# Efficient Cluster-based Fault-tolerant Schemes for Wireless Sensor Networks

Mohamed LEHSAINI
STIC Laboratory
University of Tlemcen
Tlemcen, Algeria
m_lehsaini@mail.univ-tlemcen.dz

Hervé GUYENNET
LIFC Laboratory
University of Franche-Comté
Besançon, FRANCE
herve.guyennet@univ-fcomte.fr

Mohammed FEHAM
STIC Laboratory
University of Tlemcen
Tlemcen, Algeria
m_feham@mail.univ-tlemcen.dz

***Abstract*—The deployment of wireless sensor networks (WSNs) over a geographical area for monitoring physical phenomena is prone to several failures due to energy depletion, environmental hazards, hardware failure, communication link errors, etc. These failures prevent them to fulfill their tasks normally. In addition, in safety applications, these failures lead to hazardous consequences. Thus, it is necessary to adopt an efficient fault-tolerant approach to ensure the availability of sensor data anytime and anywhere in WSNs. In this paper, we proposed two efficient cluster-based fault-tolerant schemes enabling to reduce communication and processing overhead. These schemes are respectively denoted ECFS-1 and ECFS-2. ECFS-1 could tolerate link failures and therefore guarantee routing reliability while ECFS-2 could tolerate both sensor faults and links failures. Finally, we conducted several simulations to illustrate the effectiveness of our contribution and compared obtained results to other schemes.**

***Keywords: Cluster-based, Fault-tolerance, Failure nodes, WSNs.***

## I. INTRODUCTION

Wireless sensor networks (WSNs) are self-organized networks that typically consist of a large number of low-cost and low-powered sensor devices, called sensor nodes, which can be deployed over a geographical area for monitoring physical phenomena like temperature, humidity, vibrations, seismic events, and so on [1]. Now, WSNs are permeating a variety of application domains such as avionics, environmental monitoring, structural sensing, tele-medicine, space exploration, and command and control.

Typically, a sensor node is a tiny device that includes three basic components: a sensing unit for data acquisition from the physical surrounding environment, a processing unit for local data processing and storage, and a wireless transceiver, which is used to transform the captured events back to the base station. Sensor nodes are usually powered by lightweight batteries, and replacing or recharging these batteries is often not feasible because sensor nodes may be deployed in a hostile or unpractical environment. These sensor nodes collaborate with each other to perform tasks of data sensing, data communication, and data processing.

Moreover, WSNs should have a lifetime long enough to fulfill the application requirements. However, In addition to resource constraints in WSNs, the failure of sensor nodes is almost unavoidable because the latter are prone to failure due to energy depletion since they have usually deployed in hostile environments and their batteries cannot be recharged or replaced, hardware failure, communication link errors, and so on. Therefore, fault tolerance has become more important as other performance metrics such as energy efficiency, latency and accuracy in supporting distributed sensor applications.

In WSNs, failures can occur for various reasons. First, sensor nodes are fragile, and they may fail due to depletion of batteries or destruction by an external event. Besides, sensor nodes may capture and communicate false readings because of environmental influence on their sensing components. Second, as in any wireless networks, links are failure-prone [2,3,4], causing network partitions and dynamic changes in network topology.

In general, the consequence of these failures is that a node becomes unreachable, violates certain conditions that are essential for providing a service or returns false readings which could cause a disaster especially in safety critical applications. Furthermore, the above fault scenarios are worsened by the multihop communication nature of WSNs. It often takes several hops to deliver data from a sensor node to the remote base station; therefore, failure of a single node or link may lead to missing reports from the entire region of WSNs.

Therefore, since sensor nodes are prone to failure, fault tolerance should be seriously considered in many sensor network applications. Recently, several studies have dealt with fault tolerance in WSNs, particularly in the routing process. Moreover, these works focus on the detection and recovery of failures in WSNs and aim to reduce the amount of time required for detecting and recovering from a failure as much as possible.

These observations show that the design of new fault tolerant protocols has become necessary for sensor applications to operate successfully. Moreover, these protocols should ensure reliable data delivery while minimizing energy consumption.

In this paper, we clarify the requirements for maintaining high level availability in WSNs, and investigate briefly the schemes utilized in WSNs research and engineering for fault detection and recovery at the routing level. Then, we propose

an Efficient Cluster-based Fault-tolerant Scheme (ECFS) to tolerate faults in WSNs while dissipating less extra energy and time.

The proposed scheme is based on a clustered architecture in which the clusters have a primary cluster-head and secondary one. Sensor nodes with stronger capabilities are elected as cluster-heads and thus they can perform operations for other sensor nodes that would either have to spend a significant amount of energy or would not be capable of performing these operations. Cluster-heads could aggregate sensor data before it is forwarded to a remote base station, thereby saving energy. Furthermore, in clusters, the dual cluster-heads cooperate with each other to reduce extra costs by sending only one copy of sensed data to the sink; also, dual cluster-heads check errors with each other during the collecting sensor reading. In this optic, we proposed two schemes denoted ECFS-1 and ECFS-2. ECFS-1 could tolerate link failures and therefore guarantee routing reliability while ECFS-2 could tolerate both sensor faults and links failures. Finally, we conducted several simulations to demonstrate the effectiveness of our contribution and we compared obtained results to those of GRAB [5].

The rest of this paper is organized as follows: in Section 2, we present a survey of approaches to fault detection and recovery techniques; Section 3 illustrates our fault-tolerant scheme; and Section 4 presents a performance analysis of the proposed scheme. Finally, we conclude our paper and discuss future research work in Section 5.

## II. RELATED WORK

Many recent studies dealt with fault tolerance in WSNs especially in terms of routing data to the base station. These approaches have utilized a multipath routing to guarantee reliable delivery of data to the base station because route redundancy increases the probability to reach the destination. Furthermore, in these approaches, the authors aimed to ensure load balancing among sensor nodes.

In this section, we summarize and compare existing fault tolerant techniques that allow guaranteeing a reliable routing in WSNs.

In [5], the authors have proposed a meshed multipath routing called GRAB that allows creating a forwarding mesh from the source to the sink based on the cost of delivering data at each node. Therefore, nodes farther away from the sink have the highest cost of delivering data. Sensor readings propagate along the path of least cost towards the base station. In this technique, the resulting mesh is base on a credit system in which the amount of credit assigned by the source node to the packet enable to determine the width of mesh. GRAB ensures reliable delivery of data to the base station but it consumes more energy which makes it undesirable for WSNs deployed in hostile areas.

Node-disjoint multipath [5] generates a number of alternate paths that do not share any nodes with the primary path or other alternate paths except the source and the destination nodes. This scheme ensures that failures in some nodes on the primary path do not affect alternate paths therefore the delivery of data to the base station would be guaranteed. Creating multiple disjoint routes to the base station requires that the global network topology is known. Therefore, this technique consumes more energy because there is a redundancy of data sent to the base station.

In [6], the authors have proposed a braided multipath technique that consists to use braided or partially disjoint paths. For each node on the primary path, an alternate path not including that node is determined and these alternate paths are not much more expensive than the primary path in terms of latency and overhead. This technique guarantees recovery when a few nodes on the primary path fail. However, when most of the nodes on the primary path fail, new path discovery is required, which generates significant additional overhead.

## III. CONTRIBUTION

Clustering approaches are used to enable the sensors to form a cluster. Thus, once the network is partitioned into smaller logically disjoint clusters, it is easy to keep track of sensor nodes in a cluster by carrying out maintaining cluster operation which relates managing cluster information when sensor nodes leave a cluster i.e. either when they break down or exhaust their energy, and a mechanism for communication across the clusters.

In this section, we present our proposed distributed scheme that enables to generate balanced clusters as well as maintaining them. To perform our scheme, we assume that:

- All sensors are homogeneous with constrained energy and the same transmission range,
- Sensors are stationary,
- Sensors have 2-hop neighborhood positional knowledge and operate asynchronously without a centralized controller,
- Each sensor is able to calculate its weight according to its 2-density and residual energy.

### A. Cluster Formation

Clusters formation process consists to generate 2-hop clusters (2-clusters). Each cluster has a primary cluster-head and its vice, which are elected in 2-neighborhorhood based on the weights of sensors. The weight of each sensor is a combination of 2-density and residual energy as in Eq. (1). The weight parameter is periodically calculated by each node as shown in figure 1 in order to indicate the suitability of a node for playing cluster-head's role. We involve 2-density ($\delta_2$ (u)) factor in the purpose to generate clusters whose members are associated with cluster-heads and remaining energy (E(u)) parameter to select the nodes with more energy in their 2-neighborhood.

$$Weight\ (u) = \alpha * \delta_2(u) + \beta * E(u) \qquad (1)$$
$$where \quad \alpha + \beta = 1$$

Since cluster-heads are responsible to fulfil several tasks such as coordination among the cluster members, transmission gathered data to the remote base station, and management of their own cluster; we propose to set up periodically cluster-

heads election process after each round. Therefore, cluster-heads do not rapidly exhaust their battery power.

The proposed scheme is performed in two consecutive phases: set-up and steady-state.

*1) Set-up phase*

At the beginning of each round, each sensor calculates its weight and generates a 'Hello' message including three extra fields addition to other regular contents: Weight, $P\_Node_{Ch}$ *and* $V\_Node_{Ch}$, where $P\_Node_{Ch}$ *and* $V\_Node_{Ch}$ *are* set to zero. Then, it broadcasts them to its 2-neighborhood via a 'Hello' message as well as it eavesdrops its neighbor's 'Hello' message. After these exchanges, the sensor that has the greatest weight in its 2-neighborhood, is elected as primary cluster-head and the node that has the second largest weight, be chosen as its vice during the current round. Each sensor node updates its state vector by assigning respectively to $P\_Node_{Ch}$ and $V\_Node_{Ch}$ the identifiers of the corresponding primary cluster-head and its vice. Then, primary cluster-head broadcasts an advertisement message *(ADV_CH)* including its state vector to its 2-neighboors to request them to join it. Each sensor that receives the massage and does not belong to any cluster as well as its weight is lower than CH's weight, transmits *REQ_JOIN* message to *CH* to join it. Corresponding cluster-head checks if its own cluster size does not reach $Thresh_{Upper}$, it will transmit *ACCEPT_CH* message to this sensor. Finally, cluster-heads construct a cluster-to-cluster (CH-to-CH) routing paths to use them for data transmission.

After the end of this phase, each cluster-head creates a time schedule, in which time slots are allocated for intra-cluster communication, data aggregation, inter-cluster communication, and maintenance process. Then, the generated clustered sensor network starts the steady-state phase of round to transfer collected data to the remote base station.

*2) Steady phase*

Sensor nodes within a cluster do not transmit their gathered data directly to the sink, but only to their respective cluster-heads. Accordingly, the cluster-head are responsible for coordination among the cluster members, aggregation of their data, and transmission of the aggregated data to the sink, directly or via multi-hop transmission.

During the steady-state phase, sensor nodes can begin sensing and transmitting collected readings to their respective cluster-heads. The radio of each non-cluster-head sensor can be turned off until the sensor's allocated transmission time. The cluster-heads, after receiving all data, aggregate it before sending it to the remote base station. Each cluster-head communicates using different CDMA codes in order to reduce interference from nodes belonging to other clusters.

*B. Fault tolerance for sensor data*

To deal with erroneous sensor data, the cluster-head analyzes sensor readings by calculating their average value and standard deviation. If the standard deviation exceeds a threshold value, the cluster-head removes the readings that differ from the average of the threshold value.

We consider that each sensor $s_i$ captures a physical value $\theta_i$ within its vicinity and sends it to the cluster-head. The latter calculates the average of all measurements received denoted $\bar{\theta}$ and the standard deviation $\sigma_\theta$ associated according to the following:

$$\bar{\theta} = \frac{\sum_{i=1}^{k} \theta_i(s_i)}{k} \quad (2)$$

$$\sigma_\theta = \frac{1}{k}\sqrt{\sum_{i=} \left(\theta_i - \bar{\theta}\right)^2} \quad (3)$$

For each sensor data $\theta_i(s_i)$, if $\left|\theta_i - \bar{\theta}\right| > \sigma_\theta$ , $\theta_i$ will be removed. Then, the cluster-head recalculates the average values of correct readings. Let $k_{Cor}$ is the number of these values. The new average value is:

$$\overline{\theta_{Cor}} = \frac{\sum_{i=1}^{k_{Cor}} \theta_i(s_i)}{k_{Cor}} \quad (4)$$

*C. Availibility of WSN*

The availability of a service provided by a WSN depends on the fault tolerance technique. Whatever the technique used for fault tolerance, failures are unavoidable in WSNs that make them unavailable for some time. For that, we introduce a metric denoted Avail (i) to evaluate the availability ratio when a failure i occurs. We assume that the failures occur according to a Poisson distribution. For a failure i, this metric is expressed as follows:

$$Avail\ (i) = \frac{TTF_i}{TTF_i + TTR_i} \quad (5)$$

Where $TTF_i$ represents the average time until a failure occurs in the network and $TTR_i$ is the mean time to repair it. Thus, the availability ratio of a WSN Avail(WSN) is calculated as:

$$Avail\ (WSN) = \frac{TTF}{TTF + TTR} \quad (6)$$

Where

$$TTF = \sum_{i=1}^{n} TTF_i$$

$$TTR = \sum_{i=1}^{n} TTR_i$$

## IV. EVALUATION AND SIMULATION RESULTS

In this section, we conduct extensive simulations to evaluate the performance of ECFS-2 and compare them with those of GRAB. For that, we utilize JSIM [7] to implement it and we select sensor hardware parameters similar to Berkeley

motes [8]. We use a field size of 150×150m$^2$ where 1200 nodes are uniformly distributed. The maximum transmission range of a node is 10 meters. The energy consumptions for transmitting, receiving and idling are respectively 60 mW, 12 mW and 12 mW. The time of transmission or receiving for a packet is 10 ms. A random source node generates a report every 10 seconds and in each run 100 reports are generated.

Node failures are randomly distributed over time and the fraction of failed nodes is defined as the node failure rate. To evaluate the performance of ECFS-2, we measure the success ratio, which is the ratio of the number of report packets successfully received at the sink to the total number generated at the source. This metric illustrates the degree of robustness of ECFS-2 to forward data in the presence of nodes failed.

Furthermore, we also measure total energy consumption to illustrate the robustness of ECFS-2 in terms of the cost of excessive overhead. The obtained results are averaged over 10 different runs.

We evaluate respectively the impact of rate failed nodes and network size on success ratio and energy consumption. Finally, we compare the obtained results to those of GRAB [3].

### *A. Node failure rate*

To illustrate the robustness of ECFS-2, we evaluate the success ratio of data delivery to the base station according to the number of failed nodes. For that, we vary the rate of failed nodes from 5% to 50%.

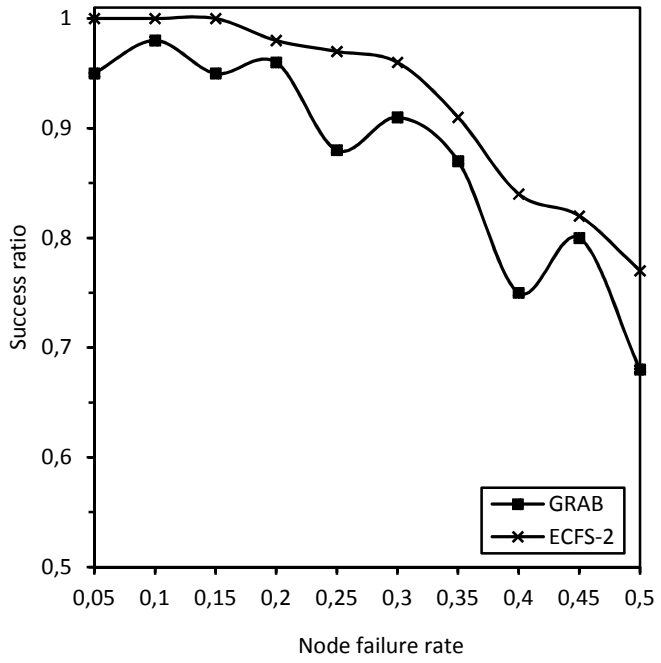


**Figure 1: Success ratio for different node failure rate**

Figure 1 shows the success ratio according to the rate of failed nodes. The success ratio is above 90% for rates of failed nodes that are below 35%. As the rate of failed nodes increases, the success ratio tends to decrease. However, ECFS-2 still maintains very high degrees of robustness compared to GRAB. The success ratio remains above 80% when 45% nodes fail, and is around 80% in the extreme case when half of the nodes fail. This shows that ECFS-2 is robust even with severe node failures compared to GRAB. The high success ratio also demonstrates that ECFS-2 is highly tolerate to inaccurate cost fields because the probability of having a cluster-head and its vice have failed and that they are involved in data forwarding, is small.

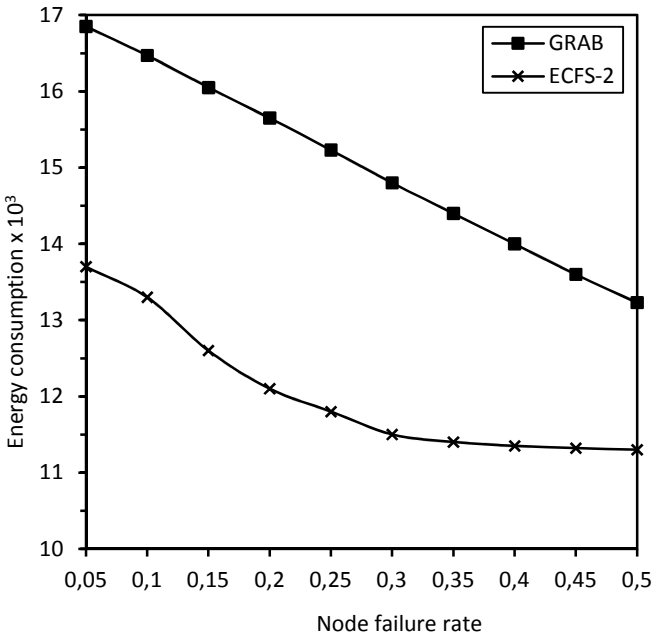


**Figure 2: Energy consumption for different node failure rate**

The energy consumptions are shown in figure 2. When node failure increases, the energy decreases linearly because the idle energy dominates the total energy consumption. A higher node failure rates means more node failures, thus proportionally less energy consumption. Furthermore, ECFS-2 consumes less energy than GRAB because ECFS-2 implies the cluster-heads and in the critical case their vices to forward data to the base station.

### *B. Impact of network size*

To find how network size can affect the robustness of ECFS-2, we keep the field size 150 × 150 m$^2$, while varying the number of nodes from 500 to 1500 and the node failure rate is 15%.

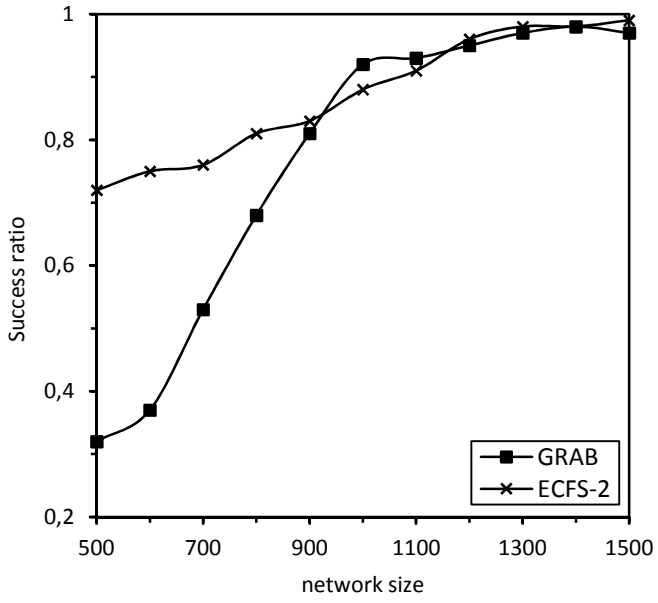


**Figure 3: Success ratio for different network sizes**

Figure 3 shows how the success ratio changes over different network sizes. The success ratio is above 72% for node numbers of 500 and 750, and it exceeds that obtained by GRAB. However, when network size exceeds 900 nodes, the success ratio remains high above 90% for all the remaining network sizes.

### *C. Availibility ratio*

The availability of services provided by a WSN depends on the duration of network operation and repair time of the failure. We assume that the failures occur following a Poisson distribution and the network becomes unavailable when there is a cluster-head and its vice that have failed and that they are involved in data forwarding.

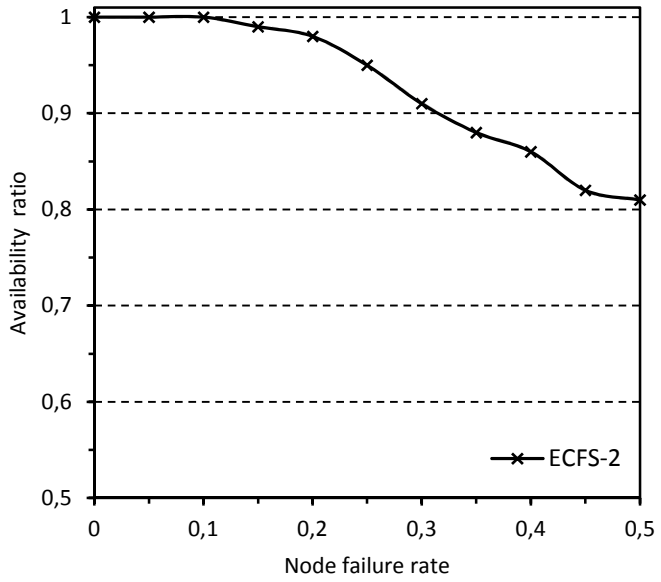


**Figure 4: Availability ratio according to node failure rate**

Fig.4 shows that the availability of services is greater than 88% when node failure rate is less than 40%. This reflects the probability of having a cluster-head and its vice have failed and that they are involved in data forwarding. This probability increases when node failure increases.

## V. CONCLUSION

In this paper, we have proposed two schemes for dealing with fault tolerance in WSNs.

The first approach enables to ensure reliable delivery of data to the base station while minimizing energy consumption to allow a long network lifetime whereas the second allows ensuring the reliability of collection data and data delivery.

In both approaches, we utilized a clustered architecture in which there is a primary cluster-head and its vice. The latter receives the data sent by the cluster members and aggregates them as the primary cluster-head and if it observes that the primary cluster-head has not forwarded the aggregated data, it would do this.

Simulation results showed that in ECFS-2, the reliability ratio of delivery data is very high compared to GRAB.

## REFERENCES

[1] [1] Akyildiz I.F., Su W., Sankarasubramaniam Y., Cayirci E.,"Wireless sensor networks: a survey". Computer Networks (Elsevier). Vol. 38, no4, pp 393-422, March 2002.

[2] A. Woo, T. Tong, and D. Culler, Taming the underlying challenges of reliable multihop routing in sensor networks. In *ACM Proceedings of* the 1st international conference on Embedded networked sensor systems (*SenSys'03)*,pp.14-27, November 2003.

[3] L. Paradis and Q. Han, A Survey of Fault Management inWireless Sensor Networks, *Journal of Network and Systems Managemen, Vol. 15, No. 2, pp.171-190, 2007.*

[4] H.M. Ammari and S.K; Das, Fault tolerance measures for large-scale wireless sensor networks, Journal ACM Transactions on Autonomous and Adaptive Systems (TAAS), Vol. 4, No. 1, pp.1-28, 2009.

[5] Ye, G. Zhong, S. Lu, and L. Zhang. Gradient broadcast: A robust data delivery protocol for large scale sensor networks. SPRINGER Wireless Networks, 11(2), pp.285-298, 2005.

[6] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. ACM Mobile Computing and Communications, Review, 1(2),pp. 251-254, October 2002.

[7] http://www.physiome.org/jsim/

[8] J. Hill, R. Szewczyk, A.Woo, S. Hollar, D. Culler and K. Pister. System architecture directions for networked sensors, in: *International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-IX)* (2000).