# Fault Management in Wireless Sensor Networks

[1,]S.Irfan , [2,]K.Sravani, [3,]K.Manohar
[1, 3] *Lecturer, Electrical Engineering Department, Wollo University, Ethiopia*
[2] *Assistant Professor, Swarna Bharathi College of Engineering, Khammam, A.P.*

***Abstract***

*Wireless sensor networks (WSNs) are resource-constrained self-organizing systems that are often deployed in inaccessible and inhospitable environments in order to collect data about some outside world phenomenon. For most sensor network applications, point to- point reliability is not the main objective; instead, reliable event-of-interest delivery to the server needs to be guaranteed (possibly with a certain probability). The nature of communication in sensor networks is unpredictable and failure-prone, even more so than in regular wireless ad hoc networks. Therefore, it is essential to provide fault tolerant techniques for distributed sensor applications. In this work we propose and evaluate a failure detection scheme using management architecture for WSNs, called MANNA. We take a deep look at its fault management capabilities supposing the existence of an event-driven WSN. This is a challenging and attractive kind of WSN and we show how the use of automatic management services defined by MANNA can provide self- configuration, self-diagnostic, and self-healing (some of the self-managing capabilities). We also show that the management solution promote the resources productivity without incurring a high cost to the network.*

***Keywords :****Wireless Sensor Networks, Fault Management, Self-management, Network Monitoring.*

## I. INTRODUCTION

Continuing advances in computational power and radio components, as well as reduction in the cost of processing and memory elements have led to the proliferation of micro-sensor nodes that integrate computation, communication, and sensing capabilities into a single device. Wireless sensor networks are self-organized networks that typically consist of a large number of such sensing devices with severely limited processing, storage and communication capabilities and finite energy supply. Sensor networks are now rapidly permeating a variety of applications domains such as avionics, environmental monitoring, structural sensing, tele-medicine, space exploration, and command and control. With multi hop wireless communication, sensor nodes have made it possible to build reactive systems that have the ability to monitor and react to physical events/phenomena. In addition to resource constraints, sensor networks are also failure-prone. Therefore, fault tolerance is as critical as other performance metrics such as energy efficiency, latency and accuracy in supporting distributed sensor applications. Due to the small dimensions, sensor nodes have strong hardware and software restrictions in terms of processing power, memory capability, power supply, and communication throughput.

The power supply is the most critical restriction, given that it is typically not rechargeable. For this reason faults are likely to occur frequently and will not be isolated events. Besides, large-scale deployment of cheap individual nodes means that node failures from fabrication defects will not be uncommon. Attacks by adversaries could happen because these networks will be often embedded in critical applications. Worse, attacks could be facilitated because these networks will be often deployed in open spaces or enemy territories, where adversaries can not only manipulate the environment (so as to disrupt communication, for example, by jamming), but have also physical access to the nodes. At the same time, ad-hoc wireless communication by radio frequencies means that adversaries can easily put themselves in the network and disrupt infrastructure functions (such as routing) that are performed by the individual nodes themselves. Finally, the sensors nodes are susceptible to natural phenomenons like rain, fire, or even falls of trees since they are commonly used to monitor external environments. For all these reasons faults in WSNs need to be tackled differently than in traditional networks. Fault management, an essential component of any network management system, will play an equal, if not more, crucial role in WSNs. Failure detection, in particular, is vital not only for fault management, but also for security and performance. If, in addition to detecting a failure, the management application can also determine (or gather indicatives) that it has malicious origin, then the management application can trigger security management services. On the other hand, if it has not a malicious origin, i.e., if it

is an accidental or natural failure, \backup nodes" could be activated in order to substitute the unavailable nodes, hence maintaining quality of service. Given the motivation for applying fault management in WSNs, in this paper we propose a failure detection scheme using MANNA - a network management architecture for WSNs proposed in [8]. Since WSNs have different characteristics and restrictions when compared to traditional networks, the adoption of a management solution which takes this into account is essential. In this work, we focus on event-driven WSNs. An event driven WSN is a type of WSN network which reports data to the observer only when certain event occur (as opposed to continuous networks which reports data at regular intervals). To the best of our knowledge, there has not been much research on failure detection in WSNs and even though proposals do exist, their focus is on continuous networks. Event driven networks pose special challenges to the problem (we discuss them in detail in Section 2).

In order to evaluate the performance the management solution, we analyze the impact of management functions over the network and also its effectiveness in detecting failures. In particular, we analyze the fault management aspect for this kind of network supposing different scenarios. As a case study, we define a simple event-driven application that runs in the WSN for monitoring the environment temperature. We show that our solution achieves a reasonable detection rate, and that it incurs an overhead that is acceptable for mission critical applications. The rest of this paper is organized as follows. In Section 2 we discuss the problem of failure detection in WSNs, briey describes the MANNA management architecture for WSNs, and propose a failure detection scheme for event driven WSNs using this architecture. In Section 3 we describe the management services and functions defined in MANNA and used in management application considered. The simulation model used in our experiments is described in Section 4 and the experimental results in Section 5. Finally, our concluding remarks are presented in Section 6.

## II. FAILURE DETECTION IN WIRELESS SENSOR NETWORKS

WSNs are embedded in applications to monitor the environment and sometimes, act upon it. In applications where we are interested in the conditions of the environment at all times, sensor nodes will be programmed to sense and send back their measurements at regular intervals or continuously. We call these networks programmed and continuous, respectively. In other applications (probably a large class of them), we are only interested in hearing from the network when certain events occur. We call these networks event-driven networks. On the other hand, when the network is able to answer to queries of the observers, we refer to this network as on demand. Configuring the network as event-driven is an attractive option for a large class of applications since it typically sends far fewer messages. This is translated into a significant energy saving, since message transmissions are much more energy-intensive when compared to sensing and (CPU) processing. For instance, if the application is temperature monitoring, it could be possible just to report data when the temperature of the area being monitored goes above or below certain thresholds. In terms of failure detection, event-driven networks present challenges not found in continuous networks. Under normal conditions, the observer of a continuous network receives sensing data at regular intervals. This stream of data not only delivers the content we are interested in, but also works as an indicative of the network operation quality.

If data are received from every single node, then it knows that all is well (of course, assuming that the messages are authenticated, and cannot be spoofed). If, however, the management application stops receiving data from certain nodes or entire regions of the network, it cannot distinguish if a failure has occurred or if no application event has occurred. Leveraging precisely on this indication, and supposing that nodes periodically send messages to the base station, Staddon et al. [5] proposed a scheme for tracing failed nodes in continuous sensor networks. Their scheme takes advantages of periodic transmission of sensor reports to do the tracing. Because we consider event-driven networks, their solution is not directly applicable. In [4], it is proposed a scheme where nodes police each other in order to detect faults and misbehaviour. More specifically, nodes listen-in on the neighbour it is currently routing to, and can determine whether the message it sent was for-warded. If the message was not forwarded, the node concludes that its neighbour has failed and chooses a new neighbour to route to. Unfortunately, this scheme does not help in cases in which an entire region is compromised.In our work, we study the problem of failure detection for an event-driven WSN and propose a fault management solution using some management services, management functions, and WSN models which are part of the MANNA architecture [8]. In MANNA management services for WSNs are defined. These management services are performed by a set of functions which take executing conditions from the WSN models. The WSN models, as defined in the MANNA architecture, represent the states of the network and serve as a reference for the management. The definition of the management services and functions is based on three management dimensions, namely management functional areas, management levels, and WSN functionalities.

In the following, we discuss how the MANNA architecture can cope with this kind of network promoting its self- managing. We also describe the management application defined for providing fault management.

## III. FAULT MANAGEMENT APPLICATION USING MANNA

In order to evaluate the fault management capabilities of the management solution proposed, we have simulated an application that monitors an environment to collect temperature data. As said before, we have considered an event-driven WSN. We suppose this network as being heterogeneous and hierarchical. The sensor nodes only disseminate data when the temperature of the area being monitored goes above or below certain thresholds. In a hierarchical network, nodes are grouped into clusters and there is a special node called cluster-head. In a heterogeneous network, the cluster-heads have more resources and, thus, are more powerful than the common-nodes. Furthermore, they are responsible for sending data to a base station (BS). The BS also communicates with the observer, which is a network entity or a final user that wants to have information about data collected from the sensor nodes. In our implementation, the management agents execute in the cluster-heads where aggregation of management and application data is performed. This mechanism decreases the information flow and energy consumption as well. A manager is located externally to the WSN where it has a global vision of the network and can perform complex tasks that would not be possible inside the network. In this work we use automatic management services and functions, i.e., executed by management entities (manager or agent) invoked as a result of information acquired from a WSN model. The computational cost of some autonomic process (automatic management services) could be expensive to the architecture proposed. The external manager then extends the computation capabilities avoiding the consumption of network energy to carry out this task. Locations for managers and agents, and the functions they can execute are suggested by the functional architecture. MANNA architecture also proposes two other architectures: physical and information. More details can be found in [8].

The management application is divided into two phases: installation and operation. The installation phase occurs as soon as the nodes are deployed in the network. In this phase, each node finds out its position in the area and reports it to the agent located in the cluster-head. The agent aggregates the information received from the nodes in the group and sends a LOCATION TRAP of its localization to the manager. The common-nodes also inform their energy level that the agent aggregates in an ENERGY TRAP sent to the manager. The management application builds all needed WSN models based on both local information and data sent by the agents, i.e., the WSN topology map model and the WSN energy model. These two models are used to build the WSN coverage area model, which the manager uses to monitor the sensing and communication coverage area. In the operation phase, while the sensor nodes are performing their functions, i.e., collecting and sending temperature data, management activities take place. Among them, energy level monitoring plays a central role. Each node checks its energy level and sends a message to the agent whenever there is a state change. This information is transmitted to the manager via another ENERGY TRAP. Any information the agent receives is recorded in its MIB. The manager can, then, recalculate the energy and topology maps, as well as the coverage area, which characterizes the coverage area maintenance service. Also, operations can be sent to the agents in order to execute the failure detection management service. The manager sends GET operations in order to retrieve the node state. The GET-RESPONSEs are used to build the WSN audit map. If an agent or a node does not answer to a GET operation, the manager consults the energy map to verify if it has residual energy. If so, the manager detects a failure and sends a notification to the observer. In this way, MANNA architecture provides failure detection in event-driven WSN. In the next section we describe the experiments conducted in order to evaluate MANNA's performance as a solution for failure detection.

## IV. SIMULATION MODEL

For our study, we have conducted a set of experiments taking into account distinct simulation scenarios. We have defined a WSN application and some management functions, as mentioned before, and evaluated the performance of the system using the Network Simulator (ns-2) [6], version 2.1b8a. Each scenario was simulated 33 times. In our application, the temperature is the monitoring object. Although the nodes sense the temperature continuously along the time, data are sent only when the minimum or the maximum value collected differs 2% from the last data sent, inducing the event-driven property to the sensing application. In order to simulate the temperature behaviour of the environment, random numbers were generated following a normal distribution, taking into consideration standard deviation of one from an average temperature of 25°C. Figure 1 presents nodes distribution in the monitored area. Table 1 describes the network parameters and the features of the nodes. We use UDP, IEEE 802.11, and single hop communication between cluster-heads and base station. Between the base station and cluster-heads we use SNMP for the application layer but between common-node and cluster head we use a new light-weight protocol, MNMP (MANNA Network Management

Protocol) which we designed [7]. Table 2 presents the parameters of the management application which was simulated.

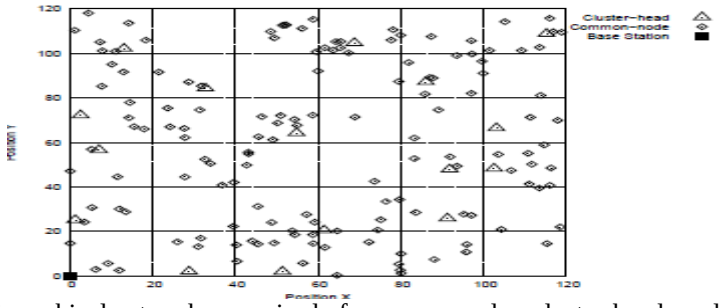



Figure 1: Hierarchical network comprised of common-nodes, cluster-heads and a base-station.

## V. EXPERIMENTAL RESULTS

In order to evaluate the results, we have considered two sets of experiments with two distinct goals. The first set aimed at evaluating the impact of management functions over the WSN, analyzing the management cost. The second one was meant to identify the effectiveness of the management architecture in detecting failures. For both sets we have simulated an unexpected event to happen at the middle of the simulation time. This event puts the nodes, confined in a predefined region, out of operation. We could think of this event as a car passing over the network, the fall of a tree, a spot of fire, or another external event which could ruin the nodes.

### 5.1 Evaluating Management Impact

In this section we evaluate the impact of management functions over the WSN, analyzing its costs. Table 3 shows the three scenarios considered in the first set of experiments in respect to management functions. For this set of experiments, we have considered the unexpected event to cause the failure of 32 nodes located at the centre of the network (which have x and y coordinates between 30 and 90). This event happens at 45 s of simulation. Three metrics were chosen in order to analyze the results. The first one was the delivery rate, which measures the ratio of messages received by the nodes in the network to messages sent by the nodes, during the simulation time. This metric computes the ability of the network to deliver messages at their destinations. The second metric chosen was the average energy consumption, which measures the ratio of total dissipated energy by the nodes to the num-ber of nodes in the network. This metric defines the cost of transmitting and receiving packets per node and sensing. The third metric chosen was the number of messages transmitted. This metric shows the traffic imposed by the nodes tasks.

**Table 1: Simulation parameters.**

| Parameter | Value |
|---|---|
| No. of nodes | 160 (144 common-nodes and 16 cluster-heads) |
| Cluster size | Variable |
| Simulation time | 100 s |
| Coverage area | 120 m × 120 m |
| Environment conditions | Variations in the environment and noise are not considered |
| Initial energy available in each node | 5 Joules |
| Network type | Heterogeneous |
| MAC Protocol | IEEE 802.11 |
| Routing Algo-rithm | None |
| Propagation Model | Shadowing |
| Node distribu-tion | Uniform random |
| Transmission power of common-nodes | 9.9 mW (90% of receiving rate at a distance of 15m) |
| Transmission power of cluster-heads | 281.8 mW (90% of receiving rate at a distance of 80 m) |
| Sensing range | 2 m |
| Node capacity | 5 buffers for receiving packets |
| Energy spent in communica-tion | 0.66 W for node transmission, 6.0 W for cluster-head transmission, and 0.2 W for re-ception |
| Energy spent in sensing | 10mW |
| Energy spent in processing | Not considered |
| Node mobility | Stationary |

**Table 3: Description of the Simulated Scenarios - First Set.**

| Scenario | Description |
|---|---|
| 1 | WSN with all the management functions |
| 2 | WSN with management, but with failure detection function removed |
| 3 | WSN without management |

**Table 2: Management parameters.**

| Parameter | Value |
|---|---|
| Agent location | cluster-heads |
| Message Size | 64 bytes |
| Management operations simulated | OPERATIONAL STATUS GET, OPERATIONAL STATUS GET-RESPONSE, POSITION TRAP, ENERGY TRAP and DELETE TRAP |
| Energy level considered to be critical | 1 Joule |
| No. of detection schedulings | 3 during the simulation time |

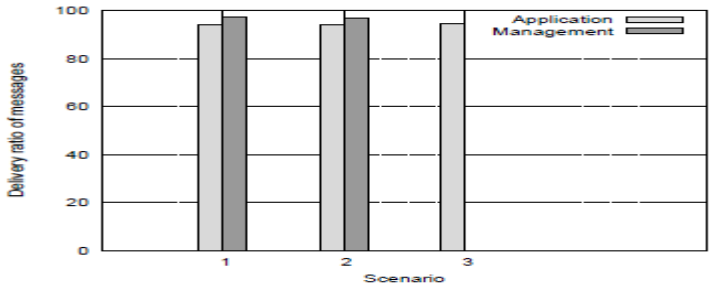



**Figure 2: Delivery rate of messages in the WSN.**

Figure 2 shows the delivery rate for sensing application and management messages. It is observed that for scenarios 1 and 2 the delivery rate for management messages and application messages were similar. This is expected since they are transmitted in the same wireless environment and to and from the same nodes. We can also notice that the introduction of detection (see results for scenario 2) had no influence on this metric. Another result exhibited in Figure 2 concerns the delivery rate of application messages. The introduction of management had little impact on the sensing application.

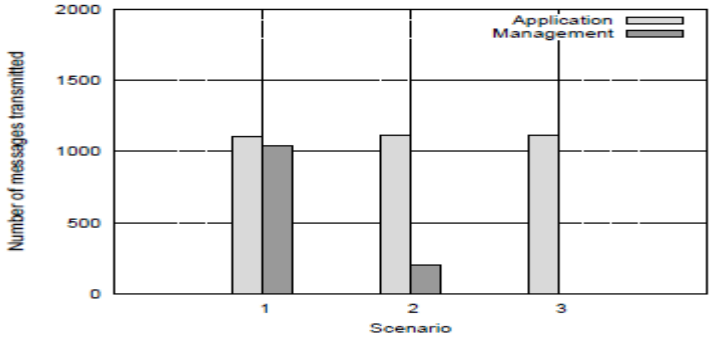



**Figure 3: Number of messages transmitted by nodes in the WSN.**

Figure 3 shows the traffic of messages in the WSN. Comparing the results for scenarios 2 and 3, we can notice that management contributed with only a small increase (18.49%) in the WSN traffic. This is due to the fact that, like the sensing application, management was implemented as eventdriven. However, the number of messages sent almost dou- bled (increase of 93.33%) when management with detection is concerned. This is an expected result since OPERATIONAL STATUS GETs have to be sent to all nodes in the network and be responded by them. Fortunately, as shown before, this is not a problem since the delivery rate of sensing application messages is not greatly impacted. Figure 4 shows the energy consumption of common-nodes and cluster-heads for scenarios 1, 2, and 3. It is observed that, as far as detection is not concerned, the energy consumption increased with management in 18% for cluster heads and 29.45% for nodes. But when the detection mechanism was taken into account, management caused an increase of 101.2% and 129.45% in the energy consumption for cluster-heads and nodes, respectively.

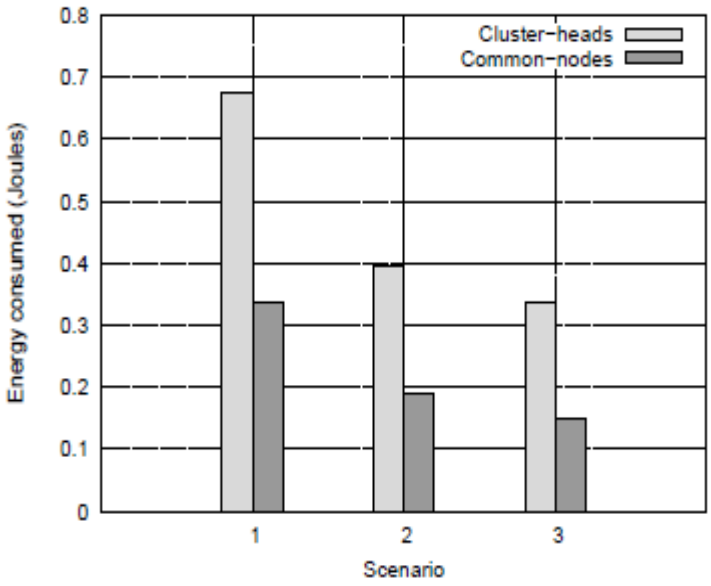
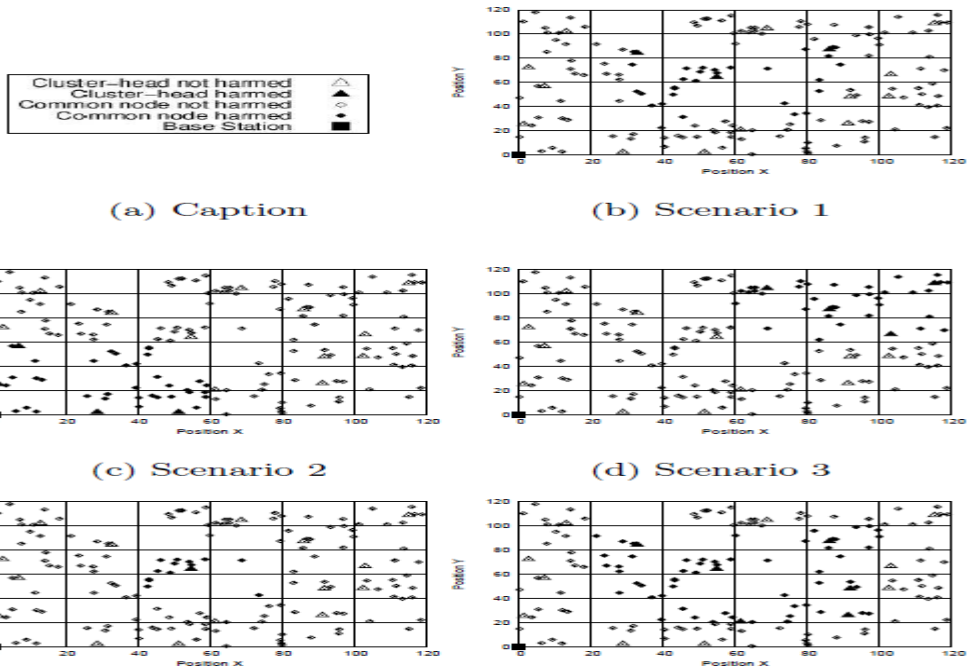



**Figure 4: Energy consumption of nodes in the WSN.**

This result was expected since the act of transmitting and receiving messages are the most determinant activities for energy consumption according to the simulated energy model.

**5.2 Failure Detection Effectiveness**

The results for the first set of experiments gave a motivation for identifying the effectiveness of the detection mechanism provided by the management architecture. The second set of experiments was conducted in order to evaluate if it is worth the increase in traffic and energy consumption. For this second set of experiments, we have tried to modify the region where the ruin of nodes occurred in terms of location and also in terms of dimension. Table 4 presents the description of the simulated scenarios, represented in the Figures 5(b), 5(c), 5(d), 5(e), and 5(f). For space reasons, the captions are omitted in these figures and shown in Figure 5(a). For these experiments, we simulated an event which harms the nodes at 45 s of simulating, putting them out of operation until the end of the simulation. The manager was programmed to start the detection mechanism, i.e., to send the OPERATIONAL STATUS GETs at times 25, 50, and 75 s and to report the results at times 50, 75, and 100 s, respect- tively. So, at the time when the unexpected event occurs, there was time enough for the manager to have come to a conclusion regarding the availability of the nodes. This means that only the reports in 75 and 100 s would have to contain any conclusion regarding this event. Thus, the report at 50 s shows the results obtained before the event occurrence. The results, shown in histograms, present the total number of failures detected by the manager for each scenario, comparing with the number of genuine (forced) failures. The number of failures detected that were not real failures (false positives) and the number of failures not detected are also presented. Just as an illustration, Figure 6 demonstrates the results obtained for one simulation, regarding scenario 1 (caption is shown in Figure 5(a)).




(a) Caption (b) Scenario 1

(c) Scenario 2 (d) Scenario 3

(e) Scenario 4 (f) Scenario 5

**Figure 5: Nodes harmed/not harmed.**

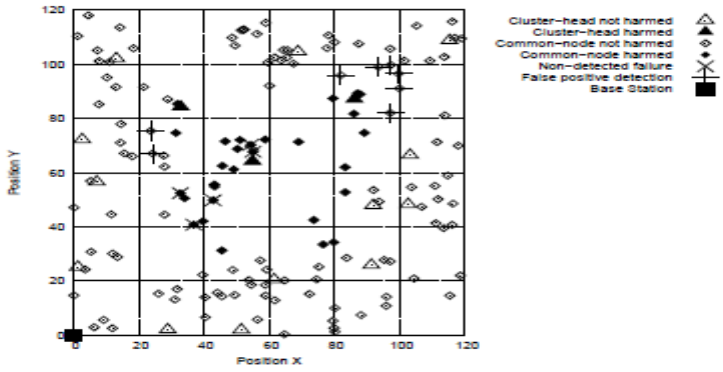



**Figure 6: Result for a case of failure detection.**

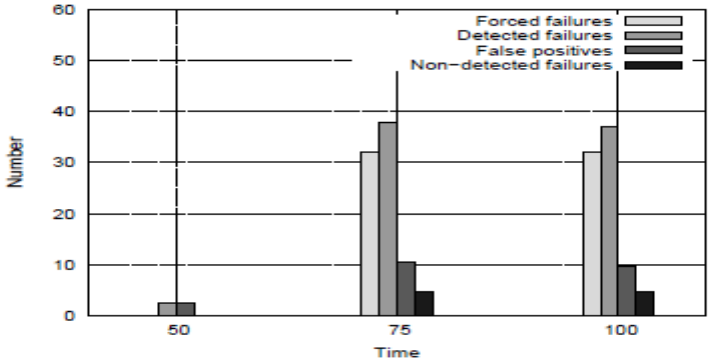



**Figure 7: Detection effectiveness for centered failures (scenario 1).**

Figure 7 shows the effectiveness of the detection mechanism for scenario 1. The numbers in the x axis represent the points in time when the manager reports the availability of the nodes in the network. We can observe in Figure 7 that there was some detection in time 50 s, although at this time the destruction of the nodes could not yet be perceived. Drops of OPERATIONAL STATUS GETs or OPERATIONAL STATUS GETRESPONSEs cause the manager to be misled and, consequently, produce false positives. This problem also occurs at points 75 and 100 for the same reason, representing 27.93% and 26.06% of the detections, respectively. The quantity of false positives at these points is considerably higher than the quantity for point 50 due to the harm of some cluster heads where the agents run. What happens is that after the unexpected event occurs, some common-nodes, which were not harmed, lost their cluster-heads if they are located inside the damaged region. As a consequence, these common-nodes stop receiving the OPERATIONAL STATUS GETs from the manager, since they are sent to them through the agents. As a result, the manager does not receive answers from these common-nodes provoking false positives. In respect to scenario 1, the number of \orphan" nodes were 8. Besides the false positives, the results in Figure 7 also show the amount of non-detected failures, representing 14.81% of the failures in both points 75 and 100. The manager cannot recognize a failure if it does not have knowledge of the damaged node. This may be caused by drops at the initial phase of the network when nodes send their positions to the agents and the agents aggregate the information received in a POSITION TRAP for the manager. Another reason is that the distribution of common-nodes and cluster-heads is random and since the transmission range is limited, there is no guarantee that every common-node will be connected to a cluster-head. Figure 8 shows the results for scenario 2. We can see that the results for point 50 are almost the same as the results for the centered region (scenario 1). As mentioned before, at that point the unexpected event had not yet been perceived, meaning that the results seem to be independent from the region chosen. However, as far as points 75 and 100 are concerned, it is possible to observe considerable dissimilarities. The number of false positives has decreased to 10.26% (point 75) and 10.36% (point 100) of the detections. The reason is that in this experiment the number of orphan nodes are only 4, i.e., two times less than the number of orphan nodes for scenario 1.

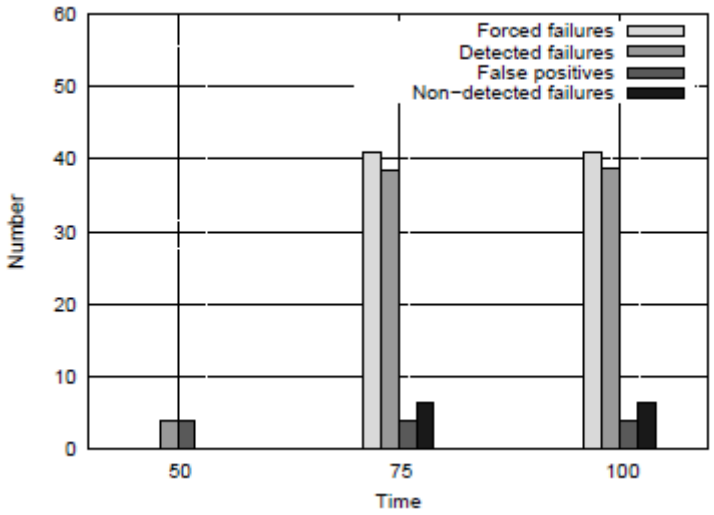



**Figure 8: Detection effectiveness for failures near the BS (scenario 2).**

Figure 8 also shows the results for non-detected failures. Comparing with the results for scenario 1, the amount of non-detections is similar, representing 15.59% of the failures.This shows that the number of initial messages drops in the center is similar to the region near the BS. Figure 9 shows the results for scenario 3. We can notice that the quantity of false positives in point 50 is smaller when compared to the previous results. However, as stated before, this result is independent from the region chosen. Regarding points 75 and 100, a slight decrease in the number of false positives when compared to scenario 1 was produced. In terms of percentage of detections, this quantity represents now 21.48% and 21.67% for points 75 and 100, respectively. The number of orphan nodes for this experiment is 7, very similar to the number produced for scenario 1. Thus, another cause for the decrease exists. Through the logs of the simulations, it is possible to notice that the highest quantity of drops generally takes place in regions far from the BS. This is due to the wireless propagation behaviour. The farthest the source of a message is from the destination, the lower is the probability of this message being delivered. For this reason, most of the false positives are nodes far from the BS (since their

agents are located far too). Therefore, when these nodes are damaged the number of false positives is likely to decrease. Figure 9 also presents the results for non-detected failures. Comparing with the previous results, the amount of non-detections is higher, representing 27.87% of the failures. This clearly shows that the initial message drop is higher at regions far from the BS, as just said. As a result, most of
the nodes, which the manager does not know, are located far from it.

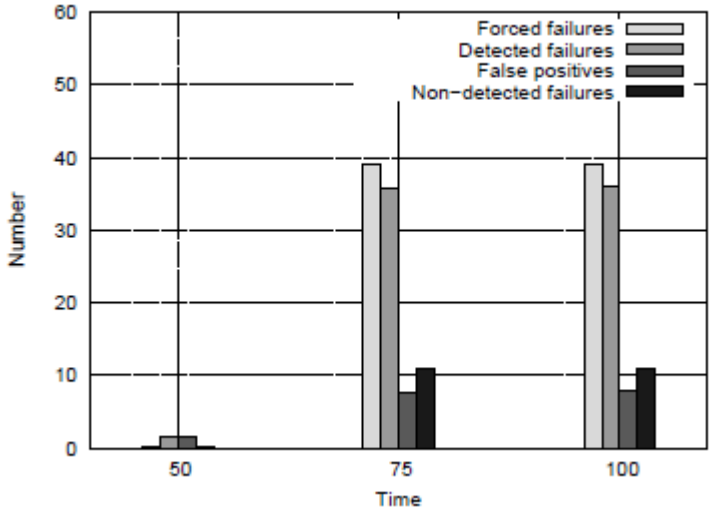



**Figure 9: Detection effectiveness for failures far from the BS (scenario 3).**

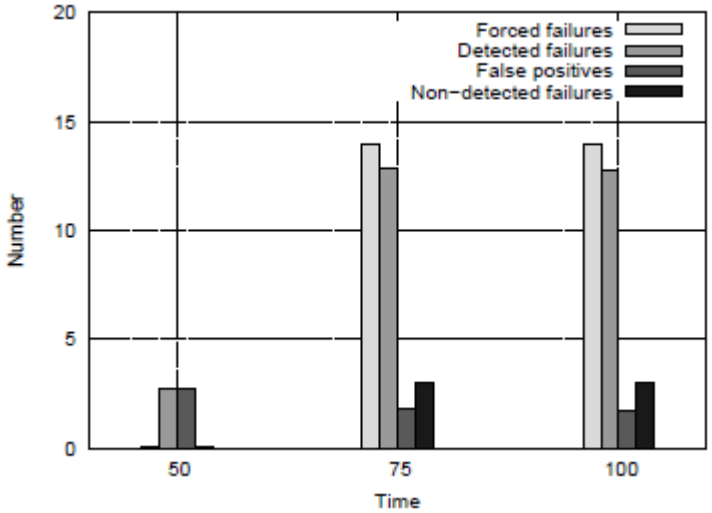



**Figure 10: Detection effectiveness for less failures (scenario 4).**

Figure 10 shows the results for scenario 4. It can be noticed that the results for point 50 are almost the same as the results shown in Figure 7 for scenario 1. As said before, the results are independent from the region chosen. On the other hand, in points 75 and 100 a great difference can be perceived in the number of false positives, which has decreased as expected, since the number of orphan nodes is smaller. The percentage of false positives in relation to detections is now 14.24% and 13.95%. Figure 10 also presents the results for non-detected failures. Comparing to the results of scenario 1, the amount of non-detections is lower. However, in terms of percentage, it represented 27.87% of the failures, i.e., a higher result. This is due to the fact that the region chosen, as seen in Figure 1, almost coincides with one group and if an initial message from this group is lost, the manager lacks the knowledge of the whole group.

Figure 11 shows the results for scenario 5. It can be noticed that the results for point 50 are almost the same as the results shown before. Nevertheless, in points 75 and 100 some differences can be observed. Regarding the false positive results and comparing with the results shown in Figure 7 (scenario 1), the quantity of false positives has increased slightly for the enlarged region. Providing that the number of orphan nodes is a lot higher, a greater increase would be expected. However, since the undamaged region is smaller, there are less drops of OPERATIONAL STATUS GETs and OPERATIONAL STATUS GET- RESPONSEs not due to unavailability. This provokes the number of false positive to reduce. The quantity of false positives shown in Figure 11 is, thus, a result of two opposed factors. As a consequence, as far as the percentage in relation to detections is concerned, there was a decrease (18.45% for both points).

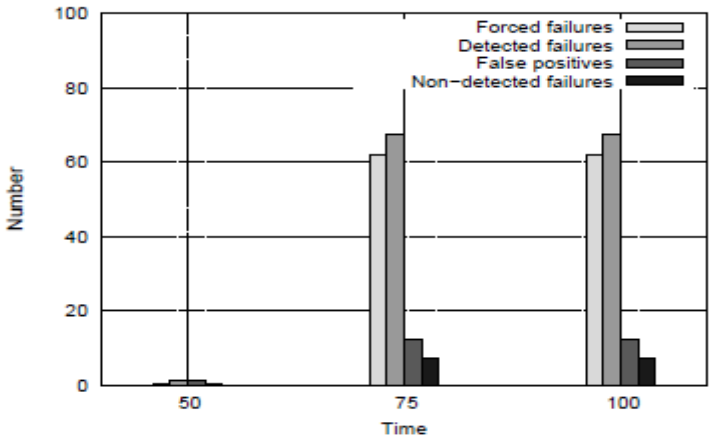



**Figure 11: Detection effectiveness for more failures (scenario 5).**

This shows that the detection mechanism seems to scale well. Figure 11 also presents the results for non-detected failures. In comparison to the results for scenario 1, the amount of non-detections is higher. This was expected since the number of damaged nodes is higher as well as their probability of being unknown to the manager. In terms of percentage, it represented 11.36% of the failures, i.e., a lower result. This shows again a good scalability.

## VI. CONCLUSION

We have conducted two sets of experiments. The results for the first one proved that the introduction of fault management in the WSN was responsible for a great increase in the number of messages transmitted in the network. Although the delivery rate of the sensing application messages was not affected, the energy consumption of the network grew considerably. In spite of that, the second set of experiments shows that the number of nodes harmed and also their location does not influence much the effectiveness of the detection mechanism. Concerning the second set and the results for point 50, we could notice a small fixed number of false positives. At point 75, the percentage of false positive in relation to the detections varies from 10% to 28% whereas the non-detected failures vary from 11% to 28% of the forced failures. Point 100 presented almost the same results, meaning that the timehas worthless influence. The main reasons for these high portions of false positives and non-detections were message drops and the creation of orphan nodes after the occurrenceof the unexpected event. Sensor nodes have to communicate via wireless channels and message drops will exist. The problem could be reduced by sending redundant information or using acknowledge schemes. However, the use of these solutions would result in an increase in the energy consumption, which is undesirable for WSNs. Hence, there is a trade-off between benefits and costs which need to be investigated more thoroughly. The problem of orphan nodes, on the other hand, could be solved by the use of an adoption schema assigning undamaged or redundant cluster-heads to the orphan nodes. The main conclusion we could draw about our approach is that its cost is fixed and its effectiveness is the same, independent from the failures which take place. Although one might think at first sight that the cost introduced by management is high enough to be paid for, this could be worth, and since failures are a common fact in WSNs. Applications which have critical requirements could be a lot beneficed with the knowledge provided by MANNA fault detection. In this paper, we evaluated the MANNA management architecture for WSNs, considering an event-driven WSN. From the experiments presented above, we can see that the solution proposed achieves a reasonable detection rate, and that it incurs an overhead that is acceptable for mission-critical applications.

## REFERENCES


[1] B. R. Badrinath, M. Srivastava, K. Mills, J. Scholtz, and K. Sollins. "Special issue on smart spaces and environments". IEEE Personal Communications, Oct 2000.

[2] D. Estrin, R. Govindan, and J. Heidemann. "Embedding the Internet". Communications of the ACM, 43(5):39{41, May 2000. (Special issue guest editors).

[3] S. Lindsey, C. Raghavendra, and K. Sivalingam. "Data gathering in sensor networks using the energy delay metric" In International Workshop on Parallel and Distributed Computing: Issues in Wireless Networks and Mobile Computing, San Francisco, USA, Apr 2001.

[4] S. Marti, T. J. Giuli, K. Lai, and M. Baker. "Mitigating routing misbehavior in mobile ad hoc networks". In ACM Mobicom, pages 255{265, 2000.

[5] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava. "Coverage problems in wireless ad-hoc sensor networks". In INFOCOM, pages 1380{1387, 2001.

[6] L. B. Ruiz. MANNA: A management architecture for wireless sensor networks, Dec 2003. Ph.D. Thesis. http://www.dcc.ufmg.br/_linnyer.

[7] L. B. Ruiz, J. M. S. Nogueira, and A. A. Loureiro. "MANNA: A management architecture for wireless sensor networks". IEEE Communications Magazine, 41(2):116{125, Feb 2003.

[8] J. Staddon, D. Balfanz, and G. Durfee. "Effcient tracing of failed nodes in sensor networks". In First ACM International Workshop on Wireless Sensor Networks and Applications, Atlanta, GA, USA, Sep 2002.