A Survey on Fault Detection and Fault Tolerance in Wireless Sensor Networks

R. V. Kshirsagar, Ph.D
PCE Nagpur

A.B. Jirapure,
PCE Nagpur

ABSTRACT

Wireless Sensor Networks (WSNs) have the potential of significantly enhancing our ability to monitor and interact with our physical environment. Wireless Sensor networks (WSN) are inherently fault-prone and the reliability of WSN is affected by faults that may occur due to various reasons such as malfunctioning hardware and software glitches, dislocation or environmental reasons. It is necessary for the WSN to be able to detect faults early and initiate appropriate recovery action to maintain the quality of service (QoS) of the wireless sensor networks. In this paper, we address these challenges by surveying existing fault detection and fault management approaches in WSN's. It is observed that the main challenge for management of WSN is to provide efficient and reliable fault tolerance (FT) mechanism while conserving the limited resources of the network**.**

KEYWORDS*:* wireless sensor network; fault detection; fault management; fault tolerance;

1. INTRODUCTION

Wireless sensor network (WSNs) consists of spatially distributed autonomous sensor nodes that collaborate with each other to collect the environmental measurements. Typical examples include temperature, light, sound, and humidity. These sensor readings are transmitted over a wireless channel to a running application that makes decisions based on these sensor readings.

Many applications have been proposed for wireless sensor networks, and many of these applications have specific quality of service (QoS) requirements.

WSN sensor nodes are typically mass produced and are often deployed in unattended and hostile environments making them more susceptible to failures than other systems [7]. Additionally, manual inspection of faulty sensor nodes after deployment is typically impractical. Nevertheless, many WSN applications are mission-critical, requiring continuous operation. Thus, in order to meet application requirements reliably, WSNs require fault detection and fault-tolerance (FT) mechanisms.

Even though FT is a well studied research field for VLSI based system, fault detection and FT for WSNs are relatively unstudied. Additionally, fault detection and FT for WSNs have added complexities due to varying FT requirements across different applications. For instance, mission critical applications (e.g., security and defense systems) have very high reliability requirements whereas non-mission critical applications (e.g., ambient conditions monitoring applications) typically have relatively low reliability requirements.

The main aim of this paper is to study the various approaches of fault detection and fault tolerance WSNs and its future prospects. To achieve this aim, we address this challenge by surveying existing approaches and providing a comprehensive overview of fault tolerance in WSNs. Existing approaches of fault management in are various in forms of architectures [5,6,8], protocols [1,3], detection algorithm [4,9-11] or detection decision fusion algorithm [13-16]. M.Yu, H.Mokhtar, and M.Merabti [26] classify these approaches into three phases, such as fault detection, fault diagnosis and recovery, as an appropriate fault management process. In the first section we discuss this classification. The rest of this paper is organized as follows: in Section II, we survey the existing fault detection approaches in WSNs; Section III discusses fault diagnosis approaches; In Section IV, it examines existing fault recovery approaches; Section V classifies existing management architectures used to support fault management in WSNs. Finally, we outline few potential future challenges of fault detection and fault tolerance in WSNs.

2. FAULT DETECTION

The goal of fault detection is to verify that the services being provided are functioning properly. It is the first phase of fault management, where an unexpected failure should be properly identified by the network system before generating great damage to the network. The existing failure detection approaches in WSNs are classified into two types: centralized and distributed approach.

A. Centralized Approach

In Centralized approach geographically or logically centralized sensor node identifies the failed or misbehaving nodes in WSNs. This centralized node may be a base station or central controller or manager having unlimited resources (e.g. energy) and is able to execute a wide range of fault management mechanism. It is believed that the network lifetime can be extended if complex management work and message transmission can be shifted onto the central node.

The central node normally adopts an active detection model to retrieve states of the network performance and individual sensor nodes by periodically sending requests (or queries) into the network. It analyzes this information to identify and localize the failed or suspicious nodes. More specifically, Sympathy [20] is a debuging tool which uses a message-flooding approach to pool event data and current states (metrics) from sensor nodes. With this information Sympathy is able to detect crash, timeout and omission failures and identify the faults that generated failure. In order to minimize the number of communication messages nodes must send and conserve node energy, a Sympathy node can selectively transmit important events to the Sympathy sink node. While, author J. Staddon et al., [19] propose an algorithm that put burden of detecting and tracing failed nodes to the base

station. At first node learn the network topology and send their portion of the topology information (i.e. node neighbor list) to the base station. Thus, the base station can construct the entire network topology by integrating each piece of network topology information embedded in, route update messages. These messages are normally forwarded by the execution of some well-known routing-discovery protocols. Once the base station knows the network topology, the failed nodes can be efficiently traced using a simple divide-and-conquer strategy.

Note that this approach assumes each node has a unique identification number and the base station is able to directly transmit messages to any node in the network. This typically requires the nodes to send additional messages, and it is consequently very expensive. Also this approach is not applicable to event driven WSN as it send the massages only when there is an event in the network.

As a summary [25], the centralized approach is efficient and accurate to identify the network faults in certain ways. However, resource-constrained sensor networks can not always afford to periodically collect all the sensor measurements and states in a centralized manner. A distinctive problem of this approach is that the central node easily becomes a single point of data traffic concentration in the network, as it is responsible for all the fault detection and fault management. This subsequently causes a high volume of message traffic and quick energy depletion in certain regions of the network, especially the nodes closer to the base station. They take extra burdens for forwarding the communication messages from other nodes. This approach will become extremely inefficient and expensive in consideration of a large-scale sensor network.

B. Distributed Approach

In distributed approach the concept of local decision-making get introduced i.e. fault management responsibility evenly distributes into the network. The goal of it is to allow a node to make certain levels of decision before communicating with the central node. It believes the more decision a sensor can make, the less information needs to be delivered to the central node. In the other word, the control centre should not be informed unless there is really a fault occurred in the network. Examples of such development are: node fault self-detection and self-correction on its hardware physical malfunction (i.e. sensor, battery, RF transceiver) [2, 21], failure detection via neighbor coordination [4, 9-11, 22], utilization of WATCHDOG [1] to detect misbehaving neighbor.

1) Node Self-detection: In many cases, nodes can identify possible failures by performing self diagnosis. In, [21] propose a self detection model to monitor the malfunction of the physical components of a sensor node via both hardware and software interface. Using a similar approach, faulty nodes could detect when they are being moved to a different location. Self-detection of node failure in [2] is somehow Straight forward as the node just observes the binary outputs of its sensors by comparing with the pre-defined fault

2).Neighbor Coordination: Failure detection via neighbor coordination is another example of fault management distribution. Nodes coordinate with their neighbors to detect and identify the network faults (i.e. suspicious node or abnormal sensor readings) before consulting with the central node. For example, in a decentralized fault diagnosis system [22], a sensor node can execute a localized diagnosis algorithm in steps and suspicious (or failed) nodes can be identified via comparing its sensor readings with neighbors' median readings.

With this motivation, Min et al., [10] developed a localized algorithm to identify suspicious node whose sensor readings have large difference against the neighbors. Although this algorithm works for large size of sensor networks, the probability of sensor faults needs to be small. If half of the sensor neighbors are faulty and the number of neighbors is even, the algorithm cannot detect the faults as efficient as expected. In addition, this approach also requires each sensor node to be aware of its physical location by equipped with expensive GPS or other GPS-less technology. While, in [11], Jinran Chen., improved such kind of approach by proposing new algorithm Distributed fault detection (DFD), which does not require node physical position. This algorithm can still successfully identify suspicious nodes even when half neighbors are faulty. It chooses the GD sensor in the network, and uses its best sensor results to diagnose other sensors' status. This information can be further propagated through the entire network to diagnose all other sensors as good or faulty. This approach (DFD) has some shortcomings such as the fault detection accuracy will decrease rapidly in the case of the number of neighbor nodes to be diagnosed is all small and the node's failure ratio is high. High fault detection accuracy can be reached only when it is applied to the sensor network with many neighbors of nodes to be diagnosed. Also if no GD sensor is determined while executing algorithm then, the network can not determine the sensor nodes' status at all and the algorithm will fail. It is essential at least one sensor node is diagnosed as GD, can this algorithm continue to execute and determine more sensors' status.

3) Clustering Approach: Clustering [24] has become an emerging technology for building scalable and energy balanced applications for WSNs. In [6], author derives an efficient failure detection solution using a cluster-based communication hierarchy to achieve scalability, completeness, and accuracy simultaneously. In this approach, the entire network split into different clusters and cluster head is appointed for each cluster. A cluster head has more resources than other cluster members. This cluster head perform major task of fault management. Intra cluster heartbeat diffusion is adopted to identify failed nodes in each cluster. Cluster head detects the suspicious nodes by exchanging heartbeat messages in an active manner with one hop neighbors (in a same radius transmission range). By analyzing the collected heartbeat information, the cluster head finally identifies failed nodes according to a pre-defined failure detection rule. Further, if a failure is detected, the local detected failure information can be propagated to all the clusters. It makes the local failure detection to be aware of the changes of network conditions or the overall objectives While, Ruiz et al., [23] adopt an event-driven detection via a manager-agent model supported by management architecture MANNA [5]. A manager is located externally to the WSN where it has a global vision of the network and can perform complex management tasks and analysis that would not be possible inside the network. Every node checks its energy level and sends a message to the manager. The manager then uses this information to build topology map and network energy model for monitoring and detecting the potential failure of the network in future.

In this random distribution and limited transmission range capability of common-node and cluster-heads provides no guarantee that every common-node can be connected to a cluster head. If a cluster head does not receive a reply from

any node then it considers that node is faulty. This violets the rule of accuracy as good nodes diagnose as faulty. In addition, failure of cluster head limits the accessibility to the nodes under its supervision. Again transmission costs for network state polling has not been considered in this approach.

3. FAULT RECOVERY

Basically, the failure recovery phase is the stage at which the sensor network is restructured or reconfigured, in such a way that failures or faulty nodes do not impact further on network performance. Efficient fault recovery techniques enable WSN to continue operating according to their specifications even if faults of certain type are present. Most existing approaches in WSNs isolate failed (or misbehaving) nodes directly from the network communication layer (e.g. the routing layer). For example, in the technique of Marti et al. [1], after the faulty neighbor is detected, a node chooses a new neighbor to route to. Thus, a network should be k- connected, which allows k-1 nodes to fail while the network would still be connected. Bredin [26] proposes algorithm that calculates the minimum amount of additional nodes and their positions that guarantee k-connectivity between nodes. Staddon et al. [19] proposed two approaches of resuming the network routing paths from the silent nodes (i.e. failed nodes), which are detected in each network routing update. Instead of taking passive recovery actions as mentioned before, the proactive action is also considered as a novel approach to prevent the potential faults in the network. As from the approach of WinMS [8], the central manager detects the network region with weak health (e.g. low battery power) by comparing the current network state (including individual nodes) with a historical network information model (e.g. an energy map or topology map). It takes proactive action by instructing nodes in that area to send data less frequently for node's energy conservation. Koushanfar et al. [2] suggested a heterogeneous back-up scheme for tolerating and healing the hardware malfunctions of a sensor node. This solution is not directly relevant to fault recovery in respect of the network system management

4. FAULT MANAGEMENT ARCHITECTURE

Fault management architectures can be classified into centralized, distributed and hierarchical models.

Centralized model, In approach a central controller is usually responsible for fault maintenance of the overall network. In order to construct the global view of the network, central controller keeps updating nodes states by message exchange. The central controller aggregates these data into information models, in terms of metrics [19], topology model [18], topology and energy map [8], WSN models [23], and cluster topology model [6]. The central node identifies any faulty or suspicious nodes by comparing the current node states against those historical information models. It is seen to be accurate solution to identify fault. However, this centralized architecture will become less efficient and more energy-expensive in consideration of large-scale sensor networks. The message flooding of such approach may greatly consume node energy because of frequent in-network message exchanges.

Distributed model, splits the entire network into several sub-regions, and distributes fault management tasks evenly into individual regions. Each region employs a central manager. Manager is responsible for monitoring and detecting failure in its region. It is also able to directly communicate with other managers in a coordination fashion for fault detection. As a result, the central controller of the overall network only needs to monitor a very small number of sensors in the network. This design conserves node energy by lessening in-network communication messages, and also enhances the system response time towards events occurred in the network. Many existing approaches have adopted Clustering to group nodes into different regions. Clustering allows sensors to efficiently coordinate their local interactions in order to achieve global goals.

Hierarchical model, it is a hybrid between the centralized and distributed approach. It uses intermediate managers to distribute manager functions. Each manager is responsible to manage nodes in its sub-network and report to its high-level central manager. In a three-layered hierarchical sensor network structure [13], cluster is adopted to reduce the number of sensor nodes monitoring the events occurred in the network. They believe only a tiny fraction of the sensors in the network can accurately detect the target events; while most sensors measurements are just pure noises. Sensors only send data of detected events to their corresponding cluster heads instead of transmitting to a far-away central fusion centre. The cluster head will make a decision about the fault events occurred within that sub-region. The decisions from cluster heads will be further transmitted to the fusion centre to inform it if there is a target or event in specific sub-regions. In order to accomplish fault management objectives in a reliable and energy-efficient way, Ying et al., [7] propose hierarchical mobile agent-based policy management architecture for sensor networks. In which, Policy Manager (PM) at the highest level, Local Policy Agent (LPA) which manages a sensor node, Cluster Policy Agent (CPA) as an intermediate management component between PM and LPA. The management commands are always propagated from the PM to CPAs to LPAs.

5. CONCLUSION

Wireless sensor network has gradually emerged as a cutting-age technology to develop new wireless applications in the 21st centaury. To achieve this goal, a robust fault management technique for the WSN is essential. In this paper we provided a thorough investigation of existing fault detection and fault management approaches in WSN. We studied the problem of fault detection and recovery, surveying different techniques currently applied in WSN research.

Through the classification proposed it is possible to compare the different techniques, identifying strong and weak points each of them. This allows for a correct selection of techniques that are more suitable to specific applications. By applying this classification we were able to verify that current approaches provide mechanisms for overcoming faults in WSN only in specific scenarios and applications. However no approach provides extensive fault tolerance support covering all types of faults that a WSN node is exposed to. Additionally there does not exist any model for fault tolerance (FT) sensor nodes, nor does there exist any model for characterizing WSN fault tolerance (FT) metrics such as reliability, MTTF (Mean Time To Failure) MTTR (Mean Time To Repair) . In the future work, we will try to propose an efficient fault tolerance mechanism for WSN which will cover all types of faults that a node is exposed to.

REFERENCES

[1] Sergio Marti, T.J.Giuli, Kevin Lai, Mary Baker. itigating Routing Misbehavior in Mobile Ad Hoc Networks. in 6th

International. Conference on Mobile Computing and Networking. 2000. Boston,Massachusetts

[2] Farinaz koushanfar, Miodrag Potkonjak, Alberto Sangiovanni- Vincentelli. Fault Tolerance Techniques for Wireless Ad Hoc Sensor Networks. 2000.

[3] A.Perrig, R.Szewczyk, V.Wen, D.Culler, J.D.Tygar. SPINS: Security protocols for sensor networks. in ACM MobiCom'01. 2001. Rome, Italy: ACM Press.

[4] Chihfan Hsin, Mingyan Liu. A Distributed Monitoring Mechanism for Wireless Sensor Networks. in 3rd workshopo on Wireless Security. 2002: ACM Press.

[5] Linnyer Beatrys Ruiz, Jose Marcos S. Nogueira, Antonio A.F. Loureiro, MANNA: A Management Architecture for Wireless Sensor Networks. IEEE Communications Magazine, 2003. 41(2): p. 116-125.

[6] Ann T. Tai, Kam S. Tso, William H. Sanders. Cluster-Based Failure Detection Service for Large-Scale Ad Hoc Wireless Network Applications in Dependable Systems and Networks DSN ' 04. 2004.

[7] Z.Ying, X.Debao. Mobile Agent-based Policy Management for WSN. in WCNM. 2005.

[8] Winnie Louis Lee, Amitava Datta, Rachel Cardell-Oliver, WinMS: WSN-Management System, An Adaptive Policy- Based Management for WSN. 2006, UWA, aUSTRALIA.

[9] C. Hsin, Mingyan Liu, Self-monitoring of WSN. Computer Communications, 2005. 29: p. 462-478.

[10] Min Ding, Dechang Chen, Kai Xing, Xiuzhen Cheng. Localized Fault- Tolerant Event Boundary Detection in Sensor Networks. in INFOCOM 2005.

[11] Jinran Chen, Shubha Kher, Arun Somani. Distributed Fault Detection of Wireless Sensor Networks. in DIWANS'06. 2006. Los Angeles, USA: ACM Pres.

[12] Xuanwen Luo, Ming Dong, Yinlun Huang, On Distributed Fault- Tolerant Detection in Wireless Sensor Networks. IEEE Transactions on Computers, 2006.

[13] XuanwenLuo, Ming Dong, Yinlun Huang, Optimal Fault-Tolerance Event Detection in WSN.

[14] Ruixin Niu, Pramod K.Varshney, Distributed Detection and Fusion in a Large Wireless Sensor Network of Random Size. Eurasip Journal on Wireless Communications and Networking, 2005. 4: p. 462-472.

[15] Pramod K.Varshney Ruixin Niu, Dale Klamer Michael Moore. Decision Fusion in a Wireless Sensor Network with a Large Number of Sensors. in IF04. 2004. Stockholm, Sweden.

[16] Thomas Clouqueur, Kewalk, Saluja, Parameswaran Ramanathan, Fault Tolerance in Collaborative Sensor Networks for Target Detection. IEEE Transactions on Computers, 2004. 53(3): p. 320-333.

[17] Tsang-Yi Wang, Yunghsiang S.Han, Pramod K.Varshney, Po-Ning Chen, Distributed Fault-Tolerant Classification in Wireless Sensor Networks. IEEE Journal on Selected Areas in Communications, 2005. 23(4): p. 724-734.

[18] S..Tanachaiwiwat, P.Dave, R.Bhindwale, A. Helmy, Secure Locations: routing on trust and isolating compromised sensors in location-aware sensor networks.

[19] Jessica Staddon, Dirk Balfanz, Glenn Durfee. Efficient Tracing of Failed Nodes in Sensor Networks. in First ACM International Workshop on Wireless Sensor Networks and Applications. 2002. Altanta, GA, USA: ACM.

[20] NIthya Ramanathan, Kevin Chang, Rahul Kapur, Lewis Girod, Eddie Kohler, Deborah Estrin. Sympathy for the Sensor Network Debugger. In 3rd Embedded networked sensor systems. 2005. San Diego, USA: ACM Press.

[21] S Harte, A Rahman, K M Razeeb. Fault Tolerance In Sensor Networks using Self-Diagnosing Sensor Nodes. in IE2005. 2005: IEEE.

[22] A. Sheth, C. Hartung, Richard Han. A Decentralized FaultDiagnosis System for WSN. in 2nd M bile Ad Hoc and Sensor Systems. 2005. Washington, USA.

[23] Linnyer Beatrys Ruiz, Isabela G.Siqueira, Leondardon B. e Oliveria, Hao Chi Wong, Jose Marcos S. Nogueira, Antonio A.F. Loureiro. Fault management in event-driven wireless sensor networks. in International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems. 2004. Venice, Italy: ACM Press.

[24] Deborah Estrin, R. Govindan, J. Heidemann, S. Kumar. Next Century Challenges: Scalable Coordination in Sensor Networks. In ACM/IEEE International Conference on Mobile Computing and networking. 1999.

[25] M.Yu, H.Mokhtar, and M.Merabti "A Survey on Fault Management in WSN" ISBN: 1-9025-6016-7, 2007

[26] J.L. Berdin, E. Demaine, "Deploying Sensor network with guarantied capacity and fault tolerance" In the proceeding of 6 international symposiums, pages 309-319.