specifically for WSNs ranged from fault prevention, fault detection, fault identification, fault isolation, and fault recovery[11][12][13][14].

3. Fault Management Frameworks.

Fault Management is a concept of network management which is concerned with detection, diagnosis, and recovery of faults. When appropriately implemented, it provides the system capable of fault tolerance and minimize error occurrence. Fault management performs some important functions: Monitor the network status and energy level constantly, Automatic correction of potential problems causing conditions, Tracing the location of errors or failure occurring in the system, Alarms that notify administrators and users about occurred failure.

For solving all these problems whole process of fault management is divided into fault detection, fault diagnosis and fault recovery.

3.1 Fault detection

Fault detection is the first phase of the fault management. In this randomly occurring faults must be properly detected. Numerous types of algorithms and model based approaches are used to detect the faults. All the fault detection schemes are broadly classified into two primary types: Centralized Approach and Distributed Approach.

3.1.1) <u>Centralized Approach:</u> In this approach, usually a base station or central controller is responsible for the whole management process of the network. The central controller normally have unlimited sources (e.g. energy). This central controller normally adopts active detection model by periodically sending requests to individual sensor nodes to send their updates of network performance. A centralized framework called MANNA was presented in [10] for fault management. In MANNA, each sensor node is assigned a role as manager or agent, and the manager can build coverage and energy models based on the information receiving from the agents (sensor nodes) in the installation phase, later in the operational phase, the manager

periodically collects information of the other sensor nodes to perform fault detection. Sympathy [19] uses a message-flooding approach to pool event data and current states (metrics) from nodes. To minimize the number of communication messages, a Sympathy node can selectively transmit important events to the sink node. In addition, the central manager in the Warfighter Information Network Management System (WinMS) [2] compares the recent or previously perceived states of sensor nodes against overall network information models to detect the fault. This Centralized approach provides good fault management while it is not suitable for large scale networks. Another drawback is that the central controller becomes a single point of data traffic concentration and hence consumes large amount of energy of the nodes. Third, this central controller becomes a single point of failure for the entire network. If this central controller fails due to some error whole process disturbs and the process should be reinitialized.

3.1.2) Distributed Approach: In this approach, a big network is divided into sub-networks and within each sub-network a central manager is elected which keeps track of faults occurring in that network. In [17], the authors designed a distributed fault management framework called WSNDiag to identify faulty nodes. It is tried that sensor nodes can detect the faults and take decision on its own level so that minimum no. of messages must be transferred to central controller. This reduces communication messages and reducing the no. of data traffic in the network. Some techniques used in the distributed approach areas follows:

i. Neighbor coordination

Neighbor coordination is an important technique of fault management. Nodes coordinate with their neighbors by sending and receiving current hardware update messages to its entire neighbor and keeping record of all its neighbors. The base station does not get the information about the failed node immediately until all the neighboring nodes are not sure about the failure of node. Nodes check the readings of the node with the perceived median readings. If the reading does not match with the expected readings of the other nodes the neighboring nodes assume that the node is in critical level region and getting faulty. The neighboring nodes then inform the managing node about the criticality of the node.

ii. Clustering

Clustering is an important technique of fault management. In this type of architecture the whole network is divided into no. of small networks which we designate as clusters.

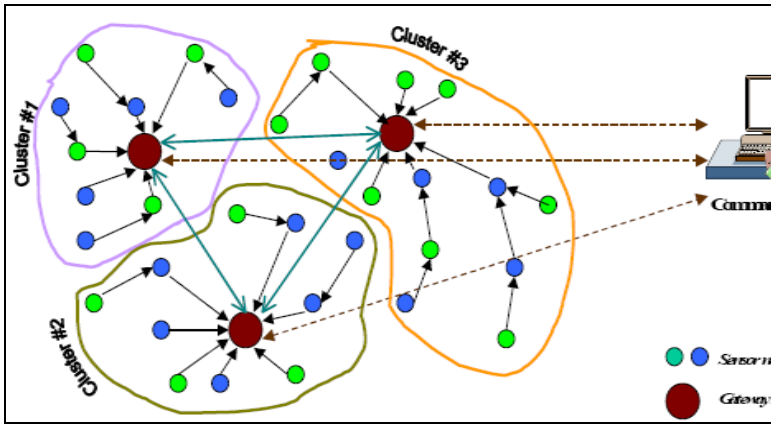

Fig1: Distribution of Clusters in a Network

From each cluster one node is selected as cluster head. This cluster head will take care of that cluster for the faults to occur. Nodes in the cluster can communicate with the sink node via this cluster head only. Even if the nodes want to communicate with the nodes of different cluster, the cluster head exchange messages with the node of different clusters to detect the faulty nodes

in the cluster as well as neighbor cluster head.

iii. Node Level Measurements

A sensor node may monitor itself for resource usage such as CPU utilization, energy and memory usage to estimate its resource utilization. It also monitors other factors such as external noise level, link strength between the nodes and compares all the values with its previously perceived data, to find the faulty nodes. All the information a node collects is location specific and not possible to be measured by other nodes.

3.2 Fault Diagnosis

Fault diagnosis is the second stage of whole process of fault management. It typically covers 3 questions where, what and how? i.e. Where the fault is located (Fault location)? What type of fault it is (fault identification) like node failure, link failure, traffic congestion, energy depletion, etc.? And, how does a fault occurs (fault root cause) i.e. physical environmental changes, hacker attacks, protocol implementation etc.?

The identification of root cause is the main task to repair the fault. There exists some assumptions about the root cause of the faults like while using TCP routing protocol, it is well known that packet loss will occur by traffic congestion hence the speed of the sender will slow down to avoid congestion. While some of the existing solutions try to recover the fault without determining the root cause of the fault, like when a link failure occurs routing protocols fix the fault by selecting alternate link to send the data or by enhancing transmission power of the sender when a link degradation is detected because of obstacles on the link or the channel has strong noise or is too busy.

Another approach to classify the faults is based on machine learning. On the basis of set of symptoms and its potential faults base station can create the cause-effect graph to classify the faults on different issues. For example, Zhang and Lee use a classification algorithm, RIPPER, to identify malicious routing table updates [18]. In [20] Kleer and Williams build a structural and behavioral model. In this model, an acyclic graph represents structure representing influence relationship between the components and behaviors as expected performance output. The combination of structure and expected behaviors can be used to determine the root cause of the fault. These model based diagnosis can be easily extended and flexible to handle new type of faults. Koushanfar *et al.* [21] assume that the system software is already fault tolerant. They focus on sensor node hardware faults, especially sensor and actuator faults, which are most prone to malfunctioning. Koushanfar *et al.* [21] adopt two fault models. The first one is related to sensors that produce binary outputs. The second fault model is related to the sensors with continuous (analog) or multilevel digital outputs. Clouqueur *et al.* [6] only consider faulty nodes due to harsh environmental conditions. In their work, faulty nodes are assumed to send inconsistent and arbitrary values to other nodes during the information sharing phase.

3.3 Fault Recovery

Fault recovery is the third phase of the fault management process. In this phase

network is restructured or reconfigured. When the fault in the network is detected and diagnosed properly then comes the turn to recover from the faults. Sometime the techniques or the protocols used in the network automatically repair or recover from the fault. As we discussed in fault diagnosis recovery from a link failure will be automatically made by routing protocol used by selecting other links in the network for the data transmission to minimize some cost metrics such as (ETX) Expected Transmission Count [19].

When the faults occurring in the network are not repaired autonomously or it is not clear that which adaptation will be most suitable in the current situation then some model based approach can be used like Chen [11] gives a centralized approach to build analytical and simulation model. The central controller feed various configurations into the network models. Each configuration includes gateway selection, channel assignment, transmission power selection, routing table entries and so on. All the configurations are quantitatively compared using desired metrics such as average throughput or power delay. Gaurav *et al.* [15] proposed a run-time recovery mechanism to enable the members of the failed gateway node to reconnect to the network. If a node is in the same communication range of multiple candidate groups, it is recommended to join the group with minimum communication energy lost. Stefano *et al.* [16] considered a solution to recover data loss after a node failure by duplicating and distributing redundant information of sensed data among other nodes in advance. The data storage space of a node is used to store its own sensed data and also the data as redundant copies for another node. Each node periodically updates its data copies stored in other sensors by sending update about the failed sensor.

4. Comparative Study

So far we have studied various special algorithms of fault management approaches each of which explains a different architecture on different parameters. Like DFDM [1] propose a decentralized cluster based method for fault detection and recovery which is energy efficient namely DFDM. In order to avoid degradation of service due to faults, it is necessary for the WSN to be able to detect faults early and initiate recovery actions. WinMS [2] provides self-management for maintaining the performance of the network and achieving effective networked node operations without human intervention. WinMS adapts to changing network conditions by allowing the network to reconfigure itself according to current events as well as predicting future events. WinMS architecture consists of a schedule-driven MAC protocol that collects and disseminates management data, to and from sensor nodes in a data gathering tree. MANNA [10] architecture provides flexibility when defining the three architectures: functional, information, and physical. The coordination among the three planes **is** based solely on policy-based management. The functional architecture allows the establishment of all possible configurations for the management entities (manager, agent, and MIB). Sympathy [9] paper presents the design and evaluation of Sympathy, a tool for detecting and debugging failures in sensor networks. Sympathy has selected metrics that enable efficient failure

detection, and includes an algorithm that root-causes failures and localizes their sources in order to reduce overall failure notifications and point the user to a small number of probable causes. Clustering Based Detection [3] proposes a Fault Detection Service based on the notion of clustering. They used cluster-based communication architecture to permit the Fault 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. SPINs [4] present a suite of security protocols optimized for sensor networks. It has two secure building blocks: SNEP and TESLA. SNEP includes: data confidentiality, two-party data authentication, and evidence of data freshness. TESLA provides authenticated broadcast for severe resource-constrained environments. Mengjie et. al [11] shown the comparison of various fault management approaches. He compares the various algorithm on the basis of various parameters like their configuration they achieve, Data storage etc. It shows that each algorithm possesses some unique characteristics which differentiate these algorithms from one another.

The overall classification and comparison of existing fault management for WSNs on the basis of different parameters are shown in the below given Table.

Approach	Configuration	Perceived Data	Detection	Action Taken
DFDM [1]	Decentralized	N/A	Active	N/A
WinMS [2]	Centralized + Decentralized	Topology & Energy Model	Active	Yes
MANNA [13]	Centralized + Decentralized	Topology & Energy Model	Passive or event driven	Yes
Sympathy [12]	Centralized	Metrics	Active	N/A
Clustering Based Detection [3]	Centralized + Decentralized	Propagating Messages	Active in each cluster	N/A
HWSN6 [9]	Decentralized + Node Coordination	Server from Sensor Readings	Active in each area	Yes
SPINs [4]	Centralized	N/A	Active	N/A
Distributed Failure Detection [6]	Decentralized + Node Coordination	Sensor Readings	Active	N/A

Table 1: Comparison of various fault management approaches

5. Conclusion and Future Scope

As we have seen different algorithms implemented so far, each of this algorithm have unique features and characteristics which make the task of fault management very unusual from the traditional approaches. But still it poses additional technical challenges like avoiding the fault or using some new technology for an efficient fault management. Certain amount of knowledge is needed to identify the different faults. Note that there is a need to address fault models not just at the level of components and individual nodes, but also at the network and system management level. Self-managed WSN requires managing the system by efficient recovery actions to remove the failure impact from the network performance. The centralized approach has been considered as the common solution, which enables the base station (or the sink node) with unlimited resources to execute a wide range of recovery functions in the network. The shortcoming of this is the communication overhead and rapid energy depletion of nodes, especially in large-scale sensor networks. The techniques we are proposing are still under development, so it would be grossly premature to suggest a solution to all the difficulties involved.

6. References

[1] S. Babaie, A.R.Rezaie. "Decentralized Fault Detection Mechanism to improving Fault Management in Wireless sensor Networks" 9th *IEEE international Conference Publications, 2011,* p.no 1026-1029.

[2] W. L. Lee, A. Datta, and R. Cardell-Oliver, "WinMS: Wireless Sensor Network-Management System, An Adaptive Policy-Based Management for Wireless Sensor Networks," School of Computer Science & Software Engineering, Univ. of Western Australia, tech. rep. UWA-CSSE-06-001, 2006.

[3] A. T. Tai *et al.*, "Cluster-Based Failure Detection Service for Large-Scale Ad Hoc Wireless Network Applications in Dependable Systems and Networks," *DSN '04*, Florence, Italy, 2004.

[4] A. Perrig *et al.*, "SPINS: Security Protocols for Sensor Networks," *ACM MobiCom '01*, Rome, Italy, 2001.

[5] J. Chen, S. Kher, and A. Somani, "Distributed Fault Detection of Wireless Sensor Networks," *ACM DIWANS '06*, 2006.

[6] T. Clouqueur, K. Saluja, and P. Ramanathan, "Fault Tolerance in Collaborative Sensor Networks for Target Detection," *IEEE Trans. Comp.*, vol. 53, no. 3, 2004, pp. 320–33

[7] Antonio J. Jara, Miguel A. Zamora, Antonio F. G. Skarmeta. "HWSN6: Hospital Wireless Sensor Networks based on 6 LoWPAN Technology: Mobility and Fault tolerance Management". In *Proceedings of International conference of Computer Science and Engineering, Murcia, Spain, 2009*.

[8] Nan Li, Bo Yan, and Gauling Chen. A measurement study on wireless camera networks. In *Proceedings of the Second International Conference on Distributed Smart Camera (ICDSC),* Stanford, CA, September2008

[9] N. Ramanathan *et al.*, "Sympathy for the Sensor Network Debugger," *ACM SenSys '05*, San Diego, CA, 2005.

[10] Linnyer Beatrys Ruiz and Antonio A. F. Loureiro**, "**MANNA: A Management Architecture for Wireless Sensor Networks," IEEE Communication Magazine'03, Minas Gerais, 2003

[11] Mengjie yu, Hala Mokhtar, and Madjid Merabti, JOHN MOORES UNIVERSITY, "Fault Management in Wireless Sensor Networks", IEEE wireless Communication,2007

[12] R. Mini, A. Loureiro, and B. Nath, The distinctive design characteristic of a wireless sensor network: the energy map. Elsevier Computer Communications, Vol. 27, No. 10, pp: 935–945, Jan. 2004.

[13] N. Ramanathan, E. Kohler, D. Estrin, Towards a debugging system for sensor networks, International Journal of Network Management, Vol. 15, No. 4, pp. 223–234, 2005.

[14] Yunhao Liu, Kebin Liu, Mo Li, Passive Diagnosis for Wireless Sensor Networks, IEEE/ACM TRANSACTIONS ON NETWORKING, Vol. 18, No. 4, Aug. 2010.

[15] G. Gupta, and M. Younis, "Fault-Tolerant Clustering of Wireless Sensor Networks," *IEEE WCNC '03*, New Orleans, LA, 2003.

[16] S. Chessa and P. Maestrini, "Fault Recovery in Single- Hop Sensor Networks," Dept. of Elec. and Comp. Eng., UC San Diego, p. 11.

[17] Chessa S, Paolo S, Crash faults identification in wireless sensor networks, Computer Communications, Vol. 25, No. 14, pp. 1273–1282, Sep. 2002.

[18] Yongguang Zhang andWenke Lee. Intrusion detection in wireless ad-hoc networks. In Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking, pages 275–283, Boston, MA, August 2000.

[19] Donglas S. J. De Conto, Daniel Aguayo, John Bicket and Robert Morris. A high throughput path metric for multihop wireless routing. In *Proceedings of the Ninth Annual International Conference on Mobile Computing and networking*, pages 134-146, San Diego, CA, September 2003.

[20] Johan de Kleer and Brian C. Williams. Diagnosis with behavioral models. In Walter Hamscher, Johan de Kleer, and LUCA Console, editors, Readings in *Model-based diagnosis*, pages 124-130. Morgan Kaufmann publishers, 1992

[21] F. Koushanfar, M. Potkonjak, and A. Sangiovanni-Vincentelli, "Fault Tolerance Techniques for Wireless Ad Hoc Sensor Networks," *IEEE Sensors Conf.*, Orlando, FL, 2002.