International Journal of Digital Information and Wireless Communications (IJDIWC) 3(1): 103-111
The Society of Digital Information and Wireless Communications, 2013 (ISSN: 2225-658X)

CMSA: a Centralized Maintenance Strategy with Anticipation for Wireless Sensor Networks

Skander Azzaz and Leila Azouz Saidane
ENSI, Tunis, Tunisia
{skander.azzaz, leila.saidane}@ensi.rnu.tn

ABSTRACT

Providing a continuous service is the main requirement for many application types deployed in Wireless Sensor Networks (WSNs). In this paper, we study two maintenance strategies using a small number of mobile maintainer robots to restore the WSN coverage and connectivity upon a sensor failure: the Centralized Maintenance Strategy (CMS) and the Centralized Maintenance Strategy with Anticipation (CMSA). The CMS and CMSA are based on the Centralized Manager Algorithm [1] used to detect, report sensor failures and coordinate the movement of robots.
In CMSA, the predictive version of CMS, a selected robot is chosen as a manager to anticipate the sensor failures and schedule the available maintainer robots to repair them before they happen. To predict the lifetime of a sensor node, we propose an energy state model that represents the behavior of a sensor node based on Markov Chain. We use this model, validated by simulations, to predict the energy consumption by a sensor node and consequently the lifetime of a wireless sensor node. The simulation results show that the CMSA ensures a null dysfunction network time and a message overhead lower than the classical centralized manager strategy.

KEYWORDS

Energy Model; Fault-Tolerance; Markov Chain; Mobile Robot; Wireless Sensor Networks.

1 INTRODUCTION

Generally, the Quality of Service (QoS) in a wireless sensor network (WSN) is measured by the sensing area coverage degree and the connectivity between the deployed sensors. To protect these WSN QoS parameters, many approaches have been proposed. Exploiting redundancy of nodes [2] is one technique to fill holes appeared in sensor network upon a sensor node failure. This approach requests mobile sensors to maintain the coverage and connectivity in WSNs.
Knowing that a sensor node able to move is expensive, Mei and all in [1] propose using a small number of mobile robots in a static sensor network to replace failed sensors. They have introduced three algorithms: a centralized manager algorithm, a fixed distributed manager algorithm, and a dynamic distributed manager algorithm to detect, report sensor failure and coordinate the movement of robots.
To detect failure, a relation guardian-guardee is established between all sensors. Each sensor node (guardee) selects its guardian (a one-hop neighbor). If a failure occurs, a guardian of the failed node sends a report to its manager robot. In the centralized manager algorithm, all failure reports in the network are addressed to a central manager robot. Once the failure report is received, the manager robot sends the order to handle the failure to the

appropriate robot (a maintainer robot). In the two distributed algorithms, the management responsibility is distributed over the robots, and each robot operates as both a manager and a maintainer. In fact, in the fixed distributed algorithm, the area covered by the sensor nodes is divided into equal-size subareas. Each robot is assigned to a particular subarea as both a manager and a maintainer robot. Using this algorithm, a sensor node selects its manager robot based on its position in the network. If a manager robot receives a failure report, it moves to the failure location and replaces the failed node. In the dynamic distributed algorithm, the manager robot is selected as the closest robot, and the failure is handled by the manager robot that received the report.

In their study, the authors of [1] have shown that the centralized manager algorithm has a low message overhead and the smallest average robot traveling distance per failure among the three algorithms.

In this paper, we present a first strategy, called Centralized Maintenance Strategy (CMS), to repair failures in the WSNs based on the Centralized Manager Algorithm [1]. We detail our improvements on the original algorithm in Section 3.

However, failure handling according to CMS still presents some drawbacks in WSNs. In fact, the connectivity and coverage of the network can be greatly affected if the failure involves a critical node during the repair time of nodes. In this paper, we focus on the improvement of the CMS. Our goal is to provide a fault-tolerant WSN for real-time applications. Our idea is to anticipate the sensor failures. To reach this objective, we need an energy model to predict the energy consumption of sensor node. In the literature, many analytical energy models have been proposed. For example [3] introduce mathematical models to model different MAC (Media Access Control) protocols such as IEEE 802.11 [4], IEEE 802.15.4 (ZigBee) [5] or SMAC (Sensor Media Access Control) [6].

Many problems were present in the deployment of these models. Firstly, these models introduce unrealistic assumptions like Poisson traffic [7] and saturation traffic assumption. Secondly, these models depend on the MAC protocol used by sensor nodes. And finally, for complexity reasons, these models cannot be implemented on the sensor nodes with limited computing capability, memory and energy supplies.

In this paper, we propose a simple analytical energy model dedicated to be implemented in a sensor network. In the next step, we use the analytical model to improve the CMS to provide a fault-tolerant sensor network for real-time applications. The new version of CMS is called Centralized Maintenance Strategy with Anticipation (CMSA).

The remainder of the paper is organized in the following way. In section 2, we present our analytical model to represent the energy consumption of a sensor node. In section 3, we present our repairing strategies CMS and CMSA dedicated to real-time applications supported by WSNs. We validate by simulation our analytical energy model in section 4 and evaluate the performances of our strategies. In section 5, we conclude the paper by giving directions for future work.

2 ANALYTICAL MODEL

The energy dissipated by a sensor node can be divided in two parts: the energy spent for radio communications by the sensor radio interface, denoted in this paper by ε_{Tot}^{Rad}, and the sensing energy (the energy it costs to take measurements) denoted by ε_{Tot}^{Sens}.

We focus first on the evaluation of ε_{Tot}^{Rad}. We claim that the status of the MAC layer of the radio interface of a sensor node can be modeled by a Markov Chain. The states of this Markov Chain depend on the MAC layer used by the sensor node. For example, if we use the IEEE 802.11 MAC layer we have three modes of operation (states): Transmit, Receive and Idle. Each state corresponds to a different power consumption level [8].

- State 1 $Transmit$: node is transmitting a frame with transmission power ε_{Tx}.
- State 2 $Receive$: node is receiving a frame with reception power ε_{Rx}.
- State 3 $Idle$: even when no messages are being transmitted over the medium, the node stays idle and listening the medium with idle power ε_{Id}.

If the sensor node uses IEEE 802.15.4 or SMAC as a MAC protocol, we have a fourth state:

- State 4 $Sleep$: The radio is turned off, and the node is not capable to detect signals. We suppose that the node uses a power ε_{Sl} in this state which is largely smaller than in any other state.

In conclusion, we model any MAC layer of a sensor node as a discrete time Markov Chain with four states: $Transmit$, $Receive$, $Idle$ and $Sleep$.

We note S_i the random variable related to the state of the MAC layer at time i. We choose a time step (unit) such that the duration of any action (e.g; transmission/reception of a frame) is a multiple of this time step and we suppose that all state transitions occur at the beginning of the time step.

The notation $(S_n = i)$ means that the MAC layer is in state i at time step n. Let P_{ij} be the probability that a node in state i will enter in state j at the next transition.

$$P_{ij} = P\{S_{n+1} = j / S_n = i\} \qquad (1)$$

Then, we note by P the transition probability matrix (the element of row i column j represents the probability P_{ij}).

Let $\Pi^{(n)} = [\Pi_1^{(n)}, \Pi_2^{(n)}, \Pi_3^{(n)}, \Pi_4^{(n)}]$ be the probability vector with $\Pi_i^{(n)}, (1 < i < 4)$ represents the probability that the MAC layer is in state i at time step n ($P\{S_n = i\}$).

Then, we have:

$$\Pi^{(n)} = \Pi^{(0)} P^n \qquad (2)$$

And:

$$\Pi_i^{(n)} = \sum_{k \in \{1,2,3,4\}} \Pi_k^{(n-1)} P_{ij} \qquad (3)$$

Knowing that at time step 0 the node is generally in the state 3 ($Idle$), we have:

$$\Pi^{(0)} = [0,0,1,0] \qquad (4)$$

Let $cons_j^{(n)}$ the energy consumed by radio communication at the n^{th} transition given that the node is in state j at time step n:

$$cons_j^{(n)} = \sum_{i \in \{1,2,3,4\}} \Pi_i^{(n-1)} P_{ij} \varepsilon_j \qquad (5)$$

Where:

$\Pi_i^{(n)}$ is obtained from equation 3 and ε_j (the element of column j of the vector E) is the energy consumed to transit from state i to state j.

$$E = \Delta t.\,[\varepsilon_{Tx}, \varepsilon_{Rx}, \varepsilon_{Id}, \varepsilon_{Sl}] \qquad (6)$$

Where: Δt is the length in seconds of a time step.

We note $cons^{(n)}$ the energy consumed for radio communication by a node at the n^{th} transition:

$$cons^{(n)} = \sum_{j\in\{1,2,3,4\}} \Pi_j^{(n)}\, cons_j^{(n)} \qquad (7)$$

Then, $\varepsilon_{Tot}^{Rad^{(n)}}$ that represents the total energy consumed by the radio interface until the time step n is:

$$\varepsilon_{Tot}^{Rad^{(n)}} = \sum_{i\in[1..n]} cons^{(i)} \qquad (8)$$

In addition to radio communication energy, the sensor consumes an amount of energy for event sensing. We note by $cost_{sens}$ the average consumption energy in sensing mode (to take measurements).

$$cost_{sens} = \frac{\lambda}{\mu}\, \varepsilon_{sensing} \qquad (9)$$

Where:

λ: is the average rate of events detected by sensor node, $\frac{1}{\mu}$: is the sensing mean time of an event and $\varepsilon_{sensing}$: is the energy consumed by node in sensing mode (the energy it costs to take one measurement).

We note $\varepsilon_{Tot}^{Sens^{(n)}}$ the sensing energy consumed by the sensor until the time-step n:

$$\varepsilon_{Tot}^{Sens^{(n)}} = n\, \Delta t\; cost_{sens} \qquad (10)$$

Finally, the total energy consumed by a sensor node until the time step n is designated by $\varepsilon_{Tot}(n)$:

$$\varepsilon_{Tot}(n) = \varepsilon_{Tot}^{Rad^{(n)}} + \varepsilon_{Tot}^{Sens^{(n)}} \qquad (11)$$

3 COORDINATION STRATEGIES

To coordinate the movement of maintainer robots upon occurred failures in the WSNs, [1] has proposed a centralized manager algorithm to detect, report and handle failure. We start with this algorithm that we modify to obtain the Centralized Manger Strategy in the following subsection. Then, we propose CMSA to achieve fault-tolerance and provide a continuous service for applications supported by WSNs.

3.1 The Centralized Maintenance Strategy

In our study, we assume a connected network and an area covered by static wireless sensors. We suppose also that all nodes know their location by means of a localization technique like the GPS (Global Positioning System) [9].

At the beginning, robots are uniformly distributed over the considered area. One robot is selected to act as a central manager. Failures are detected by the guardian nodes and reported to the central manager robot. The manager then forwards the report to a selected maintainer robot to handle the failure. Like the centralized manager algorithm, we distinguish three stages in this strategy:

In the first stage (the initialization stage), we have an exchange of three types of messages: (i) Sensor node broadcast messages: this message containing the identity of sensor node and its location is periodically broadcast to the one-hop neighbors sensor nodes. Upon receiving such messages, any sensor node of the network can determine the list of its guardees (its one-hop neighbors). Hence, the guardian-guardee relationship between sensor nodes is established, (ii) Manager broadcast messages: The manager robot broadcasts periodically its identity over the network (to all sensor nodes and maintainer robots) and (iii) Maintainer reply messages: When a maintainer robot receives a manager broadcast message, it responds with a message containing its identity and location. Hence the manager can determine the location of all maintainer robots in the network.

To establish a guardian-guardee relation between nodes in WSN, the centralized manager algorithm [1] supposes that a guardee node selects the nearest node as a guardian. This choice presents some drawbacks. Indeed, if a failure occurs on the guardee node that is the relay point for the guardian to the manager robot, the report failure cannot be sent to the manager robot. In our strategy, we propose that any node is guarded by all its one- hop neighbors. In this case we guarantee that at least one report failure is received by the manager.

After initialization, any sensor node periodically sends its broadcast message (message type 1) to its one-hop neighbor nodes. If a guardian has not received any messages from a guardee for a certain amount of time, the guardian deduces that the guardee has failed and sends a report failure containing the identity and the location of failed node to the manager robot (the failure detection and reporting stage).

To report failures, Mei and all in [1] have used a geographic routing protocol (knowing that we assume network connectivity and area coverage). But upon multiple simultaneous failures, this assumption is not realistic. So, using the nearest node to the manager as a relay point by a guardian node to send the report failure may not be a good choice to communicate with the manager. For this reason, we have used in CMS an adhoc routing protocol such as AODV (Ad hoc On Demand Distance Vector) [10] to ensure that the report failure is received by the manager (if a route exists).

In the last stage (Failure handling) and upon the reception of a failure report, the manager selects the maintainer robot whose current location is the closest to the failure, and sends an order for this robot to handle the failure. Once the maintainer has finished the task, it sends to its manager robot a message containing its new location.

In [1], to handle failure, the manager selects the nearest maintainer robot to the failure regardless of its state (busy or available) and the requests are served by the maintainer robot as first-come-first-served. Using this technique, we risk an overload on robots which are close to the sink node, given that the network activity of nodes (and consequently the failure by energy depletion) increases in neighborhood of the sink node. In our strategy, we suppose that the repair requests queue is managed by the manager robot. Then, to handle failure, the manager robot selects the closest available maintainer robots to repair a given failure to ensure load sharing over robots.

3.2 The Centralized Maintenance Strategy with Anticipation

The main goal of the CMSA is to predict the energy failure. To achieve this goal we must compute the energy consumed by the sensor in the future. We use then our analytical energy model presented in section 2. Each sensor node must first measure the different transition probabilities ($Pij \quad 1 < i,j < 4$). Once the transition probabilities are determined, they are sent with the value of the residual energy ε_0 and the node location to the manager robot. With the reported measurements, the manager can compute the lifetime of sensors using the analytical model. Indeed, it is equivalent to finding the time ttl corresponding to:

$$\begin{cases} \varepsilon_{Tot}(ttl) & < \quad \varepsilon_0 \\ \varepsilon_{Tot}(ttl+1) & > \varepsilon_0 \end{cases} \tag{12}$$

Where: $\varepsilon_{Tot}(n)$ is calculated according to the equation 11.
Therefore, with our proposed analytical model, the manager robot can first estimate the N expected sensor failures in the network designated by f^i ($i \in [1, N]$). To determine the maintainer robot of an estimated failure f^i, the robot manager identifies the list of robots able to handle the anticipated failure at its required replacement time $f^i.ttl$ with a null off-service time and selects among them the nearest robot as the f^i maintainer robot. To ensure a null off-service time for the failure f^i , the selected maintainer robot R_{f^i} must be scheduled to repair the expected failure at:

$$f^i.ttl - ||R_{f^i}, f^i||.S_{R_{f^i}} \tag{13}$$

With:

- $||R_{f^i}, f^i||$ the distance between the current position of R_{f^i} and f^i.
- $S_{R_{f^i}}$ represents the speed of R_{f^i}.

Similarly to the CMS, we have three stages: the initialization stage, failure detection and reporting and failure handling. However, in the initialization stage, we have an additional type of messages: Measures transition probabilities messages that contain the transition probabilities matrix sent by sensor nodes to the manager robot. In the failure handling stage, the manager robot can send an order to repair the failure before it occurs. This order is based on the analytical model computed for each node sensor.

4 EXPERIMENTS

In this section, we validate our analytical model to estimate the energy consumption of a sensor node with different MAC layer protocols. We also compare the performances of the two coordination strategies presented in this paper.

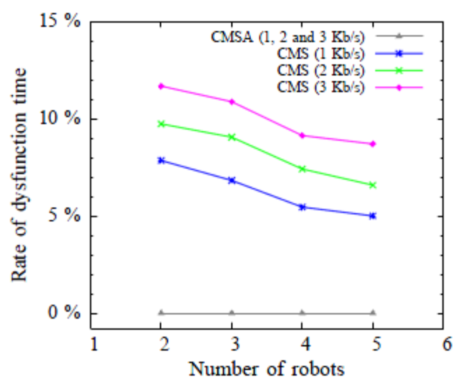
4.1 Experimental Setup

We have implemented our analytical model and the two coordination strategies in the NS-2 Simulator [11]. We have implemented also an on-demand mobility model in which robots move on demand after receiving a failure report. The failure detection, failure report, failure repair request and the analytical model are implemented at the application level. We have used AODV as a routing protocol.
We have selected the following simulation parameters: the sensor area is $200\text{x}200\ m^2$ with a total of 50 sensor nodes; the robot's speed is $1m/s$, based on the specification of Pioneer 3DX robots [12]; the number of robots varies from 1 to 11 robots; the simulation time is 86400 seconds and the sensor node generates a traffic with a constant bit rate and a constant packet size of 128 bytes.
IEEE 802.15.4 is used as a sensor MAC Layer in the WSN. The sensor power consumption in each MAC layer state ($sleep$, $transmit$, $receive$ and $idle$)

used in simulation is equal to (0.1404 Watt, 0.1404 Watt, 0.0018 Watt, 0.000018 Watt). References values are taken from a ZigBee node implementing IEEE 802.15.4 medium access.

4.1 The Network Dysfunction Time

In a first scenario, we suppose that sensor failures are caused only by energy depletion. Experiments are run with different bit rates per sensor node: 1, 2 and 3 kb/s. With CMS and CMSA, simulation has shown that we must have at least 2 robots to guarantee the network connectivity and coverage of the considered area. We designate by network dysfunction time, the sum of duration for which the network connectivity and the coverage are not guaranteed. In other words, it's the period during which at least one node is failed in the WSN. Figure 1 represents the variation of the percentage of the network dysfunction time when CMS and CMSA are used for different number of maintainer robots with different node bit rates (1, 2 and 3 Kb/s).

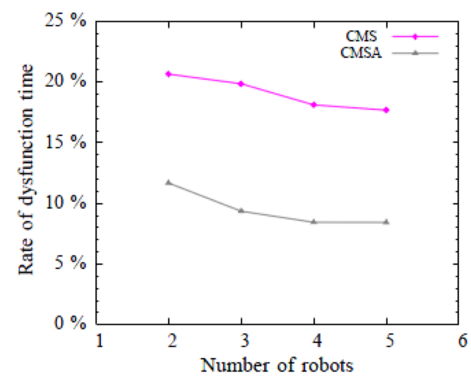


1. CMS vs CMSA: The network dysfunction time rate versus the number of maintainer robots with energy depletion failures only.

Using CMSA, figure 1 shows that only 2 robots are fully sufficient to guarantee the network connectivity and coverage of the considered area with a null network dysfunction time for the considered node bit rates: 1,2 and 3 Kb/s.

Minimizing the network dysfunction time is the primary requirements for real-time applications. To achieve this goal by using CMS, the solution, as the figure 1 shows, is to increase the number of maintainer robots in the WSN. This solution cannot be retained not only because it is expensive but also because the improvement given by this solution (on the network dysfunction time) remains always limited by the failure detection time and the failure handling time spent by robot to move to the occurred failure.

On the other hand, simulation results show that CMSA can provide a null network dysfunction time with the minimal number of robots used by CMS if only energy depletion failures are considered.



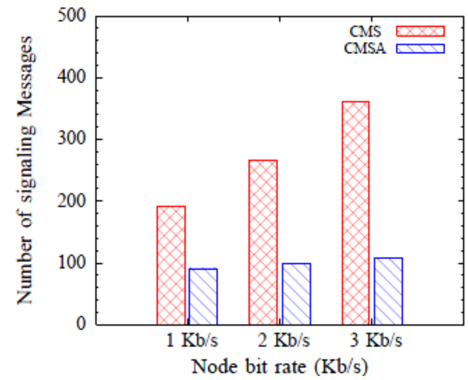
2. CMS vs CMSA: The network dysfunction time versus the number of maintainer robots with unpredictable failures.

But generally, failures occurred in WSNs are caused by many reasons other than the energy depletion such as hardware failures. To reflect this kind of failures in the simulation, we propose a

second scenario where we add random failures generated according to a Poisson distribution with an average of 2 failures per hour. Therefore, two types of failures can occur: failures caused by energy depletion which are anticipated and repaired before they happen and unpredictable failures which are detected by guardian nodes and reported to the manager.
We represent in figure 2 the network dysfunction time given by CMS and CMSA for different number of maintainer robots and a fixed node bit rate equal to 3Kb/s. This figure shows that CMSA reduces considerably the network dysfunction time using the minimal number of robots. In fact, the network dysfunction time provided by CMSA is due only to the repair time of unpredictable failures generated during the simulation, which explains the low percentage of the network dysfunction time compared to CMS.

4.1 Message Overhead

The message overhead is measured as the number of transmissions incurred by failure detection, failure reporting, and coordination messages. Since the two strategies are similar in number of coordination messages, we focus on the overhead introduced by failure detection and failure reporting.
For CMS, when a failure happens, all the guardians (the one-hop neighbors) of faulty node report the failure to the manager robot. However, the use of CMSA, implies one message for each sensor node sent to the manager containing the probability transition matrix at the initialization stage.
As the figure 3 shows, the number of messages induced by CMSA is equal to the initial sensor number plus the number of replaced nodes.



3. Message overhead versus node bit rate.

However, the Centralized Manager Strategy gives a high rate of signaling messages that increases considerably with the failure number. In fact, using the original strategy, all guardians of a failed node must send a report failure to the manager robot. This explains the important number of signaling messages.

6 CONCLUSION

In this paper, we have presented a proactive maintenance strategy for fault-tolerant wireless sensor networks. We have presented an analytical energy model to anticipate sensor failures, in which each node of the sensor network is modeled by a Markov Chain. The state of the model depends on the used MAC layer.
Based on the proposed energy model, we have investigated two versions of failure repair strategies: CMS and CMSA. Two types of robots are distinguished: a manager robot and several maintainer robots. In the CMS version, the manager receives a report failure from a guardian node upon a failure. Thereafter, it selects the closest maintainer robot to handle the occurred failure. For a real-time sensor

network, the reaction time upon a failure cannot be tolerated as simulation results have shown. To remedy this problem, we have proposed an anticipated version of Centralized Manager Strategy: CMSA. In this technique, the manager receives the transition probabilities matrix from each node sensor in the network. The transition probabilities are used by the analytical model to estimate the lifetime of sensor nodes. Before the estimated sensor lifetime expires, the manager sends an order to a maintainer robot to replace the corresponding sensor. Simulation results have shown that using this technique, we obtain a null dysfunction time of the wireless sensor network with a minimal number of robots.

In a future work, we propose to improve the two distributed strategies given in [1] to coordinate the movement of robots based on fixed distributed manager algorithm and dynamic distributed manger algorithm. Thanks to our analytical model predicting the lifetime of a sensor node, these strategies will be made able to support real-time applications.

6 REFERENCES

1. Y. Mei, C. Xian, Y. Charlie , Repairing Sensor Network Using Mobile Robots, Workshop on Wireless Ad hoc and Sensor Networks, 2006.
2. Y. Gao, K.Wu, and F. Li, Analysis on the redundancy of wireless sensor networks, ACM international conference on Wireless sensor networks and applications, September 2003.
3. A.Kumar, E. Altman, D. Miorandi, and M. Goyal, New Insights from a Fixed Point Analysis of Single Cell IEEE 802.11 WLANs, In Proceedings of IEEE INFOCOM, 2005.
4. IEEE 802.11 standard for Wireless Local Area Networks, http://standards.ieee.org/getieee802/802.11.html.
5. B. Bougard, F. Catthoor, C. Daly, A. Chandrakasan, and W. Dehaene, Energy efficiency of the IEEE 802.15.4 Standard in dense wireless microsensor network: Modeling and improvement perspectives , Design, Automation and Test in Europe,2005.
6. W. Ye, J. Heidemann and D. Estrin , An Energy-Efficient MAC Protocol for Wireless Sensor Networks, IEEE Infocom 2002
7. S. Tang, An analytical traffic flow model for cluster-based wireless sensor networks, International Conference on Pervasive Computing, 2006.
8. S. Mahfoudh, Energy effciency in wireless ad hoc and sensor networks: routing, node activity scheduling and cross-layering, thesis INRIA 2010.
9. A. Nemra and N. Aouf, Robust INS/GPS Sensor Fusion for UAV Localization Using SDRE Nonlinear Filtering, Sensors Journal , 2010.
10. C. Perkins, E. Belding-Royer and S. Das, *Ad hoc On demand Distance Vector (AODV) Routing*, IETF RFC 3561, 2003.
11. The Network Simulator ns-2, *http://www.isi.edu/nsnam/ns/*
12. Y. Mei, Y.-H. Lu, Y. C. Hu, and C. G. Lee, *A Case Study of Mobile Robot's Energy Consumption and Conservation Techniques*, In International Conference on Advanced Robotics, pages 492–497, 2005.