ISSN 1512-1232

FAULT MANAGEMENT IN WIRELESS SENSOR NETWORKS

Muhammad Zahid Khan

M.Zahid-Khan@2008.ljmu.ac.uk
School of Computing & Mathematical Sciences
Liverpool John Moores University.
Liverpool, L3 3AF, United Kingdom

Abstract

The design of wireless sensor networks (WSNs) is influenced by many factors, including fault tolerance. Fault tolerance is the ability to ensure the functionality of the network in the events of faults and failures. Sensor nodes in WSNs are expected to operate autonomously for a long period of time and may not be easily approachable for battery replacement and maintenance due to their physical deployment locations. Therefore, in order to guarantee the network quality of service and performance, it is essential for the WSNs to be able to detect faults and failures, and to perform something akin to healing and recovering from events that might cause faults or misbehavior in the network. Therefore, fault tolerance should be seriously considered in many WSNs applications. A set of functions or applications designed specifically for this purpose is called a fault-management platform [1]. Concerning fault management various schemes have been proposed. However, it is evident from the existing literature that very least attention has been paid to evaluate or analyzed these schemes. The contribution of this work is to fill this gap by analyzing some of the dominant schemes in this area. We believe this work will benefits either in development of some new solutions or modifying existing fault management schemes.

Keywords: *Wireless Sensor Networks, Fault Management, Fault Detection, Fault Diagnosis, Fault Recovery.*

1. INTRODUCTION

Recent technological advances in wireless networking and communication, the development of MEMS (Micro-Electro-Mechanical-Systems), and its integration with embedded microprocessors has enabled a new breed of wireless networks [2, 3] known as wireless sensor networks (WSNs). WSNs are composed of a large number of self-organized sensor devices (homogenous and heterogeneous) that work in collaboration to monitor the physical environment and object of interest and relay messages to the Sink or Base Station. Similar to wireless ad-hoc network [4], the design of WSN is influenced by many factors, including fault tolerance [5, 6]. Fault tolerance is the ability to ensure the functionality of the network in the events of faults and failures [7]. Sensor nodes in WSNs are expected to operate autonomously for a long period of time and may not be easily approachable for battery replacement and maintenance due to their physical deployment location. Furthermore, harsh physical environment, e.g. rain, fire and falling of hard objects on senor hardware can also completely damage the device, hence faults and failures are normal facts in WSNs. Thus, in order to guarantee the network quality of service and performance, it is essential for the WSNs to be able to detect faults, and to perform something akin to healing and recovering from events that might cause faults or misbehaviour in the network, hence fault tolerance should be seriously considered in many wireless sensor network applications [8].

A set of functions or applications designed specifically for this purpose is called a fault-management platform, which is an integral part of a network management system. Thereby, a network's management system with an efficient fault management platform makes the network fault tolerant in the events of faults and failures. Fault management is a very important component of network management concerned with detecting, diagnosing, isolating and resolving faults and failures [7]. Proper implementation of fault management can keep the network running at an optimum level and minimize the risk of failure, consequently, make the network more fault tolerant [9]. It is important to mention that in the wireless ad-hoc sensor’s network, that problem has been addressed in most of the dominant schemes. An example would be MAODDP [10, 11] which introduces its own fault discovery and management mechanism. Concerning fault management various schemes have been proposed. However, It is evident from the reported literature that less attention has been paid in analyzing these schemes both on their own and against each other. In addition, considerable effort is needed to implement some of these schemes in a real environment. This work analyses some of the dominant schemes in this area to explore these with respect to their relative benefits and weaknesses. In this context, this work has been organized as follows. In section 2, fault and fault management in WSNs has explained, in section 3 dominant fault management schemes have been analyzed and mention issues in the existing fault management schemes. It followed by discussion in section u and conclusions and future work is given in section 5.

2. Faults and Fault Management in Wireless Sensor Networks

WSNs are uniquely characterized by their limited resources and are often deployed in remote and hostile environments. These highly dynamic networks are very prone to failure and are usually kept unattended. Therefore, an effective and proper fault management strategy is essential to the operation of large-scale WSNs. To comprehend fault management, it is important to point out the difference between faults, error and failures [12].

- A fault is any kind of defect that leads to an error.
- An error corresponds to an incorrect (undefined) system state that may lead to failure.
- A failure is the observable manifestation of an error which occurs when the system deviates from its specification and cannot deliver its intended functionality.

Koushanfar et al. [13] categorized faults into three types: *permanent faults*, *Intermittent faults* and *Transient faults*.

Permanent faults – permanent faults are continuous and stable in nature, e.g. hardware faults within a component. Permanent faults completely disconnect the sensor nodes from other nodes, and bring significant impact on the performance of the network.

Intermittent faults – an intermittent fault has the occasional (such as a regular or irregular interval) manifestation due to unstable characteristics of the hardware, or as a consequence of a program being in a particular subset of space.

Temporary or transient faults – these faults are the result of some temporary environmental impact on otherwise correct hardware, e.g. the impact of cosmic radiation on the sensor node or the obstacle or weather conditions in the harsh environmental might temporarily disrupt the radio communication link between sensor nodes and the network [14].

Potential faults – potential faults are usually occurring due to the depletion of node hardware resources, such as node's battery energy exhaustion. Such faults can cause node's sudden death, and ultimately impair the network performance and lifetime [14].

2.1 Fault Model

A system failure occurs when it deviates from its specification and can't deliver its specified services. For example, transmission link failure, software failure, hardware component failure, and malicious attack, unexpected increase in traffic, interface disconnection and miss-configuration. Software and hardware faults affect the whole system state and its operational behavior, such as memory or registrar contents, program execution and control flow, and communication link, etc. [9, 15]. Data generation and delivery in sensor networks are also inherently faulty and unpredictable.

Failures in WSNs can occur for various reasons. Figure 1 [12], represents a layered classification of components in a WSNS that can suffer faults.

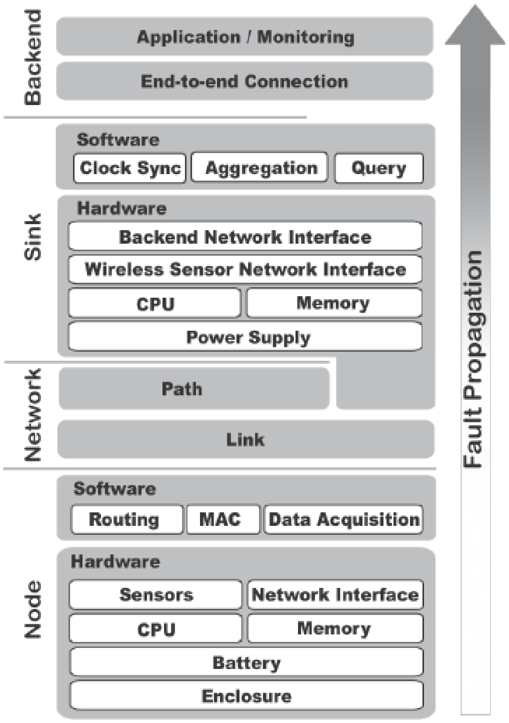

Figure 1. Fault Model for Wireless Sensor Networks

A fault in each layer of the system has the possibility to propagate to above levels. Some of the prominent sources of faults are:

A) Node Level Faults

Sensor nodes are fragile; as they may fail due to depletion of batteries [5] or nodes hardware and software malfunction and the external impacts of harsh environmental conditions on the node's enclosure leads to node faults and incorrect readings (as low battery of the node also affects the reading). The failure of cluster head nodes affects the WSNs. Sensor nodes sending incorrect values to the sink and erroneous data aggregation of cluster heads will cause the base station to receive incorrect information of an entire region of the network [12]. In other words, failure of the node's software or hardware can lead to permanent faults in the system.

B) Network Level Faults

Faults in the routing layer can lead to drop or misguided messages, or unacceptable delays. Instability of the link between nodes; as links are failure-prone in any ad hoc wireless networks, hence causing network partitions and dynamic changes in network topology. Node's dislocation, where the node is unreachable results in complete loss of data. Radio interference, path error and permanently or temporarily blockage and collision of messages can also cause the link between nodes to become faulty. Congestion may occur due to a large number of nodes and simultaneous transition from a power state to an active transmission state in response to an event-of-interest [5]. . Software bugs in routing layer can generate circular paths or simply deliver messages to the incorrect destination [12].

C) Sink Faults

On a higher level of the network a vice (sink) that collects all the data generated in the network and propagates it to the back-end system is also subject to faults. The failure of the sink leads to a massive failure of the network. A fault in the sink can prevent it to transmit tasks to the sensors as well as relay the data to the Base Station (end user). Finally, the software that stores the data collected from the network, processes it and sends it to the back-end-system, is subject to bugs that when present can lead to loss of data within the period where the fault occurred [12]. Additionally, congestion that starts in one local area can propagate all the way to the sink and affect data delivery from other regions of the network [5].

D) Faults Caused by Adversaries

Attacks by adversaries may cause node faults and consequently, leads the network to failure; because these networks are often deployed for critical applications. The deployment of sensor nodes in a hostile environment, such as in enemy territories for battlefield surveillance, can lead to a worse attack where adversaries cannot only manipulate the environment (i.e. by jamming the signal), but can also physically capture the sensor node. The lack of infrastructure and broadcast nature of the wireless medium enable adversaries to intrude into the network, and disrupt the whole functionality (e.g. routing, aggregation, etc.) of an individual sensor node [16].

2.2 Fault Management Architecture and Phases

Fault management has become a crucial function in the network management as a result of the rapid growth and complexity of the WSNs. The large size and its nature of deployment in an inaccessible and inhospitable environment have almost rendered human administrators obsolete; as a result recently automated fault management systems are being adopted. Important functions of fault-management include:

- Definition of thresholds for potential failure conditions
- Constant monitoring of system status and usage level
- General diagnostics
- Alarm and the notification of any error or malfunctions
- Tracing the location of potential and actual malfunctions
- Automatic correction of potential-problem causing conditions
- Should keep the probability of false alarm as minimum as possible.
- Recovery of failures

Fault management for WSNs is different from traditional networks. Recent research has developed several schemes and techniques that deal with different types of faults at different layers of the network. To provide resilience in faulty situations three main actions (fault detection, fault diagnosis and fault recovery) (Figure. 2) must be performed [5, 6, 12].

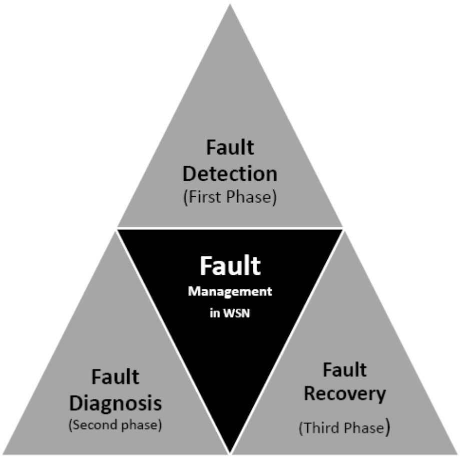

Figure 2. Fault Management in WSNs and its three phases

A) Fault Detection

Fault detection is the first phase of fault management, where an unexpected failure in the networks should be properly identified by the networks system. Fault detection in sensor networks largely depends on the type of applications and the type of failures.

B) Fault Diagnosis

Fault diagnosis is a stage in which the causes of detected faults can be properly identified and distinguished from other irrelevant alarms. A work on this phase has been done in [17-19] to achieve fault diagnosis; however, there is still a need of a more comprehensive model of faults in sensor networks to support the network systems for accurate fault diagnosis [6].

C) Fault Recovery

After fault detection and fault diagnosis; it is seen in fault recovery that how faults can be treated [5]. The failure recovery phase is the stage at which the sensor network is restructured or reconfigured, in such a way that failures or faults nodes do not impact further on network performance [6].

2.3 Fault Management System Network Architecture

Fault management in WSNs can be classified according to their management system network architecture [20, 21]: Centralized, Distributed or Hierarchical

A) Centralized Architecture

In a centralized management architecture, the base station acts as a central controller or a central manager station that collects information from the whole network and controls the entire network.

Advantages - The base station or the central manager has rich and unlimited resources, hence to perform complex management tasks, reducing the processing burden on resource constrained nodes in the sensor network. The central manager has the global knowledge of the entire network (i.e. the topology map, residual energy of the nodes, communication coverage map, etc.). Therefore, it can provide accurate and reliable management decisions.

Disadvantages - A centralized architecture incurs a high message overhead (bandwidth and energy) from data polling, and this limits its scalability. Since the whole network data traffic is towards the base station which results in high congestion rate, which degrade the performance of the network. Moreover, the central controller is the single point of potential failure. Finally, if a network is partitioned, then nodes that are unable to reach the central base station are left without any management functionality, and that node is simply isolated.

B) Distributed Architecture

Instead of having a single central controller, distributed management architecture employs multiple manager station throughout the whole network. Each manager controls a sub-region of the network and may communicate directly with other manager stations in a cooperative manner in order to perform management functions. Local processing and management reduce network bandwidth requirements and processing at the central controller.

Advantages - Distributed management has lower communication costs than the centralized one, and hence provides better reliability and energy efficiency.

Disadvantages - Distributed management algorithms may be computationally too expensive for resource constrained sensor nodes. It is also very complex and difficult to manage as compare to centralized management. Lastly, distributed approaches are very costly in terms of memory usage due to its complex computations.

C) Distributed Agent-Based Approach

A mobile agent-based management approach is an example of distributed management system implementation.

Advantages - Agents can be designed to distribute management functions in the network, e.g. an agent can relay traffic for an overloaded node in the network, and can also shift the transmission and debugging load from low-power sensor nodes to extend the network lifetime. Furthermore, the mobile agent can move easily and flexibly from one area to another to cover an area of interests.

Disadvantages - In agent-based approaches there is a need for a special node to behave as an agent and to perform management functions. Moreover, a human intervention is required to locate these agents intelligently to cover all the nodes in the network. Therefore, the network needs to be pre-configured and the human manager needs to know about the optimal location of the agents for a particular application. In agent based approaches the network experiences a significant amount of delay when the manager wants to retrieve the network states of the node, because the manager needs to wait for an agent to visit the node. Furthermore, agent-based approaches do not perform well in large-scale WSNs, because as the number of sensor node's increases. The number of agents must also be increased.

D) Hierarchical Architecture

Hierarchical management architecture is a hybrid between the centralized and distributed approach. Sub-controller or managers are distributed throughout the network in a tree shape hierarchical manner, having levels of lower and higher level of hierarchy. These managers are referred to as the Intermediate managers, manage a sub-section of a network and perform the management functions, but they don’t communicate with each other directly. There is a complete management hierarchy among the nodes, i.e. the lower-level managers pass information to its higher-level manager; and also disseminate management functions received from the higher-level manager to its sub-network. Most of the contemporary management architectures have used the clustering-based hierarchy approaches, where a common node is selected as a CH which acts as an Intermediate manager.

3. Existing Fault Management Schemes for Wireless Sensor Networks

Fault management in WSNs is different from traditional networks. Recently, researchers have developed various techniques and approaches to deal with various types of faults at different layers of the network. To provide resilience in faulty situations these three main actions (fault detection, fault diagnosis and fault recovery) must be performed [5, 6]. We categorize these existing approaches according to different phases of the fault management architecture, i.e. fault detection, fault diagnosis and fault recovery. In this section, we will discuss these phases and state of the art approaches to perform these functions. We also highlight different issues and problems in the proposed fault management approaches for WSNs in section 2.4.

Table 1 show the overall classification and comparison of existing fault management approaches and architecture. The table describes the approaches with their operation organization and types of faults they detect, diagnosis and recover from.

Schemes	Management System Organization	Types of faults & failures addressed	Action taken
Sympathy [22]	Centralized Hierarchical, Pro-active monitoring	Node self, Network faults, Sink fault, Crash & time-out omission failures	Fault Detection & Diagnosis
MANNA [23]	Centralized + Distributed Passive monitoring	Node faults	Detection, Diagnosis & Recovery
WinMS [24]	Centralized + Distributed (Hierarchical) Pro-active monitoring	Node faults (week or faulty)	Detection & Recovery
WSNMP [25]	Centralized + Distributed (Hierarchical Clustering based)	Node faults, Network faults	Detection & Recovery
Cluster-Based approach [26, 27]	Centralized + Distributed	Node faults (energy failures), Network faults (network connectivity), Permanent faults	Detection & Recovery
Passive Diagnosis of WSNs [28]	Centralized + Hierarchical, Probabilistic approach Passive monitoring	Node faults, Network faults, Transient faults	Detection, Diagnosis & Recovery
Efficient Tracing of failed nodes [29]	Centralized & Active monitoring	Node faults, Route Faults	Detection, Diagnosis & Recovery

Table 1. Fault Management Approaches Categorization

A) Centralized Architectures

In centralized approach, most of the management and maintenance tasks are carried out by the central entity, which has unlimited and powerful computing and energy resources. Centralized approaches provide accurate and reliable management decisions with reducing management burden on resource constrained sensor nodes. Examples of centralized management approaches are: SNMS [30], Sympathy (a debugging system for sensor networks) [22], WinMS (Wireless sensor network Management System) [24], and Efficient tracing of failed nodes in sensor networks, proposed by Staddon et al. [29].

A centralized sink location based scheme Sympathy [31] provides a debugging technique to detect and localize faults that may occur from interactions between multiple nodes. However, Sympathy does not provide an automatic debugging mechanism [21]. For detecting and debugging Sympathy uses a message-flooding technique to pool event data and current states (metrics) from sensor nodes. Staddon et al. [29], while tracing failed nodes in the network proposed a centralized management approach, whereby the manager monitors the health of an individual sensor node to detect node failures in the network. The base station constructs the whole map of the network topology with the help of nodes routing to update message providing a method for recovering corrupted routes.

Deb et al. [32] proposed a sensor network management framework called sNMP (Sensor Network Management Protocol). sNMP, (Sensor Network Management Protocol) framework defines sensor models (network topology, energy map and usage patterns, etc.), that represents the current state of the network and defines various network management functions. It also provides tools and algorithms for retrieving network states through the execution of different network management functions. The human manager in sNMP periodically monitors the network states, and maintains the network by identifying which part of the network has a low performance, and takes the corrective actions as necessary. The periodic monitoring of the network states helps in analyzing the network dynamics to predict potential failures and then to take preventive actions. However, the centralize-processing approach requires continuous polling of data from nodes to the base station, which puts an extra burden on energy-scarce sensor nodes [21].

However, these approaches suffer from many problems such as: insufficient scalability, availability and flexibility when a network becomes more distributed [9]. Moreover, by concentrating all the tasks around a single controller, this controller becomes a potential failure point.

B) Distributed Architectures

Instead of having a single central controller, distributed management architecture employs multiple manager station throughout the whole network. Each manager controls a sub-region of the network and may communicate directly with other manager stations in a cooperative manner in order to perform management functions. Distributed architectures encourages sensor nodes to self-manage and self-configure themselves up-to certain level. It has been verified that the more decision a node can make, the less numbers of communication messages need to be exchanged with base station. Neighbour coordination is a typical example of fault management distribution [33].

In self-managed distributed approaches, fault detection and recovery are carried out by local nodes, which involve checking and assessing their own residual energy status. Local processing and management reduce network bandwidth requirements and processing at the central controller. Examples of distributed fault management approaches are: MANNA (Management Architecture for WSNs) [23], Two-Phase (TP) self-monitoring mechanism [34] and a cellular self-organization architecture for WSNs.

MANNA provides a general framework for policy-based management system of WSNs. In MANNA, the management system collects dynamic management information, map this into WSN model, and execute management functions and services based on wireless sensor network models. MANNA implements the concept of external managers and agent nodes. A manager is located externally to the wireless sensor network where it has the global vision of the network and can perform the complex management tasks based on the information received through agent nodes. In MAANA, every node monitors its energy level and informs the manager or agent, whenever there is a state change. The manager uses this information to build the topology map and network energy model for monitoring and detecting the potential failure in the network. To detect node failure, the manager node commands the agents to execute the failure detection management service by sending a GET the operation message for retrieving node states. Without hearing from the nodes, the manager node will analyze the energy map to check whether a node has any residual energy. If so, the manager detects a failure and sends a notification to the observer. However, the scheme has a drawback, that it may be possible that a GET and GET-RESPONSE packets may be lost due to noise, which may provide false diagnostic message. Furthermore, the transmission cost for network state polling has not been considered in the approach [35]. In [36] authors proposed a distributed fault detection and recovery architecture for homogenous WSN. They divide the network in a virtual grid and regard each cell in the grid as a cluster. The design is energy-efficient and light weight with minimum communication cost and provides better reliability and energy efficiency. However, the architecture only considers permanent faults in the network.

Koushanfar et al. [13] suggested a heterogeneous back-up scheme for tolerating and healing the hardware faults of a sensor node, but this solution is not directly relevant to fault recovery in the system [6]. Marti et al. uses a technique where when a faulty node is detected, a node chooses a new neighbour to route to. Su et al. [37] introduces an adaptive and fault tolerant method for gateway assignment in WSNs. The approach is fully distributed, and it allows surviving gateways to recover for other failed gateways. Each gateway adaptively controls its region of influence based on local conditions such as remaining energy level and traffic load.

Local processing and management reduce network bandwidth requirements and processing at the central controller. However, distributed management algorithms may be computationally too expensive in terms of memory usage.

C) Hierarchical Architectures

Hierarchical management architecture is a hybrid between the centralized and distributed approaches. Sub-controller or managers are distributed throughout the network in a tree shape hierarchical manner, having levels of lower and higher level of hierarchy. Hierarchical model distributes fault management tasks according to the node management functionalities and responsibilities in the network. It splits the whole network into several regions. Each region consists of a limited number of sensor nodes. A manager node is selected to be responsible for the fault management within its region [14]. In the following sub section, we give more details about hierarchical cluster based schemes, since our design is based on a hierarchical clustering paradigm.

1) *Hierarchical Cluster-Based Schemes*

Most of the contemporary management architectures have used the clustering-based hierarchical approaches. In clustering paradigm, sensor nodes in the network are grouped together to efficiently relay the sensed data to the Sink/Base Station. Each group of sensor or cluster nodes has a cluster head and gateway node. Clustering has become an emerging technology for building scalable and energy-balanced applications for WSNs. Examples are: Distributed fault detection by using clustering mechanisms [17, 27, 38], WinMS [24] and localized fault-tolerant event boundary detection in a sensor network [39]. WinMS allows individual nodes to act as agents and perform management functions autonomously based on their neighborhood states.

In a cluster based sensor networks, when sensors are first activated, the neighbouring sensor nodes organize themselves into clusters to reduce the sensing redundancy and to avoid the reuse of scarce limited resources. To improve the clustering efficiency, many clustering protocols to have been proposed; among them Low Energy Adaptive Clustering Hierarchy (LEACH) [40] is the most famous one. LEACH is a self-organizing, adaptive clustering protocol that selects cluster heads randomly to distribute the energy load evenly among the sensor nodes in the network. The role of a cluster head is rotated randomly in order to prevent the energy drainage of a particular single sensor node. However, the randomized rotation does not take into account the current energy level of the node and may choose a node with very little remaining energy as a cluster head with the danger of fast death of that node. Furthermore, this algorithm allows only 1-hop clusters to be formed, which may lead to the formation of a large number of clusters. The approach causes a problem with energy-efficiency and scalability, because when the network size grows the cluster head will not be possible to reach the Sink or base station [33].LEACH-C (LEACH-Centralized) [33] is an improved version of LEACH, which forms clusters at the beginning of each round using a centralized decision making algorithm. LEACH-C selects cluster heads based on their location information and energy level. LEACH-C performs well, but frequent communication between the base station and sensor nodes increase communication cost and energy usage.

Hierarchical Clustering introduces an extra level of management nodes that facilitate the distribution of control over the entire network. It saves energy and reduces network contention by enabling locality of communication: nodes communicate their data to their cluster head over a short distance, while these cluster heads further forward data to their high level manager in the hierarchy or directs it to the base station [41]. Most of the existing hierarchical clustering approaches assume a single hop communication model in terms of members' nodes. For instance, Siqueia et al. [42], proposed a 3-tier hierarchical clustering architecture, which has a single hope communication model between cluster-head and sensor nodes, or between cluster-heads and the base station. They work well for small networks but their performance is heavily impaired when the number of clusters increased in large-scale sensor networks. To improve the robustness and efficiency of clustered-based scenario, Lai and Chen [43] proposed a CMATO (Cluster-Member-based fAult Tolerant mechanism) algorithm. CMATO views the cluster as a whole and takes advantage of the inter-cluster monitoring of nodes to detect the faults. When the cluster member detects a fault that is caused by the cluster head, they act co-operatively to select new cluster head to replace the failed one.

WSNMP (Wireless Sensor Networks Management Protocol) [25] is a hierarchical network management system which also uses tier architecture. In this approach, a central manager is set at the highest level of the network i.e. the sink node; the intermediate manager works at the cluster heads and management agents are the normal sensor nodes. Intermediate managers are used to distribute management functions and collect and collaborate with management data from the entire network. They execute management functions based on their local network states whereas a central network manager has the global knowledge of the network states and entire topology map. Once the topology of the network is modeled the Central Manager can reconfigure the network with minimum overhead. It also detects a fault in the network by identifying the non-response nodes and if required to reconfigure the routing path [25]. The architecture of WSNMP is shown in figure 2 [25], which represents the relationship among management services and management functionalities. WSNMP provides the method to monitor the network states by collecting management data and accordingly control and maintain the network resources. However, to build the entire topology map for the whole network incurs extra over-head and is more energy consuming for resource constrained WSNs.

2) *Self-Managed Schemes*

Self-managed fault management means that a WSN must perform fault management tasks and services with a minimum or no human intervention with the goal of promoting network productivity and quality of service [44]. The self-managed fault tolerant WSNs must be able to detect and recover from various networks and sensor faults locally in a distributed way with minimum resource utilization [24].

TP [34], Sympathy [22], and MANNA [23] focuses on fault management in WSNs. In TP each node monitors its own health and its neighbours' health, thus providing local fault detection. Sympathy provides a debugging technique to detect and localize faults that may occur from interactions between multiple nodes. MANNA performs centralized fault detection based on the analysis of gathered WSN data through agent nodes. TP, Sympathy, and MANNA focus only on fault detection and debugging, they do not provide an autonomic network reconfiguration to allow the network to recover from faults and failures [24].

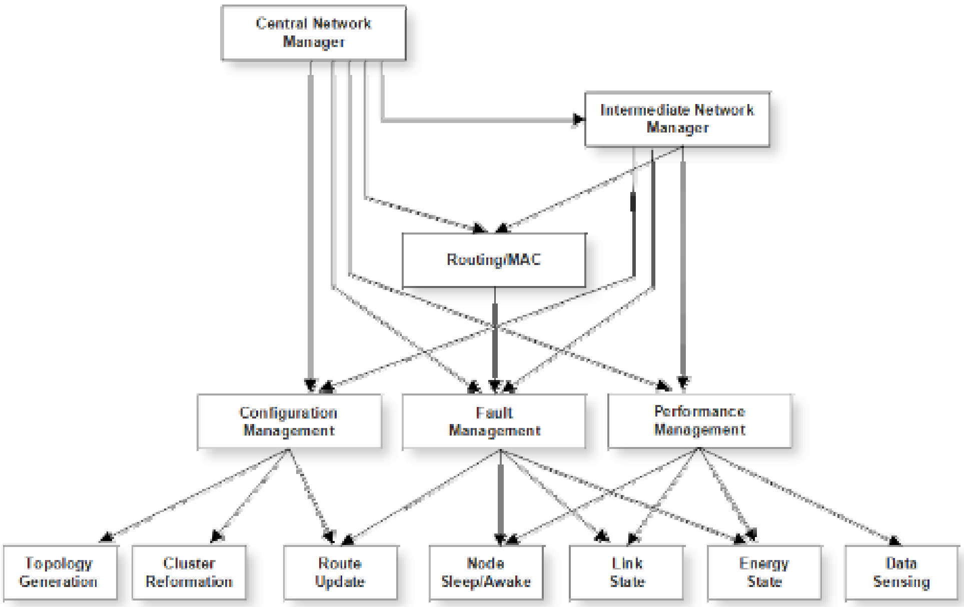

Figure 3. WSNMP Architecture

Hierarchical clustering based distributed approaches provide a major shift in the design of fault management architecture for WSNs. Management responsibilities are transferred more towards the sensor nodes, instead of a central manager, which ultimately makes the network more reliable and self-managed. Most of the schemes (centralized and distributed) discussed earlier, are not fully adaptive and self-managed. Fault management and recovery are carried out by exchanging excessive messages among nodes. To overcome this problem Yu et al. [14], proposed a biologically inspired self-managed fault management architecture for WSNs. The proposed self-managed hierarchical architecture fully distributes the management tasks among different sensor nodes in the network. The scheme introduces more self-managing functions to the sensor nodes, which encourages them to be more self-dependent on monitoring their own status instead of frequent consulting with their cluster-head. In additions, they also give a solution for faulty nodes replacement in a self-configurable WSN.

2.4 Issues in Existing Fault Management Approaches

In this section, we highlight different issues and problems existed in already proposed fault management approaches for WSNs. We believe that there is a need for comprehensive fault management architecture with a more holistic approach, which can perform fault detection, diagnosis and fault recovery on an efficient basis. It is evident from the literature survey that different approaches for fault management suffer from the following problems:

- The application-specific requirements of WSNs are varied in terms of data routing, resource utilization and communication pattern; hence, it is very challenging to apply existing fault management architecture from one application to another.
- Most existing approaches [34, 45] mainly focus on failure detection. However, there is still no comprehensive solution for fault management in WSNs from the management architecture perspective.
- Different mechanisms proposed for fault recovery [13] are not directly relevant to fault recovery in respect of the network system level management (e.g. network connectivity and network coverage area, etc.).

- Fault recovery mechanisms are mainly application specific (e.g. gateway recovery, common node recovery, etc.) and focus on a small region or individual nodes thereby are not fully scalable.
- Some decentralized approaches, e.g. Hsin *et al.* [34] require the network to be pre-configured, which is very costly for resource constrained WSNs.
- Some management frameworks require the external human manager to monitor the network management functionalities e.g. TinyDB, MOTE-VEW and sNMP.
- Some schemes [26, 36] only consider permanent faults and avoid Intermittent and Transient faults.
- Most existing approaches in WSNs isolate [46] failed or misbehaving nodes directly from the network communication, but there is no adequate fault recovery procedure available.
- There is no comprehensive description or model to distinguish various faults in WSNs that is capable of supporting the network system in achieving accurate fault diagnosis and fault recovery action.

4. Discussion

Due to energy-scarce nature of WSNS, traditional network management solutions are not suitable for WSN. Moreover, fault tolerance, reliable data dissemination and scalability also pose challenges for network management in WSNs. In the light of the literature review, it can be concluded that there is a need of designing network management solutions, which can optimize application’s quality with efficient fault management operations. Moreover, such solutions should be capable of minimizing the resource consumption and maximize the network lifetime [20, 47]. Concerning network management and related challenges in WSNs many solutions have been reported. However, existing literature reported different weaknesses in the proposed solutions. Among the proposed solutions, the architecture-based solutions classified into three categories according to their management system network architecture [20, 21] i.e. Centralized, Distributed, and Hierarchical. It is known that these approaches suffer from problems such as insufficient scalability, availability and flexibility, when the network becomes more distributed [9]. Distributed management algorithms are suitable for scalable network. However, they are considered computationally expensive in terms of memory usage.

LEACH is a self-organizing, adaptive clustering protocol that selects cluster heads randomly to distribute the energy load evenly among the sensor nodes in the network. The role of a cluster head is rotated randomly in order to prevent the energy drainage of a particular single sensor node. However, the randomized rotation does not take into account the current energy level of the node and may choose a node with very little remaining energy as a cluster head with the danger of fast death of that node. TP [34], Sympathy [22], and MANNA [23] focuses on fault management in WSNs. In TP, each node monitors its own health and its neighbors’ health, thus providing local fault detection. Sympathy provides a debugging technique to detect and localize faults that may occur from interactions between multiple nodes. MANNA performs centralized fault detection based on the analysis of gathered WSN data through agent nodes.

TP, Sympathy, and MANNA focus only of fault detection and debugging, they do not provide an autonomic network reconfiguration to allow the network to recover from faults and failures [24]. Most existing approaches [34, 45] mainly focus on failure detection. However, there is still no comprehensive solution for fault management in WSNs from the management architecture perspective. Different mechanisms proposed for fault recovery [13] are not directly relevant to fault recovery in respect of the network system level management (e.g. network connectivity and network coverage area, etc.). Fault recovery mechanisms are mainly application specific (e.g. gateway recovery, common node recovery, etc.) and focus on a small region or individual nodes thereby are not fully scalable. Some decentralized approaches, e.g. Hsin et al. [34] require the network to be pre-configured, which is very costly for resource constrained WSNs. Some schemes [26, 36] only consider permanent faults and avoid Intermittent and Transient faults. Most existing approaches in WSNs isolate [46] failed or misbehaving nodes directly from the network communication, but there is no adequate fault recovery procedure available.

It is clear from the above discussion that energy-efficient network management is an important aspect to be seen in the context of WSNs. In particular, a fault-tolerant network management scheme is required, which can detect and recover fault on an efficient basis. This in-fact is needed to support smooth operation of WSNs. Moreover, due to their nature, on-site maintenance of faults is infeasible for WSNs. Therefore, scalable self-management is crucial for the deployment of large-scale WSNs.

5. Conclusion and Future Work

The contribution of this paper is to present an in-depth critical review of some of the dominant fault management schemes of wireless sensor’s network. It has been cleared in the light of discussion that in order to incorporate fault management functionalities (such as fault detection and recovery) via an effective design into the network management infrastructure of WSNs. This is to improve their robustness, reliability and to enable a wider adoption of WSNs applications and technology. It is well known that the unique characteristics and restrictions of WSNs will be taken into account when proposing fault management architecture for WSNs. In this context, the architecture will detect and recovers from various types of faults such as permanent and potential faults. Moreover, it will tackle faults at a number of different levels (such as node level and network level, etc.) with low overhead in terms of computing bandwidth, and energy consumption. Fault management functionalities associated with our management architecture will be compared with existing approaches to measure its effectiveness and reliability. We believe this study will provide a base for developing new or to modify some existing fault management schemes. We are working towards a novel fault management structure capable of addressing weaknesses as highlighted in this work. We are committed to sharing our research findings with the ongoing research in this area.

REFERENCES

ISSN 1512-1232

1. F. Koushanfar, M. Potkonjak, and A. Sangiovanni-Vincentelli, "Fault Tolerance in Wireless Sensor Networks, Book chapter in Handbook of Sensor Networks," vol. 36, Section VIII, p. 24, 2004.
2. W. Dargie and C. Poellabauer, *FUNDAMENTALS OF WIRELESS SENSOR NETWORKS THEORY AND PRACTICE*: A John Wiley and Sons, Ltd., Publication, 2010.
3. K. Sohraby, D. Minoli, and T. Znati, *Wireless Sensor Networks: Technology, Protocols, and Applications*. Hoboken, New Jersey: John Wiley & Sons, Inc, 2007.
4. C. Cordeiro and D. P. Agrawal, *Ad hoc & sensor networks, Theory and Applications*: World scientific publishing, 2006.
5. L. Paradis and Q. Han, "A Survey of Fault Management in Wireless Sensor Networks," *Journal of Network and System Management, Springer Science + Business Media, LLC,* vol. 15, pp. 171-190, June 2007.
6. Y. Mengjie, H. Mokhtar, and M. Merabti, "Fault Management in Wireless Sensor Networks," *IEEE Wireless Communications,* vol. 14, pp. 13-19, 2007.
7. L. M. d. Souza, H. Vogt, and M. Beigl, "A survey on Fault Tolerance in Wireless Sensor Networks," *[Online]. Available: http//:digbib.ubka.uni-karlsruhe.de/volltexte/documents/11824,* 2007.
8. H. Liu, A. Nayak, and I. Stojmenovic, "Fault-Tolerant Algorithms/Protocols in Wireless Sensor Networks," in *Guide to Wireless Ad Hoc Networks*, ed: Springer-Verlag London, 2009, pp. 265-295.
9. M. Al-Kasassbeh and M. Adda, "Network fault detection with Wiener filter-based agent," *Journal of Network and Computer Applications,* vol. 32, pp. 824-833, 2009.
10. H. Bakht, M. Merabti, and R. Askwith, "Mobile ad-hoc on-demand data delivery protocol," in Proceedings of the 3rd Annual Post Graduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting, Liverpool, UK., 2002.
11. H.bakht, "Mobile ad-hoc on-demand data delivery protocol," *IEFT draft,* June 2001.
12. L. M. d. souza, H. Vogt, and M. Beigl, "A survey on Fault Tolerance in Wireless Sensor Networks," *www.digbib.ubka.uni-karlsruhe.de/volltexte/documents/11824*. , n.d.
13. F. Koushanfar, M. Potkonjak, and A. Sangiovanni-Vincentell, "Fault tolerance techniques for wireless ad hoc sensor networks," in *Proceedings of the IEEE Sensors*, 2002, pp. 1491-1496.
14. M. Yu, H. Mokhtar, and M. Merabti, "Self-Managed Fault Management in Wireless Sensor Networks," in Proceedings of the The Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM '08), 2008, pp. 13-18.
15. G. Gupta and M. Younis, "Fault-Tolerant Clustering of Wireless Sensor Networks," in *IEEE Wireless Communications and Networking, WCNC'03*, 2003, pp. 1579-1584.
16. L. B. Ruiz, I. G. Siqueira, L. B. e. Oliveira, H. C. Wong, J. M. S. Nogueira, and A. A. F. Loureiro, "Fault Management in Event Driven Wireless Sensor Networks," presented at the International Workshop on MSWiM’04, Venezia, Italy, Oct. 2004.
17. A. T. Tai, K. S. Tso, and W. H. Sanders, "Cluster-based failure detection service for large-scale ad hoc wireless network applications," in *Proceedings of the International Conference on Dependable Systems and Networks*, 2004, pp. 805-814.
18. C. Hsin and M. Liu, "Self-monitoring of wireless sensor networks," *Computer Communications,* vol. 29, pp. 462-476, 2006.
19. C. Thomas, K. S. Kewal, and R. Parameswaran, "Fault Tolerance in Collaborative Sensor Networks for Target Detection," *IEEE Trans. Comput.,* vol. 53, pp. 320-333, 2004.
20. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks," *IEEE Communication Magazine,* pp. 102-114, 2002.
21. W. L. Lee, A. Datta, and R. Cardell-Oliver, *Network Management in Wireless Sensor Networks*: Handbook on Mobile Ad Hoc and Pervasive Communications American Scientific Publishers, 2006.

22. N. Ramanathan, E. Kohler, L. Girod, and D. Estrin, "Sympathy: a debugging system for sensor networks [wireless networks]," in *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks*, pp. 554-555.
23. L. B. Ruiz, J. M. Nogueira, and A. A. F. Loureiro, "MANNA: a management architecture for wireless sensor networks," *IEEE Communications Magazine,* vol. 41, pp. 116-125, 2003.
24. 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, The University of Western Australia, CSSE Technical Report UWA-CSSE-06-001,* June 2006.
25. M. M. Alam, M. Mamun-Or-Rashid, and C. S. Hong, "WSNMP: A Network Management Protocol for Wireless Sensor Networks," presented at the 10th International Conference on Advanced Communication Technology, (ICACT'08) 2008.
26. G. Venkataraman, S. Emmanuel, and S. Thambipillai, "A Cluster-Based Approach to Fault Detection and Recovery in Wireless Sensor Networks," in *Proceedings of the 4th International Symposium on Wireless Communication Systems, (ISWCS'07)*, 2007, pp. 35-39.
27. C. Yao-Chung, L. Zhi-Sheng, and C. Jiann-Liang, "Cluster based self-organization management protocols for wireless sensor networks," *IEEE Transactions on Consumer Electronics,* vol. 52, pp. 75-80, 2006.
28. K. Liu, M. Li, Y. Liu, M. Li, Z. Guo, and F. Hong, "Passive diagnosis for wireless sensor networks," in *Proceedings of the 6th ACM conference on Embedded network sensor systems, Sensys'08*, Raleigh, NC, USA, 2008, pp. 113-126.
29. S. Jessica, B. Dirk, and D. Glenn, "Efficient tracing of failed nodes in sensor networks," presented at the Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications, Atlanta, Georgia, USA, 2002.
30. G. Tolle and D. Culler, "Design of an application-cooperative management system for wireless sensor networks," presented at the Proceeedings of the Second European Workshop on Wireless Sensor Networks, 2005.
31. N. Ramanathan, E. Kohler, L. Girod, and D. Estrin, "Sympathy: a debugging system for sensor networks [wireless networks]," in *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, 2004, pp. 554-555.
32. B. Deb, S. Bhatnagar, and B. Nath, "Wireless Sensor Networks Management " *http://www.research.rutgers.edu/□bdeb/* sensor networks.html, 2005.
33. A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Communications,* vol. 30, pp. 2826-2841, 2007.
34. C. Hsin and M. Liu, "A Two-Phase Self-Monitoring Mechanism for Wireless Sensor Networks," *Journal of Computer Communications special issue on Sensor Networks,* vol. 29, pp. 462-476, February 2006.
35. M. Yu, H. Mokhtar, and M. Merabti, "A Survey on Fault Management in Wireless Sensor Networks," presented at the 8th Annual Postgraduate Symposium on the Convergence of Telecommunication Networking and Broadcasting (PGNet'07), Liverpool John Moores University, UK, 2007.
36. M. Asim, H. Mokhtar, and M. Merabti, "A Fault Management Architecture for Wireless Sensor Network," presented at the International Wireless Communications and Mobile Computing Conference, IWCMC '08. , 2008.
37. W. W. Su and L. Sung-Ju, "An adaptive and fault-tolerant gateway assignment in sensor networks," in *Proceedings of the IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, 2004, pp. 576-578.
38. G. Venkataraman, S. Emmanuel, and S. Thambipillai, "A Cluster-Based Approach to Fault Detection and Recovery in Wireless Sensor Networks," presented at the 4th International Symposium on Wireless Communication Systems, ISWCS'07, 2007.

39. M. Ding, D. Chen, K. Xing, and X. Cheng, "Localized fault-tolerant event boundary detection in sensor networks," presented at the 24th Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM'05, Miami, 2005.
40. W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," presented at the 33rd Annual Hawaii International Conference on System Sciences, 2000.
41. V. Srikanth and I. R. Babu, "Cluster Head Selection for Wireless Sensor Networks: A Survey," *The Icfai University Journal of Information Technology,* vol. 5, pp. 44-53, March 2009.
42. [42] I. G. Siqueira, L. B. Ruiz, A. A. F. Loureiro, and J. M. Nogueira, "Coverage area management for wireless sensor networks," *Int. J. Netw. Manag.,* vol. 17, pp. 17-31, 2007.
43. L. Yongxuan and C. Hong, "Energy-Efficient Fault-Tolerant Mechanism for Clustered Wireless Sensor Networks," in *Proceedings of 16th International Conference on Computer Communications and Networks (ICCCN'07)*, 2007, pp. 272-277.
44. L. B. Ruiz, T. R. M. Braga, F. A. Silva, H. P. A. Assuncao, J. M. S. A. Nogueira, and A. A. F. A. Loureiro, "On the design of a self-managed wireless sensor network," *IEEE Communications Magazine,* vol. 43, pp. 95-102, 2005.
45. A. Peffig, R. Szewczy, J. D. Tygar, Victorw, and D. E. Culler, "SPINS: Security Protocols for Sensor Networks," in *Proceedings of the ACM MobiCom' 01*, Rome, Italy, 2001, pp. 189-199.
46. S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking*, Boston, Massachusetts, United States, 2000, pp. 255-265.
47. H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*: John Wiley & Sons, Ltd, West Sussex, England, 2005.

Article received: 2012-02-05