A Hierarchical-Cellular Fault Management Scheme for Ad Hoc Wireless Sensor Networks

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

***Abstract*—Ad Hoc Wireless sensor networks are used in many domestic and critical applications including military, medical, pervasive and ubiquitous computing. Maintenance of wireless sensor networks is of primary concern. Therefore, fault management techniques are devised to avoid network partitioning. This paper proposes a new scheme for fault management in wireless sensor networks. The new technique is hierarchical-based; however, it applies the cellular concept as well. The proposed scheme is energy-efficient, accurate, highly scalable, dynamic and avoids bottlenecks in the network. The algorithm combines the advantages of the hierarchical and cellular schemes**

Keywords—Fault Management; Wireless Sensor Networks; Hierarchical Scheme; Cellular Scheme

I. INTRODUCTION

Recently, Ad Hoc Wireless Sensor Networks (WSN) became of vital importance. They are used in many areas such as military, medical, as well as in harsh environments. Moreover, they are used in home and personal security. Sensors are limited in their computation power, memory capacity and energy level. Therefore, maintaining network connectivity despite of the failure of one or more nodes is of a primary concern.

Fault management techniques include fault detection, identification, recovery and prediction. By fault detection, it is meant that the network is aware of the existence of a faulty node. Fault identification specifies the faulty node. The error is then overcome by a fault recovery algorithm. At the highest level, a fault may be predicted before its occurrence. Special measures take place to avoid the disconnection or the partitioning of the network.

Efficient fault management techniques should meet the following minimum criteria:

- Energy efficient: since the sensors are limited in power, then the algorithm devised to support the network connectivity should cause the least possible overhead. Computations consume from the sensor's energy level.
- Accurate: the devised algorithm should take into consideration all possible types of errors that may occur in the network. In fact, there is a trade-off between accuracy and energy efficiency.
- Dynamic: the algorithm should respond quickly to the changes in the network. New sensor nodes may join the network, and others may vanish because of energy depletion.

This paper proposes a new scheme for fault management in wireless sensor networks. The paper is divided as follows: Section II emits light on relevant background. Section III exposes some related work in the literature. Section IV proposes the new scheme. The paper is concluded in Section V.

II. BACKGROUND

There are many fault management techniques mentioned in the literature. However, they can be classified into three main categories [1]:

- Central scheme,
- Distributed scheme,
- Hierarchical scheme.

In the central scheme all sensor nodes exchange messages with the sink node. This imposes network bottleneck near the central node. In addition, it consumes a lot of the energy of the sink node. This leads to the quick failure of the network. Figure 1 shows the distribution of the sensors in a central scheme.

On the other hand, the distributed scheme resolves the problem of the energy depletion by virtually dividing the nodes into clusters. Other than the sink node, there are two types of nodes in such architecture: the Cluster Header (CH) and the common nodes. Each cluster is managed by a cluster header. At the installation phase, the CH and the common

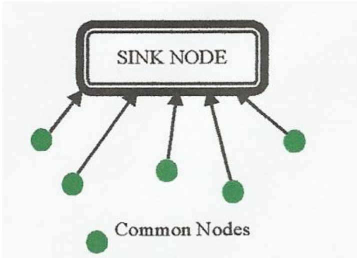

Figure 1. Central Approach

nodes in its cluster exchange short messages. The CH stores the information of all its common nodes in a table. This information usually includes the ID of the common node, its location and its energy level. Figure 2 depicts such architecture.

Faults may be detected in two different ways in such architecture. In the first technique, a node is considered faulty if it does not send any data to its cluster header in a specific period of time. The second technique relies on predefining a threshold to the energy level of the nodes. Then, if a node goes below this threshold, it notifies its neighbors that it is going to die. The cluster header is then notified. Cluster headers are inter-communicated to detect faulty ones.

The hierarchical scheme is a tree-like architecture consisting of three layers. The upper layer consists of a single node which is the sink node. The intermediate layer consists of all cluster headers. There is no communication between cluster headers. The lower layer consists of the common nodes. Figure 3 illustrates the hierarchical architecture.

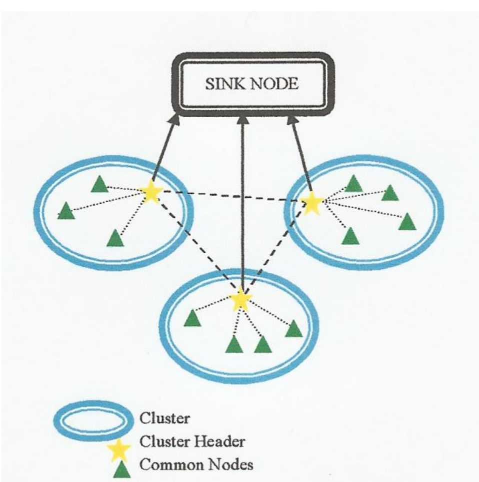

Figure 2. Distributed Approach

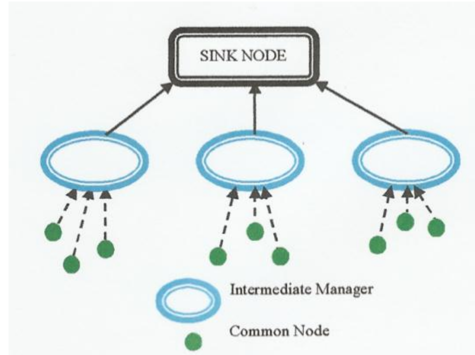

Figure 3. Hierarchical Approach

The central scheme is considered the least efficient architecture for many reasons: fast energy depletion, network inability to scale, highly probable bottleneck near the sink node, and the slow response to the network dynamics. The distributed scheme is more efficient as compared to the central scheme. The design saves nodes' energy. In addition, the network traffic is decreased near the sink node, since the network management is distributed amongst cluster headers. The organization of the network allows a relatively fast response to the addition or deletion of nodes. The main problem of such scheme lies in the case of failure of a cluster header: this leads to network partition. Finally, the hierarchical scheme is more energy efficient as compared to the distributed scheme. In addition, it is more scalable and dynamic. The main problem in such scheme is the lack of communication between the cluster headers lying in the intermediate layer of the architecture. In addition, there is no technique – to the extent of our knowledge – that checks for the energy status of the cluster headers. In this paper, a scheme is proposed to resolve such problem.

III. RELATED WORK

MANNA [2] is a fault detection and identification based on the central approach. It provides flexibility by defining three architectures: functional, information and physical. The coordination between these architectures is depends solely on policy-based management.

Sympathy [3] is a debugging tool for collecting system performance metrics with minimal overhead; recognizing events based on these metrics and collecting events based on their spatio-temporal context.

FlexiMAC [4] implements a synchronized and flexible slot structure in which the nodes can build, modify or extend their scheduled number of slots during execution based on their local information; thus fulfilling the energy and memory constraints of the sensors.

PAD [5] is a probabilistic diagnosis approach for inferring the causes of abnormal phenomena. It employs a packet marking scheme for the construction and maintenance of the inference model.

A distributed approach is proposed in [6]. The algorithm divides the network into virtual clusters. It distinguishes between three types of nodes: cluster headers (CHs), Gateway nodes (GW) and common nodes. At the installation phase, nodes with energy level above a predefined threshold are nominated to be CHs. Candidate cluster headers then exchange short messages with common nodes. A common node responds to the closest CH. By this way clusters are formed. It may happen that a node belongs to more than a cluster: these are GWs. They are used to communicate between cluster headers and to disseminate information about faulty nodes to the whole network. A node is considered faulty if it does not send any data to its cluster header for two successive periods. A non-responsive CH is considered dead.

Venkataraman devised a cluster-based algorithm for fault tolerance in WSN [7]. The technique relies on the neighborhood notification concept in which a node notifies its neighbors that it is going to die. A fault recovery mechanism therefore takes place by attaching the parent of the dying node to its children. By this way, network holes are avoided.

A fault management technique based on the cellular approach is devised in [8]. The algorithm divides the network into a virtual grid. For each cell, the node with the highest level energy is assigned as a primary manager. The latter selects the node with the next highest energy level to be a secondary manager. The secondary manager plays the role of backup for the primary manager in case of its failure. The fault management technique is also based on the neighborhood notification. Cluster managers communicate with each other to identify a faulty cluster manager.

The hierarchical approach is presented in [15]. The algorithm maintains the network by monitoring its status, collecting management data, and taking the necessary action accordingly.

IV. PROPOSED SCHEME

The proposed scheme relies on the hierarchical and cellular architectures previously explained in Section II with slight modifications. The main problem of the hierarchical scheme lies in the intermediate layer. The failure of a cluster header leads to the isolation of all its subnet. Moreover, there is no communication between the CHs to check their statuses.

Therefore, the suggested idea is to construct a hierarchy of clusters. There are still three layers including the sink node in the upper layer. The intermediate layer consists of a single cell of cluster headers; each CH is responsible for a cell of common nodes in the lower layer. No common node is allowed to follow more than a single CH. Finally, the third layer consists of a number of clusters – equal to the number of cluster headers – containing the common nodes. This is illustrated in Figure 4.

At the installation phase, common nodes send their information to their corresponding cluster headers. In turn, CHs store this information in a permanent table. The information includes the node ID, location and energy level. During operation, updated data are immediately reflected in the table. A common node is considered faulty if its energy level goes below a predefined threshold. In this case, the dying node broadcasts a message notifying its peers and CH that it is going to disappear. In return, the CH deletes the corresponding information from its table.

If the energy of a certain cluster header is going to vanish, the latter selects from the table of common nodes the one with the highest energy level. This becomes the new CH. The dying CH transfers to the new one its table and disappears. At the same time, the new CH goes up one level and becomes the parent of the cluster. The table of the sink node containing information about the cluster headers is accordingly updated.

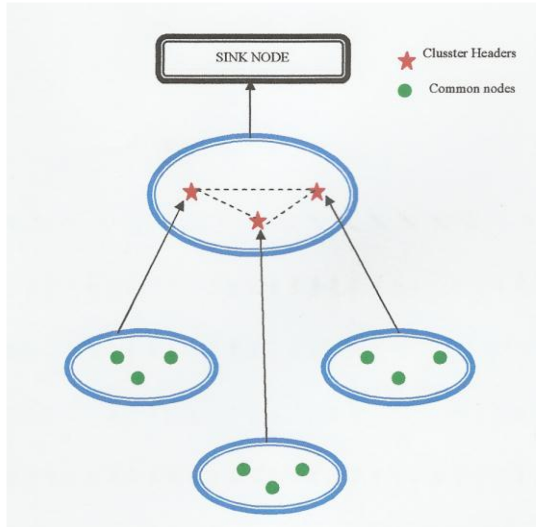

Figure 4. Hierarchical-Cellular Approach

When the number of the nodes in a cluster exceeds a predefined threshold, a cluster division takes place. The node with the highest energy level in the new cluster is then taken as cluster header and goes up one level. The new CH exchanges messages with the nodes in the newly created cluster and stores the information in its table. On the other hand, the information of the moved nodes is deleted from the table stored by the original CH. This action is necessary to decrease the memory capacity requirements of a CH and therefore increase efficiency.

The proposed scheme overcomes the problem of the fragility of the intermediate nodes in the hierarchy. However, it still preserves the advantages of the energy saving, the scalability, the low probability of bottlenecks occurrence and the fast response to network dynamics. In addition, the inter-communication of cluster headers in the intermediate layer gives a global view of the network to each CH. This is required for routing.

V. CONCLUSION AND FUTURE WORK

Wireless sensor networks invaded all aspects of our life nowadays. They are used in both domestic and critical applications. Since sensors are in their nature limited in their power level, an efficient fault management algorithm is therefore a necessity. On the other hand, sensors have small memory capacity that the devised algorithm should be as compact as possible. In all cases, the algorithm should be accurate and fast. Also, it should not involve too much computation that takes off the sensors' energy quickly.

In this paper, a new scheme is proposed. The main target is to keep the network scalable, dynamic, and energy-efficient. In addition, its organized scheme does not allow for the occurrence of bottlenecks. Moreover, each cluster head has a global view of the whole network.

Fault management techniques include fault detection, identification, recovery and prediction. The proposed algorithm predicts the near death of a sensor by predefining a threshold for energy level. Then, the fault is recovered by maintaining the communication between the sensor nodes. A faulty cluster header is also taken into consideration. In addition, the algorithm maintains the network efficient by avoiding bottlenecks in any cluster: if the number of sensor nodes exceeds a certain level, the cluster is divided into two.

Simulations of the proposed scheme are to be performed using Matlab. Results are to be published in the literature.

ACKNOWLEDGMENT

My acknowledgment goes sincerely to King Saud University (KSU) that supported me to put my efforts in 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] 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.

[3] N. Ramanathan, E. Kohler, L. Girod and D. Estrin. "Sympathy for the sensor network debugger". 29^{th} IEEE International Conference on Local Area Networks, 2004.

[4] 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". 14^{th} IEEE International Conference on Networks (ICON), 2006.

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

[6] 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.

[7] 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.

[8] 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.

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