Fault Tolerance Models in Ad Hoc Wireless Sensor Networks

Maram Al-Shablan, MSc. Student
Computer Science Department
King Saud University
Riyadh, Saudi Arabia
431920608@student.ksu.edu.sa

Soha S. Zaghloul, PhD
Computer Science Department
King Saud University
Riyadh, Saudi Arabia
smekki@ksu.edu.sa

***Abstract*—Ad Hoc wireless sensor networks are widely used in different domains. These include domestic uses such as personal security and critical ones such as military and medical applications. Unfortunately, sensors are in their nature prone to errors. Also, they are resources constraints. Efficient fault management techniques are therefore necessary. This paper surveys the different models of fault management in literature: central, cellular and hierarchical approaches. We also illustrate the advantages and drawbacks of each model.**

Keywords—ad hoc wireless sensor networks; fault tolerance models; survey.

I. INTRODUCTION

An Ad Hoc wireless Sensor Network (WSN) is a group of sensors connected remotely. Recently, Ad Hoc WSNs are broadly used in many applications in ubiquitous and pervasive computing. For example, they are used to replace human beings in risky environments such as volcanoes and mines to measure specific metrics. These are then sent to the base station to decide upon required actions and issue relevant commands accordingly. In addition, patients who may fall in critical healthy crisis use sensors to be remotely connected to hospitals; they are therefore continuously monitored by health centers so that emergency measures take place quickly whenever needed. Other applications fall under military, domestic, and educational areas.

Unfortunately, sensors are characterized by limited computation power, little memory capacity and low energy levels. In addition, they are prone to different types of errors. Hardware errors may be caused by energy depletion or physical damage. Software errors may result from bugs that cause the sensor to send data continuously to the base station. In addition, the nature of sensors makes it prone to environmental noise and radio interference. On the other hand, packets may be lost due to traffic congestion or network partitioning [1]. This makes the design of fault management techniques in Ad Hoc WSN more challenging than in traditional networks.

Accordingly, Fault Tolerance Techniques are important in order to maintain the network connectivity despite the failure of one or more nodes. Since sensors are limited in computation power, memory capacity and energy level, then designing an effective fault tolerant technique imposes a great challenge. The applied technique should have the least possible overhead on the sensor to avoid fast energy depletion. In addition, it should use the least possible memory capacity. Since computation power takes off the energy level, then the former should be minimized. On the other hand, it is common to add new nodes to the network during its operation. Therefore, the developed technique should be scalable, adaptable to updated configurations, and flexible to network dynamics. In the first place, the devised technique should be accurate enough to detect any possible error that occurs in the network.

Generally speaking, fault management techniques go through four phases: fault detection, fault identification, fault recovery and fault prediction. In the first phase, the network becomes aware of the existence of a fault. The second phase depicts the erroneous sensor node. The third phase overcomes the problem. Fault prediction is a smart way to expect that a fault is nearly to take place. In fact, not all fault management techniques in the literature do the four afore-mentioned phases. Algorithms for error detection and identification are covered in [2][3][4]. References [5][6] present fault recovery methods after detecting and identifying the error in an Ad Hoc WSN. Finally, a comprehensive fault management technique for an Ad Hoc WSN that includes the four afore-mentioned phases is proposed in [7].

The fault management architectures presented in the literature may be broadly classified into three models: central, distributed and hierarchical. Sections II, III and IV discuss these models respectively. Section V is a thorough comparison between the three models. Then, we conclude the paper in Section V.

II. Central Scheme

In a central scheme, all the sensor nodes of the network are supervised by a central manager. This is usually the sink node. Refer to Fig. 1. At the installation phase, the central manager sets a global view of the whole network by exchanging short messages with all sensors. Later, the central manager checks periodically for the sensors. If it doesn't receive a response from a node, then the latter is considered faulty.

The central scheme is simple in its design and implementation. Moreover, it is highly accurate in identifying faults within the network. However, since all messages target the same destination –the sink node-, a bottleneck is highly probable to occur in the network. This will lead to a decrease in the network efficiency. For big networks, it would be expensive that the central manager periodically exchanges messages with a large number of nodes. Therefore, it should be continuously provided with energy.

MANNA [8] is an example of the centralized scheme. The algorithm applies only the fault detection and identification phases, with no recovery. However, it classifies nodes into "manager" or "agent". Having several "managers" therefore alleviates the bottleneck problem.

FlexiMAC [9] is another fault-tolerant protocol which is based on the MAC protocol. It follows the centralized scheme. The main contribution of FlexiMAC is that it is energy-efficient since nodes send and/or receive only during their scheduled time slots. On the other hand, FlexiMAC is proven to be scalable and robust to network dynamics. Moreover, it implements error recovery in the network.

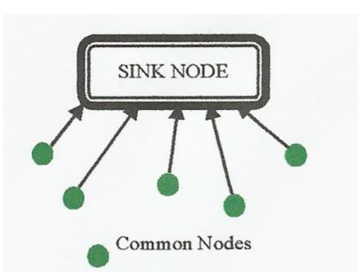

Figure 1. Central Approach

In addition, Sympathy [10] is a fault identification protocol that classifies the sensors into "sink" and "node". PAD [11] is another centralized fault management protocol.

III. Distributed Scheme

In the distributed model, the network is virtually divided into regions. Sensor nodes may be either a "common" node or a "region manager". In the installation operation, each manager collects information about the common nodes that fall in its cluster by exchanging short messages. This information usually includes the node ID, its location and its energy level. Then, the region manager checks periodically for the statuses of the nodes in order to detect any faults. Usually, the nodes with the highest energy-level are selected as region managers. On the other hand, managers communicate with each other to detect a fault of a region manager. All region managers have direct communication with the command node. Refer to Fig. 2.

The distributed scheme is suitable for scalable networks. In addition, it overcomes the drawbacks of the traffic bottleneck and the fast energy depletion of the central manager found in the central scheme. However, its main disadvantage shows if a region header falls down. This may lead to network partition. Therefore, many techniques are developed to resolve this problem.

In [12], a cluster-based fault tolerance technique is devised. The algorithm goes through three stages: 1) selection of cluster heads (CHs), 2) formation of clusters, and 3) fault detection and identification. In the first stage, nodes whose energy level exceeds a predefined threshold are selected to be CHs. In the second stage, candidate CHs broadcast messages to all nodes. Each common node selects the closest CH as its manager and replies with a short message including its information (ID, location, and energy level). The CH stores the information of the nodes of its cluster in a table. Note that some nodes may follow two CHs if they fall on the cluster border; these are known as Gateway nodes (GW). GWs are used to communicate between different CHs, and to disseminate the information of a faulty node to the whole network. In the third stage, a fault is detected if a common node does not send anything to its CH for two successive periods. Moreover, each CH calculates a median value for each node from the corresponding collected information at the installation phase. The median value is recalculated each time the CH receives data. The two median values are compared: if the difference exceeds a predefined

ISBN: 978-0-9853483-6-6

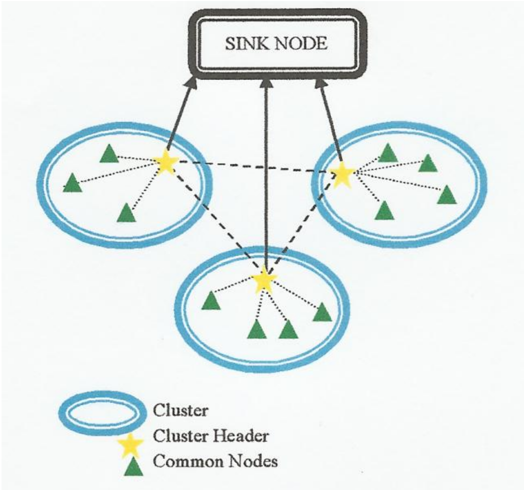

Figure 2. Distributed Approach

threshold, the node is considered faulty. The median value method is useful to identify external errors such as noise and interferences. In order to detect a faulty header, CHs periodically exchange messages carrying information of the relevant nodes in the cluster. In case there is no response from a CH, the latter is considered faulty. Then, the node with the highest energy level in the cluster is assigned as the new CH. The drawback of this algorithm lies in the calculations performed by the CH with each received message. In addition, the periodic messages sent from the cluster headers to the nodes negatively affect the CH energy level.

Other techniques rely on neighborhood notification. For example, in Venkataraman algorithm [13], when a node reaches a certain energy level, it sends messages to its parent and children notifying them that it is going to die. A failure recovery technique takes place by connecting the parent to the children to avoid holes in the network.

A cellular approach is used in [14] for fault detection and recovery. The algorithm virtually divides the network into cells. The node with the highest energy level is selected as a cell manager. In addition, a secondary cell manager is selected that acts as a backup in case of the failure of the primary cell manager. Both managers have information about all nodes in the cell. A faulty node is identified by either sending regular messages to the managers, or by notifying the neighbors before its energy is completely depleted.

The advantages of [14] over [12] lie in two points: less network traffic towards the cell manager, and better energy consumption of the nodes.

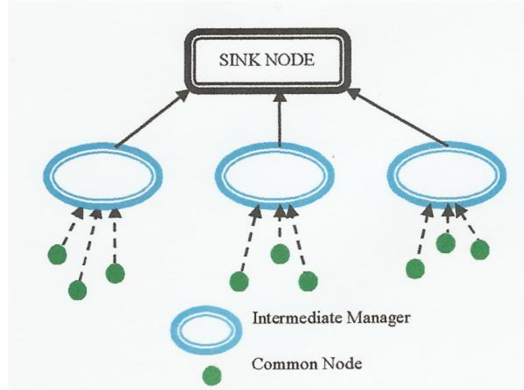

Figure 3. Hierarchical Approach

IV. HIERARCHICAL SCHEME

The hierarchical model divides the sensors into layers of nodes. Each layer is controlled by the upper layer. A cluster header manages a group of nodes in the lower layer. However, there is no communication between the nodes of the same layer. The hierarchical concept can be modeled as a tree in which the root node is the sink node, the lower layer consists of the cluster headers, and the children are the common nodes. Refer to Fig. 3.

The hierarchical model is scalable, and therefore suitable for large networks. It is also energy-efficient as compared to the distributed scheme. There is less possibility of bottleneck occurrence as compared to the central framework. The main drawback of the hierarchical model is the vulnerability of the layer of cluster headers.

The hierarchical model is first presented in [15]. Although it was rarely used in the past few years, but it had received increased attention recently.

V. SUMMARY

In this paper, we discussed the three models of fault tolerance applied in wireless sensor networks. Table 1 introduces a brief comparison between them.

TABLE I. COMPARING THE THREE MODELS

	Central	**Distributed**	**Hierarchical**
Architecture	All nodes send to a central manager	The network is partitioned into virtual clusters, with a cluster header for each cluster	The network is partitioned into three layers: sink node, managers, and common nodes
Partition name	Not applicable	Clusters or cells	Layers
Fault	Central	Cluster head	Subnet

detected by	manager	or cell manager	manager
Managers communica-tion	Not applicable	Cluster heads exchange messages	Not applicable
Network partition problem	Least probable	Highly probable	Most probable
Bottlenecks	Highest	Low	Least
Energy-Efficient	Least	High	Highest
Scalability	Least	Highly scalable	Highest
Populariy	Least	Most popular	Oldest, but is regaining popularity

VI. CONCLUSION

This paper exposes the three main categories of fault management models used in wireless sensor networks. These modes are the central, the distributed and the hierarchical approaches. The central architecture is simple; however, it is subject to bottlenecks and consumes energy. The cellular approach is more energy-efficient. However, it may cause network holes or partitions. The hierarchical model is gaining attention in recent years. It is the most efficient approach in terms of energy. However, the intermediate layer of cluster headers is fragile. The main reason is the lack of communication between the cluster headers. In case of the failure of a cluster header, all its subnet is disjoined from the network.

ACKNOWLEDGMENT

We acknowledge King Saud University for their encouragement and support to conduct this research.

REFERENCES

[1] H. Huangshui and Q. Guihe, "Fault management frameworks in wireless sensor networks". Fourth International Conference on Intelligent Computation Technology and Automation, 2011.

[2] P. Yu, S. Jia and P. Xi-yuan, "A self detection technique in fault management in WSN". Instrumentation and Measurement Technology Conference (I2MTC), 2011 IEEE.

[3] T. Wang and C. Yu, "Collaborative event region detection in Wireless Sensor Networks Using Markov random fields". Second International Symposium on Wireless Communication Systems, 2005.

[4] J. Shell, S. Coupland and E. Goodyer, "Fuzzy data fusion for fault detection in wireless sensor networks". Workshop on Computational Intelligence, 2010.

[5] Z. Guan, Z. Gao, Y. Yang, Y. Li and X. Qiu, "A distributed fault recovery method in clustering-based wireless sensor networks". International Conference on Advanced Intelligence and Awareness Internet (AIAI), 2010.

[6] M. Asim, H. Mokhtar and M. Merabti, "A fault management architecture for wireless sensor network". International Wireless Communications and Mobile Computing Conference (IWCMC), 2008.

[7] P. Mehta et al. "Distributed detection for landslide prediction using wireless sensor networks". Global Information Infrastructure Symposium (GIIS), 2007.

[8] L.B. Ruiz, J.M. Nogueira, and A.A.F. Loureiro. "MANNA: a management architecture for wireless sensor networks". IEEE Communications Magazine, Vol. 41, No. 2, pp. 116-125, Feb. 2003.

[9] W.L. Lee, A. Datta and R.C.-O. "FlexiMAC: a flexible TDMA-based MAC protocol for fault-tolerant and energy-efficient wireless sensor networks". 14th IEEE International Conference on Networks (ICON), 2006.

[10] N. Ramanathan, E. Kohler, L. Girod and D. Estrin. "Sympathy: a debugging system for sensor networks". 29th IEEE International Conference on Local Area Networks, 2004.

[11] Y. Liu, K. Liu, and M. Li. "Passive diagnosis for wireless sensor networks". IEEE/ACM Transactions on Networking, Vol. 18, No. 4, August 2010.

[12] M. Afsar, M. Yaghmaee and E. Zeinali. "A fault tolerant protocol for wireless sensor networks". Seventh International Conference on Mobile Ad-hoc and Sensor Networks, 2011.

[13] G. Venkataraman, S. Emmanuel, and S. Thambipillai. "Energy-efficient cluster-based scheme for failure management in sensor networks". IET Communications, Vol. 2, Issue 4, pp. 528-537, April 2008.

[14] M. Asim, H. Mokhtar, and M. Merabti. "A cellular approach to fault detection and recovery in wireless sensor networks". Third International Conference on Sensor Technologies and Applications (SENSORCOMM), 2009.

[15] M. Alam, M. Mamun-Or-Rashid, C. Hong. "WSNMP: a network management protocol for wireless sensor networks". 10th International Conference on Advanced Communication Technology, Vol. 1, pp. 742-747, Feb. 2008.