Telecommun Syst
DOI 10.1007/s11235-016-0236-5

Novel approach for replacement of a failure node in wireless sensor network

Abdelmalek Boudries[1] · **Mourad Amad**[2] · **Patrick Siarry**[3]

© Springer Science+Business Media New York 2016

Abstract In the recent years, there has been a growing interest in wireless sensor networks (WSN). Network's lifetime depends on energy efficiency and load balancing where connectivity is a very important factor. However, such connectivity can be lost due to the failure of some sensor nodes which creates disruptions to the network operations, lead to a reconfiguration of the network by generating energy losses, or in another case, the network mission fails. Energy conservation is a very important problem in WSN. In this paper, we propose a new solution for the connectivity problem when failure nodes are considered. The replacement of failed nodes is done in two phases: the first one is the search of redundant nodes using the clusterheads; the second phase is a restoration of connectivity. Performance evaluation of the proposed replacement approach shows that the results are globally satisfactory.

Keywords Failure node · Connectivity restoration · Replacement of a failing node · Sensors networks

1 Introduction

Recent technological advances have led to the emergence of pervasive networks of both small and low power devices that integrate sensors and actuators with limited on-board processing and wireless communication capabilities [1]. The wireless communications play a crucial role in data-processing networks. They offer open solutions to provide mobility as well as essential services where the installation of infrastructures is difficult or not possible. These networks are under active development because of their interface flexibility allowing user mobility. The control of mobility is a critical issue in communication field, because the mobile environment is characterized by several critical aspects: a frequent disconnection, a modest debit of communication, and especially the limited source of energy.

Since the majority of the low power devices have batteries with limited lifetime and the replacement of these batteries on thousands of these devices is infeasible, especially in areas where access is difficult or sometimes even impossible, it is well approvable that a sensors network should be deployed with a strong density in order to extend the network lifetime [2]. In a high density network, if all the sensor nodes act in an active mode, then an excessive quantity of energy will be wasted. In one hand, the data of sensors gathered are likely to be strongly correlated and redundant. In the other hand, an excessive collision of packages can occur, because sensors send simultaneous packages with the presence of some releases events. Consequently, it is neither necessary nor desirable that all nodes act simultaneously in an active mode. One of the emerged questions in such a high density of sensors networks is the density control. In [3], the authors choose the prolongation of the operating time system while keeping only a necessary set of sensors in an active mode and putting the remained sensors in sleep mode. Another category

---

✉ Abdelmalek Boudries
abdelmalekboudries@gmail.com

Mourad Amad
amad.mourad@gmail.com

Patrick Siarry
siarry@univ-paris12.fr

[1] Laboratory LMA, University of Bejaia, Bejaïa, Algeria

[2] LaMOS Research Unit, Faculty of Exact Sciences, University of Bejaia, Bejaïa, Algeria

[3] Laboratory LiSSi, University of Paris-Est Creteil Val-de-Marne, Creteil 94010, France

of approaches for the connectivity maintenance in wireless sensor networks (WSN) is the restoration of the connectivity after a sensor node failure [4–7].

Supposing a WSN deployed in a difficult accessing zone, the lifetime of this network depends strongly on the connectivity factor between its nodes. Several factors can be at the origin of the connectivity rupture, between the nodes of the considered network, such as the lack of energy on the significant node level, infection of a vital node by a malevolent code and a logical/physical failure of a primary node. The failure of a sensor node can leave its entire zone (or a part of it) without coverage, and it can generate the partition of the network if it is a gateway node (*a relay node*). This means that the network will be divided into two or several small networks where some nodes can be disconnected from the network. This implies a loss of connectivity between the parts of the network. Our objective is to restore connectivity after failure of a sensor node by taking into account the constraint of energy.

In this paper, we propose a new approach for the replacement of a failed node (called NARF), while considering the network lifetime. In our proposition, the consumed total energy for the restoration of connection is shared by several nodes, so that the consumption of individual energy would be tiny and thus extending the network lifetime.

The remainder of this paper is organized as follows: in Sect. 2, we present some related works on failure node detection and replacement methods. Section 3 describes some important processes. In Sect. 4, we introduce our proposed approach called $NARF$ (Novel approach for replacement of a failure node in WSN), we first describe the network clustering as our solution uses it. The performance evaluation is presented in Sect. 5. Finally, in Sect. 6, we conclude and state prospects.

2 Related work

Several works are proposed for the connectivity maintenance problem in WSNs. In the literature, the existing approaches can be classified in two classes (see Fig. 1).

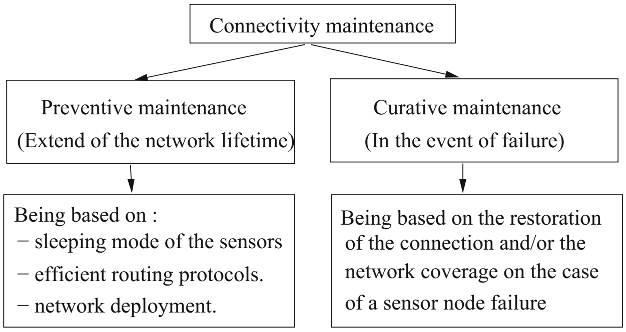


Fig. 1 Classification of the connectivity maintenance solutions

In the first class, authors try to maintain the sensors network connected longest time possible; they seek solutions to extend the lifetime of the network like [1,8,9] which try to use, at a given time, a minimum number of sensors that ensure the connectivity and/or the network coverage. Efficient routing in a sensor network requires that the routing protocol must minimize energy dissipation [10–12]. In [13], Samia and Shreen propose an approach where fault tolerant is incorporated for chain based routing protocols. They proposed two techniques of fault detection and recovery in chain based routing protocols. The two techniques employ the same strategy for fault detection. Each sensor node in every chain identifies whether its successor neighbor in its chain is faulty by NOTIFY and READY messages. However, they proposed two different techniques for fault recovery phase. The first technique overcomes the fault by having every predecessor node to a failed node instead of sending its data to the failed node forward it to the successor node of the failed node. The second technique gets around the fault by choosing a backup node for the faulty from the neighboring chain closest the sink which satisfies minimum energy consumption. The authors of the second class work on the connectivity maintenance in case of failure of a sensor node [4–7]. They try to solve the connectivity problem while considering the network partitioning and/or the covering of the monitoring zone problem. We classify our work in the second class (*Curative maintenance*).

The authors in [14] propose two energy efficient protocols for connectivity maintenance of WSNs whenever backbone nodes move or deplete of energy. Kang and al. [15] propose the distributed coverage hole detection and recovery algorithms, it is based on boundary critical points, and can be run on a single node with verifying boundary critical points from neighbors. In [16], a modification of an existing topology management scheme, Naps, is proposed to ensure very minimal data loss while also handling node failures. The authors propose to conserve energy to prolong the lifetime of the nodes and the sensor network while maintaining a constantly connected sensor network. When detecting the failing node, a neighbor node can be selected at random and the energy in this random node is increased in order to increase the communication range of the node. Another alternative is to be more selective in determining which node to increase the energy to offer compensation for a failed node and pick the neighbor node which has the most energy remaining. Authors in [17] deal with heterogeneous sensors equipped with actuation facilities to assist in the sensor self-deployment, they consider the problem from two perspectives: global deployment of sensors and local sensor network repair. A coverage-aware sensor automation protocol is proposed to realize an automated smart monitoring network. Two centralized algorithms are included in the protocol suite: enhanced virtual forces algorithm with boundary forces and sensor self-organizing algorithm. In [18] authors focus on reducing the response time in the case of node failures, reducing the overall moving distances and prolonging overall network lifetime in that way.

In the paper of Tamboli and Younis [5], the replacement of the failed node is done only by its direct neighbors. The exhaustion of energy due to the repeated physical movements of these nodes is an obvious concern to be taken into account. If we limit the replacements only to the direct neighbors of this node, we probably cause the failure of several nodes in a short time and put in danger the entire network. However, if we want to extend the lifetime of the sensors network, the replacement function of the failed node must be shared by several neighbors using replacements chain. These replacements will continue until arriving at a node having a coverage range completely covered by its neighbors, we call this node, redundant node (*in this case, the algorithm will finish*) or arrive at an extremity node less significant in terms of connectivity of the global network (*in this case, if we do not take into account the network coverage, the algorithm will finish. But if we take into account the network coverage, the algorithm will continue to be executed*).

In DARA approach [4], the failing node detection can't require an adjustment in the network topology if the node does not divide the network into parts. It means that the coverage zone, of the failing node, can be without coverage if the failure of the considered node does not partition the network even if there are redundant nodes in the neighborhood of the failing node. $DARA$ focuses on the connectivity maintenance without being concerned with the network coverage. In this approach, a substitute of the failure or the moved node leaves its zone definitively without coverage even if no other node can replace it afterwards. In terms of coverage, this node can be considered failing by leaving its zone without coverage.

In [7], the authors propose a DRFN approach (*detection and replacement of a failing node*) for the connectivity maintenance problem by carrying out a replacement chain according to a distributed algorithm; their solution generates a lot of messages overheads.

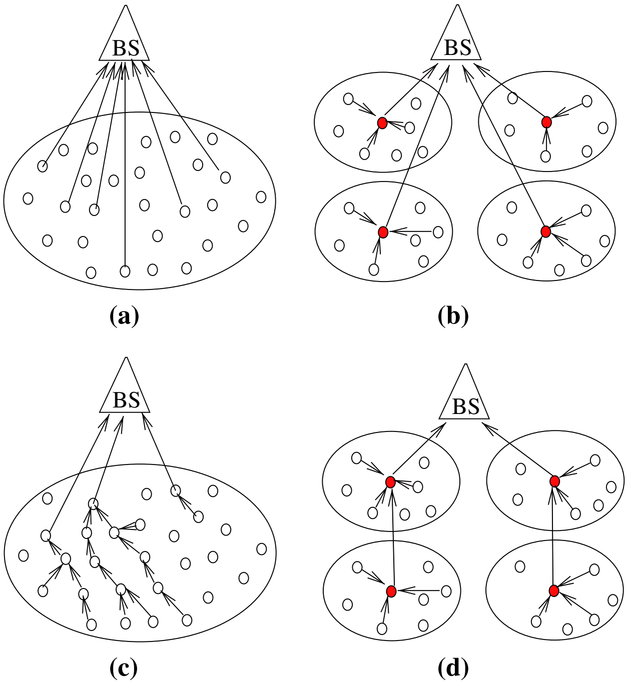
In order to give better performances, the architecture of our proposed solution is based on clustering the network.

3 Some important processes

In this section, we describe the clustering and the energy dissipation model used in our proposed method.

3.1 Clustering

Clustering is a method that aggregates the nodes into groups; these groups are known as clusters. Node clustering is a useful topology management approach to reduce the communication overhead and exploit data aggregation in sensor networks. WSN topologies are classified into four types of models as shown in Fig. 2 [19].



(a) (b)
(c) (d)

Fig. 2 Classification of WSN topologies. **a** Single-hop flat model, **b** Single-hop clustering model, **c** Multi-hop flat model, **d** Multi-hop clustering model

In the single-hop models (Fig. 2a, b), all sensor nodes transmit their data to the base station (BS) directly. These architectures are impractical in large-scale areas because transmission cost becomes expensive in terms of energy consumption and in the worst case, the base station may be inaccessible.

In the multi-hop models as in [19], the authors use the flat model (Fig. 2c) and the clustering model (Fig. 2d). In the multi-hop flat model, overhead and energy consumption can be increased, because all nodes should share the same information such as routing tables. On the other hand, in the multi-hop clustering model, sensor nodes can maintain low overhead and energy consumption, because particular clusterheads aggregate data and transmit them to the base station. In addition, wireless medium is shared and managed by individual nodes in the multi-hop flat model, which results in low efficiency in the resource usage. In the multi-hop clustering model, resources can be allocated orthogonally to each cluster for reducing collisions between clusters and be reused cluster by cluster. The multi-hop clustering model is appropriate for the sensor network deployed in remote large-scale areas.

In our solution, we took the model of Fig. 2d, by offloading clusterhead the task of routing between clusterheads. This

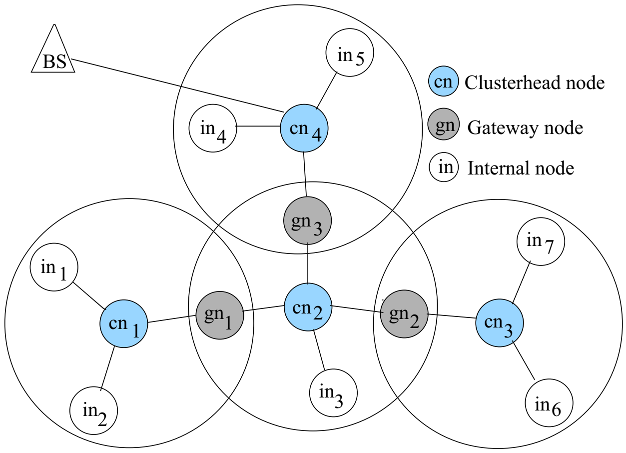


Fig. 3 Multi-hop clustering model

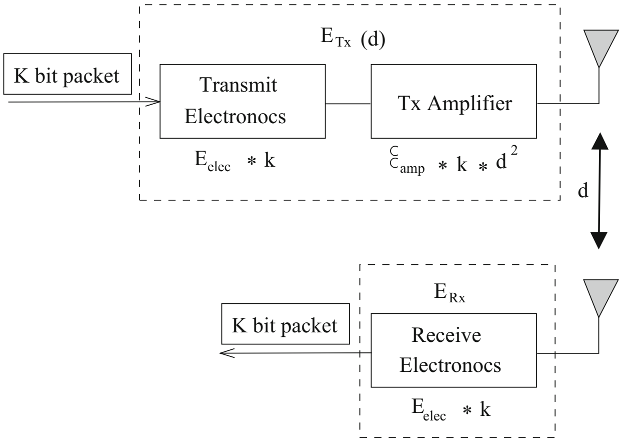


Fig. 4 First order radio model

task is assigned to the gateways. As shown in the multi-hop clustering model presented in Fig. 3, each cluster refers to a representative node. Representatives of clusters (called *clusterheads*) are responsible for coordinating the routing in their cluster, and Gateways are responsible for providing the connection inter clusters. The clusterheads election and the choice of Gateway nodes are well studied in the literature [20,21].

Electing a specific node as a clusterhead is a very important process. Various factors can be considered for electing the best node as a clusterhead [22]. Some of these factors include the location of the node with respect to other nodes, mobility, energy, trust and throughput of the node. Nodes of WSN have limited battery and resources. The process of election increases overall processing overhead of the network. So the election process must also consider the processing and the energy limitations of the nodes [23]. Thus, a compromise between the potential energy and the distance between its neighbors is essential; because if this distance is large, the probability that this node would be elected as a clusterhead for its neighbor must decrease, which decreases its weight. For that, we define the weight of a node (wn), according to both its percentage of energy and the distance separating it from its neighbor, in the Formula 1.

$$wn = \frac{\%energy}{\alpha \times distance} \tag{1}$$

where:

- $distance$ is the distance separating the node from its neighbor,
- α is an empirical variable (*fixed to* 1 *for simplification purpose*).

The weight of a node exchanged with its neighbors for the election of a clusterhead, varies according to the distance separating this node from the communicating node. A node that has more energy will have more chance to be elected clusterhead to its near neighbors than to the remote nodes.

3.2 Energy dissipation

Currently, there is a great deal of research in the area of low-energy radios. Different assumptions about the radio characteristics, including energy dissipation in transmit and receive modes, will change the advantages of different protocols. In our work, we use the model presented in Fig. 4 [24].

Thus, to transmit a k-bit message on a distance d using the radio model defined in [24], the radio expends:

$$E_{Tx}(k,d) = E_{Tx-elec}(k) + E_{Tx-amp}(k,d)$$
$$E_{Tx}(k,d) = E_{elec} * k + \varepsilon_{amp} * k * d^2$$

and to receive a k-bit message, the radio expends:

$$E_{Rx}(k) = E_{Rx-elec}(k)$$
$$E_{Rx}(k) = E_{elec} * k$$

where:

- $E_{Tx-elec}$: is the transmission electronics energy consumption,
- $E_{Tx-amp}/\varepsilon_{amp}$: is the energy dissipated by transmitter amplifier to achieve an acceptable signal
- $E_{Rx-elec}$: is the reception electronics energy consumption,

This model assumes that: $E_{Tx-elec} = E_{Rx-elec} = E_{elec}$.

4 NARF: novel approach for replacement of a failure node in wireless sensor network

The division of the network into a set of clusters facilitates the search of redundant nodes, which is the first phase of our solution. A redundant node is a node where its area is fully covered by its direct neighbors. The clusterheads look first to find redundant nodes in their cluster and broadcast to each other to update its information. The second phase is to restore connectivity.

4.1 Redundant nodes research process

After the network deployment and the clusterheads election, the clusterheads identify the redundant sensors. For this, each clusterhead identifies redundant sensors in its cluster and diffuses through the Gateway sensors, a message announcing its redundant sensors. As each clusterhead knows the redundant sensors in only its own cluster, a clusterhead receiving a message containing redundant sensors updates its RST table (*redundant sensors table*) and sends again the message to all Gateways except the one that receives this message for eliminating the infinite loop.

The enumeration process and the sending of redundant sensors list (RSL) executed by the clusterhead is described in Algorithm 1 (enum-red()), which describes the census of the redundant sensors by the clusterhead, and the sending of the redundant sensors list to its neighbors.

Algorithm 1 enum-red()

```
1: SL $\leftarrow$ {}; // Initialisation
2: For any sensor S $\in$ in its cluster Do
3:     If (S is redundant) Then
4:         RSL $\leftarrow$ RSL $\cup$ S.id_sensor; //S.id_sensor:identifier of the sensor S;
5:     End If;
6: End For;
7: For any sensor S $\in$ in its cluster Do
8:     If (S.typ_sensor = "Gateway") Then
9:         send RSL_message;
10:    End If;
11: End For;
```

Lines 2–6 describe the enumeration process of the redundant sensors in its cluster. Lines 7–11 describe the sending of *RSL_message* containing the RSL list, its affiliates and its location to all Gateways. The algorithm 1 complexity depends on the number of clusters and maximum number of nodes in a cluster. Let n be the number of clusters, and m be the maximum number of nodes in a cluster. The complexity of algorithm 1 is $O(mn)$.

The RSL_message receiving process is described in Algorithm 2 (RSL-receiv()). This latter is executed when a sensor node receives the redundant sensors list message, called *RSL_message*, it describes the actions to perform by the receiving node.

Algorithm 2 RSL-receiv()

```
1: If (RSL_message received) Then
2:     If (typ_sensor = "clusterhead") Then
3:         update RST; //Redundant Sensors Table
4:     End If;
5:     If (typ_sensor = "Gateway") Then
6:         For any sensor S $\in$ CL Do // CL: clusterheads list
7:             If (RSL_message source $\neq$ S) Then
8:                 send RSL_message;
9:             End If;
10:        End For;
11:    End If;
12:    If (typ_sensor = "internal node") Then
13:        ignore the message;
14:    End If;
15: End If;
```

Lines 2–4 describe the RSL_message receiving process by a Clusterhead. Lines 5–11 describe the RSL_message receiving process by a sensor Gateway. Lines 12–14 describe the RSL_message receiving process by an internal sensor node to cluster. The complexity of the algorithm 2 is $O(m)$, with m is the maximum number of nodes in a cluster.

4.2 Failing node detection process

After the network deployment, the sensor node sends a short message "*detect*" then waits for a pre-defined short time T before judging that the sensor node is failing. After time T expiration, which is equal to the message go-back time and processing time, if the sensor node, transmitting the message "*detect*", does not receive response, then it considers the recipient node of the message as a failing one. The process is still repeated after each pre-defined waiting time.

4.3 Failing node replacement process

Once a redundant node is designated to replace the failed node, we should determine how to move the sensor to the place of the failed node in order to replace it. A simple and obvious solution is to move the redundant sensor directly at the place of the failed node. However, it can take longer time than that required by the application. For example, a sensor monitoring a very important area fails and the application of this sensor requires that the space between the data measures should not exceed a certain period (eg. 30 s). If we assume that the redundant sensor is 50 m far from the failed sensor and it takes one minute to travel that distance, then it does

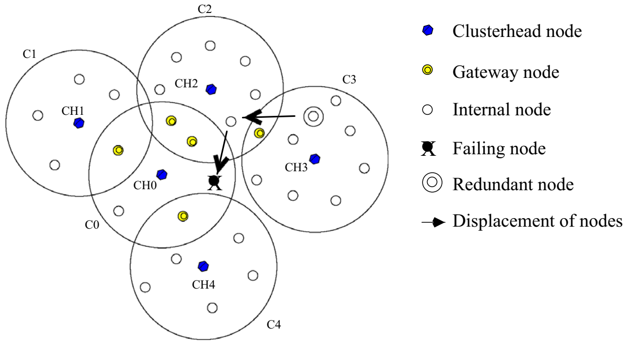


Fig. 5 Cascaded movement

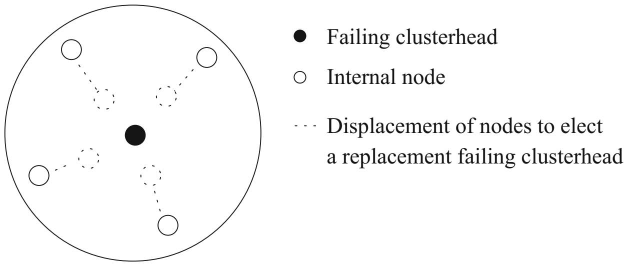


Fig. 6 Failing clusterhead detection

not accomplish the application requirements. Besides, traveling this distance by a sensor causes the consumption of a large amount of energy; it can cause failure of the moved sensor. So, sharing this energy loss by several sensors seems a reasonable solution. Thus, we use cascaded movement, presented in Fig. 5, to replace the failed node. For the following two reasons: less waiting to replace the failed node and sharing of consumed energy. It constitutes a key element for our proposed solution.

In Fig. 5, no direct neighbor of clusterhead $CH0$ has a redundant node, the clusterhead which will be charged to replace the failed node is $CH2$, because it has more nodes among the neighbors of the clusterhead $CH0$, where CHi is the clusterhead of the cluster $Ci\ /\ i \in \{0, 1, 2, 3, 4\}$. The clusterhead $CH2$, in its turn, searches among its neighbors the one which has a redundant node (or that has more internal nodes), and finds $CH3$ which asks its redundant node to move to replace the moved node of $C2$ cluster which has replaced the failed node of the cluster $C0$, from where the denomination of the cascaded movement, thus the zone of the failed node does not stay for a long time without surveillance.

If an internal sensor node detects a failed node fn or receives a message elect_message (*message for the election to replace the failed clusterhead*) (see Fig. 6), then it executes algorithm 3 (detect-fail()).

Algorithm 3 detect-fail()

```
1: If (sensor $fn$ failed detected) Then
2:     elected ← false;
3:     If (fn.typ_sensor = "clusterhead") Then
4:         elected ← true;
5:         Move a distance of Rc/2 to $fn$;
6:         Broadcast elect_message containing his amount of energy;
7:         Repeat
8:             wait a period of time t1;
9:             If (elect_message received and amount of energy < amount of energy on the transmitter sensor) Then
10:                elected ← false;
11:                back in its place;
12:            End If;
13:        Until (elected = false) or (time lapse t2 = $\alpha$t1);
14:        If (elected = true) Then
15:            to displace to ensure the functions of the failing clusterhead;
16:        End If;
17:    End If;
18: End If;
```

Algorithm 3 describes the actions to be performed by an internal node of the cluster in the case of failed clusterhead detection. Lines 9–12 describe the *elect_message* reception process by an internal node sensor. The complexity of the algorithm 3 is $O(in)$, with in is the number of sensor nodes.

When a clusterhead detects a Gateway node gn failing, then its cluster can be disconnected from the other clusters, especially if this is the only bridge that connects it with the rest of the network. So, its first priority is to reconnect by choosing an internal node of the cluster and to put it in the place of the Gateway node fails. If the Gateway node is detected by more than one clusterhead, this can be a problem. As a primitive solution, the clusterheads must first agree on who will load reconnection between their clusters as described in Algorithm 4. The complexity of this algorithm is $O(ch)$, with ch is the number of clusterheads.

Algorithm 4 detect-Gat()

```
elected ← false; // Initialisation
If (failed sensor gn detected and gn.typ_sensor = "Gateway") Then
    elected ← true;
    Move by a distance of Rc/2 to gn;
    broadcast elect_Gat_message containing the number of the internal nodes in his cluster;
    Repeat
        wait a period of time t1;
        If (elect_Gat_message received and nodes number < internal nodes number of the clusterhead sender of the message) Then
            elected ← false;
            back in its place;
        End If;
    Until (elected = false) or (time lapse t2= $\alpha$t1);
End If;
```

After the detection of the Gateway node failing by clusterheads which move a distance of Rc/2, and after a defined time sufficient for sending and receiving the messages ”*elect_Gat*”, the elected clusterhead executes the algorithm 5 (repl-Gat()), which describes the internal node election by the clusterhead to replace the failing Gateway. The clusterhead broadcasts a message to inform its sensor nodes and asks them to send their rate of energy, after receiving the response of all its nodes, the clusterhead elects the node which has a high rate of energy.

Algorithm 5 repl-Gat()

```
1: If (elected) Then
2:    For any sensor S ∈ in its cluster Do
3:       If (S.typ_sensor = ”internal node”) Then
4:          send repl_Gat_message;
5:       End If;
6:    End For;
7:    Repeat
8:       message waiting;
9:       If (resp_repl_Gat_message received) Then
10:         If (first message) or (amount_energy of the transmitter > amount_energy of the substitute) Then
11:            substitute ← transmitter;
12:         End If;
13:      End If;
14:   Until (receive the response of all internal nodes);
15:   send replacement_message to the elected node for replacement;
16: End If;
```

To replace the failed Gateway node, the clusterhead responsible of this function sends a replacement message ”*repl_Gat_message*” to the internal nodes in its cluster (*lines* 2–6). After the reception of the replies to messages repl_Gat_message, the clusterhead selects internal node with a higher amount of energy (*see lines* 7–14). Once it has received responses from all internal nodes, the clusterhead sends replacement_message (*line* 15) to the elected node; containing the node which must be replaced and ensure its functions. The complexity of the algorithm 5 is $O(2in)$, where in is the maximum number of internal nodes in a cluster.

If a clusterhead detects a failed or a moved internal node, then it consults its redundant sensors list RSL and chooses a sensor node in its affiliates or its neighboring clusters if its RSL is not empty (*we can see in Algorithm 1 that identifying redundant nodes is done just for the cluster and its direct neighboring clusters*). If the RSL of clusterhead is not empty, the clusterhead selects a redundant node in its cluster. Otherwise, it chooses a redundant node in their neighbors' Clusters. If the RSL of clusterhead is empty, then it sends a message search_red to its neighbor's clusters in order to seek a redundant node. Details are illustrated in algorithm 6 (fail-detect()). The complexity of the algorithm 6 is $O(r)$, where r is the maximum number of redundant nodes.

Algorithm 6 fail-detect()

```
If (RSL $\neq \phi$ ) Then
   If (there are redundant sensors in its cluster) then
      select from them according to the rate of energy criterion
   Else
      Send a message, containing the coordinates of the selected node, to the clusterhead which has more redundant nodes;
   End If;
Else
   send search_red message, containing the location of the failed node to its direct neighbors;
End If;
```

The Reception of the *search_red* message by a clusterhead, invokes the execution of the algorithm 7 (receipt-search-red()). This algorithm describes the reaction of a clusterhead following a searching message of a redundant node by its neighbors. The clusterhead's neighbors elect one of them that attend to do the replacement, depending on the number of its redundant nodes or the number of its internal nodes. If the elected clusterhead has redundant nodes, it selects one among them which has a high rate of energy. Otherwise, it selects one of its internal nodes which has the same criterion (high rate of energy). The complexity of this algorithm is $O(ch)$, with ch is the number of clusterheads.

Algorithm 7 receipt-search-red()

```
If (search_red received) Then
   elected ← true;
   Move by a distance of Rc/2 to the transmitter node;
   Broadcast message containing the number of the redundant nodes in RSL list or internal nodes number in its cluster;
   Repeat
      wait message;
      If (message received) Then
         If (RSL $\neq \phi$ ) and (redundant nodes number in RSL of the received message is > to its number in its RSL) Then
            elected ← false;
            back in its place;
         Else If (RSL = $\phi$ ) and (received RSL not empty or the internal nodes number in the received message is > to its internal nodes number) Then
            elected ← false;
            back in its place;
         End If;
         End If;
      End If;
   Until (sufficient time for the election);
   If (elected = true) Then
      If (RSL $\neq \phi$ ) Then
         Select the redundant node that has the most energy, and send a message containing the coordinates of the node to be replaced
      Else
         select one of its internal nodes, and send a message containing the coordinates of the node to be replaced;
      End If;
   End If;
End If;
```

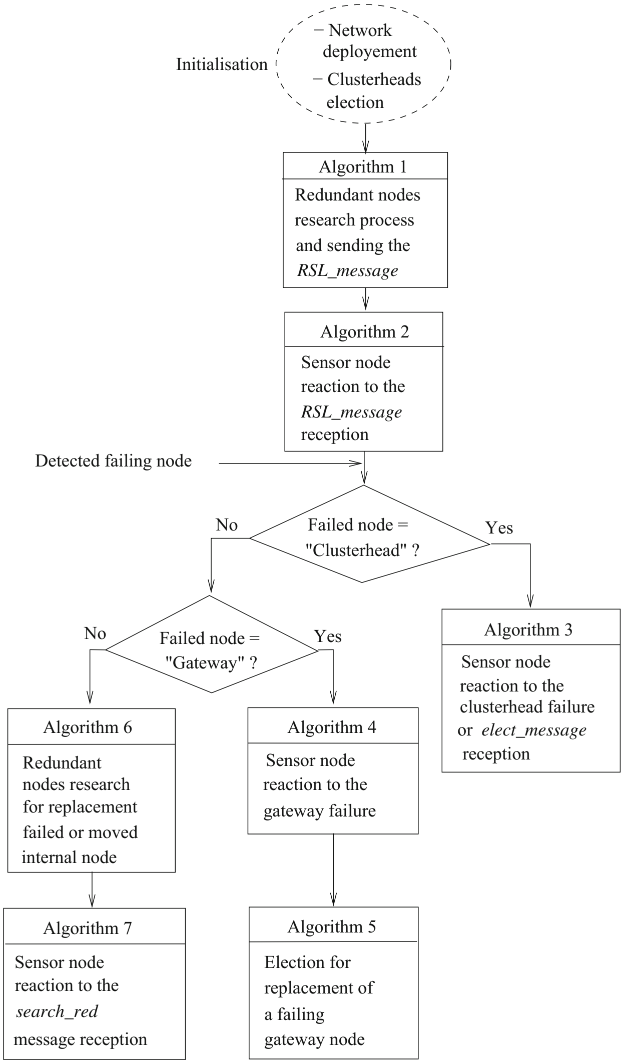


Fig. 7 Flowchart associated with the preceding algorithms

The flowchart associated with the operation of different algorithms is shown in Fig. 7. After the network deployment and the election of clusterheads, the *redundant node search process* starts by clusterheads and the results are diffused between them. If a failure sensor node is detected, depending on its type and whether it is ”*Clusterhead*”, ”*Gateway*” or ”*internal node*”, algorithms 3, 4, 5, 6 and 7 are performed.

5 Performance evaluation

We evaluate our proposed $NARF$ approach and compare it to the $DRFN$ [7] and C^3R [5] approaches, taking as metric the number of displacements, the energy consumption, and the generated messaging overhead. The simulation results are given by our simulator. In the experiments, a set of mobile sensor nodes is initially randomly deployed in a 50 m × 50 m target field. All nodes are assumed to have the same communication and sensing ranges.

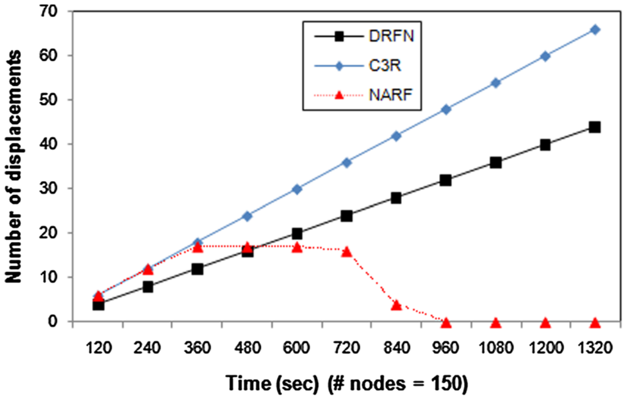


Fig. 8 Displacements number per time unit for a period of 1320 s

5.1 Number of displacements

Considering the following assumptions:

- All nodes are mobile;
- All nodes are homogeneous;
- The displacement time of a node to its neighbor is 10 s;
- The network application requires measurements each 30 s for each sensor;
- The sensors monitoring zone is assumed without obstacles.

Figure 8 shows the result of simulation according to the stated assumptions. As shown in this figure, the displacements number in our proposed $NARF$ approach exceeds the displacements number in the $DRFN$ approach for a certain period and equal to the displacements number in the C^3R approach for a certain period, then it decreases until reaching zero displacements. This implies that the consumed energy rate will be smaller comparing to the $DRFN$ and C^3R approaches. This is explained by the fact that in our $NARF$ approach, we have taken into consideration redundant nodes and that the chain replacement process finishes as soon as we reach a redundant node.

5.2 Energy consumption

The total energy consumed due to the displacement of nodes for the failure node replacement is shown in Fig. 9. It shows that the consumed energy in our approach $NARF$ is much smaller than in the $DRFN$ and C^3R approaches from a certain time. This is due to the fact that in the approach $DRFN$, replacement algorithm does not end, and in C^3R approach the replacement of the failed node is done only

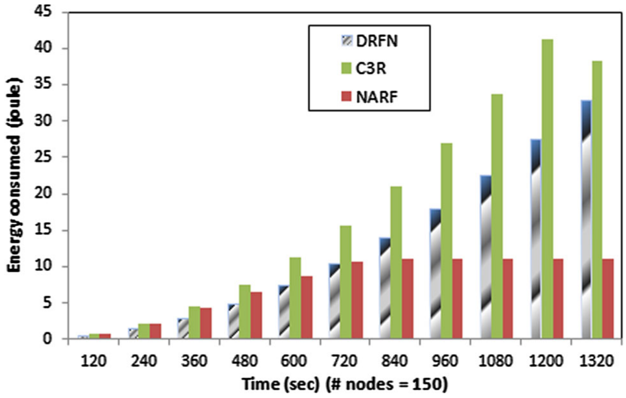


Fig. 9 Energy consumption per time unit for a period of 1320 s

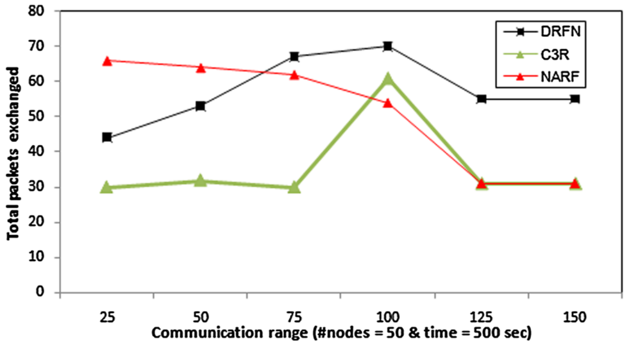


Fig. 10 Overhead evaluation

by its direct neighbors, the repeated physical movements of these nodes consume a lot of energy. However, in the new approach $NARF$, the replacement algorithm ends when the redundant node is reached.

5.3 Additional overhead

The detection of the failing node and the search of redundant nodes generate additional overhead in our approach. However, it does not degrade the global performance.

Figure 10 illustrates the generated overhead message in our $NARF$ compared to that of $DRFN$ and C^3R approaches. When the range of communication increases, the overhead decreases linearly compared to $DRFN$ and C^3R approaches.

6 Conclusion

Wireless sensors networks are generally deployed in hard and difficult access environments, where the breakdowns or failures of sensor nodes are possible. These nodes failures can harm the connectivity of the entire network. In other words, the network can be partitioned where a set of nodes can be disconnected from the total network; and consequently, the connectivity between the separate parts of this network loss. To answer this connectivity loss, we have proposed a novel approach for replacement of a failure node based on two phases, by carrying out replacements according to a distributed algorithm. The principal goal of this proposed approach is to restore the network connectivity by exploiting the sensors mobility. The idea is to share the consumption of energy needed to the connectivity restoration, with several sensors in order to minimize the early failures of the sensor nodes, and thus to prolong the lifetime of the entire network. Performance evaluations show that our $NARF$ approach consumes less energy, and generates less overhead messages when increasing the communication range, but when the communication range is small, our approach generates more messages compared to $DRFN$ or C^3R approaches.

As prospects, we envisage, in one hand, to compare our $NARF$ approach with other related approaches and with other metrics. On the other hand, to adapt to some specific environments such as underwater sensor networking constitutes a potential application, and we envisage in the future work to treat the case of heterogeneous sensor nodes. In this case, the batteries of sensor nodes have various capacities, we must initially replace $\%energy$ (percentage of energy) in Formula 1 by $energy\_rate$ (rate of energy) in measuring unit Joule, because in our work -replacement of a failing node- the important criterion is a remaining rate of energy.

Acknowledgments The authors would like to thank Dr. Fouzi Semchedine from Setif University, Algeria for his valuable comments.

References

1. Zhang, H., & Hou, J. C. (2005). Maintaining sensing coverage and connectivity in large sensor networks. *Ad Hoc and Sensor Wireless Networks Journal (AHSWN)*, *1*(1–2), 89–123.
2. Shih, E., Cho, S., Ickes, N., Min, R., Sinha, A.,Wang, A., & Chandrakasan, A. (2001). Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks. In proceedings of MobiCom'01, 7th annual international conference on mobile computing and networking, Rome.
3. Ye, F., Zhong, G., Lu, S., & Zhang, L. (2003). Peas: a robust energy conserving protocol for long-lived sensor networks. In The 23nd international conference on distributed computing systems (ICDCS), Providence.
4. Abbasi, A. A., Younis, M., & Akkaya, K. (2009). Movement-assisted connectivity restoration in wireless sensor and actor networks. *IEEE Transactions on Parallel and Distributed Systems*, *20*(9), 1366–1379.
5. Tamboli, N., & Younis, M. (2010). Coverage-aware connectivity restoration in mobile sensor networks. *Journal of Network and Computer Applications*, *33*(4), 363–374.
6. Zamanifar, A., Kashefi, O., & Sharifi, M. (2009). AOM: an efficient approach to restore actor-actor connectivity in wireless sensor and actor networks. *International Journal of Computer Networks and Communications*, *1*(1), 61–72.

7. Boudries, A. M., Aliouat, M., & Siarry, P. (2014). Detection and replacement of a failing node in the wireless sensors networks. *Computer and Electrical Engineering Journal*, *40*(2), 421–432.
8. Tian, D., & Georganas, N. D. (2002). A coverage-preserving node scheduling scheme for large wireless sensor networks. *A first ACM international workshop on wireless sensor networks and applications* (pp. 32–41), Georgia.
9. Boudries, A. M., & Aliouat, M. (2012). Connectivity maintenance in the wireless sensors networks: Detection and replacement of a failing node. In the 2nd international congress on telecommunication and application, ICTA'2012. Bejaia, pp. 11–12.
10. Ahvar, Ehsan, Lee, Gyu Myoung, Crespi, Noel, & Ahvar, Shohreh. (2016). RER: a real time energy efficient routing protocol for query-based applications in wireless sensor networks. *Journal of Telecommunication Systems*, *61*(1), 107–121.
11. Hussain, Sajid, Azim, Anwarul, & Park, Jong Hyuk. (2009). Energy efficient virtual MIMO communication for wireless sensor networks. *Journal of Telecommunication Systems*, *42*(1), 139–149.
12. Kariman-Khorasani, Mohsen, & Pourmina, Mohammad Ali. (2015). Maximum lifetime routing problem in asynchronous duty-cycled wireless sensor networks. *Journal of Wireless Networks*, *21*(8), 2501–2517.
13. Ali, Samia A., & Refaay, Shreen K. (2012). Chain based fault tolerant routing protocols. *Journal of Network Protocols and Algorithms*, *4*(3), 79–103.
14. Ali, Samia A., Shaaban, Khaled M., & Alkabbany, Islam M. (2012). Distributed patching for mobile wireless sensor networks. *Journal of Network and Computer Applications*, *35*(5), 1598–1605.
15. Kang, Z., Yu, H., & Xiong, Q. (2013). Detection and recovery of coverage holes in wireless sensor networks. *Journal of Networks*, *8*(4), 822–828.
16. Frye, L., Liang, C., Du, S., & Bigrigg, M.W. (2006). Topology maintenance of wireless sensor networks in node failure-prone environments. In Proceedings of IEEE international conference on networking, sensing and control, pp. 886-891.
17. Lin, T.-Y., Santoso, H. A., & Wu, K.-R. (2015). Global sensor deployment and local coverage-aware recovery schemes for smart environments. *IEEE Transactions Mobile Computing*, *14*(7), 1382–1396.
18. Cheng, C.-F., & Huang, C.-W. (2015). An energy-balanced and timely self-relocation algorithm for grid-based mobile WSNs. *IEEE Sensors Journal*, *15*(8), 4184–4193.
19. Ramesh, K., & Somasundaram, K. (2011). A comparative study of clusterhead selection algorithms in Wireless Sensor Networks. *International Journal of Computer Science & Engineering Survey (IJCSES).*, *2*(4), 153–164.
20. Venkata Subbaiah, K., & Naidu, M. M. (2010). Cluster head Election for CGSR Routing Protocol Using Fuzzy Logic Controller for Mobile Ad Hoc Network. *International Journal of Advanced Networking and Applications*, *1*(4), 246–251.
21. Nocetti, Fabian Garcia, Gonzalez, Julio Solano, & Stojmenovic, Ivan. (2003). Connectivity Based k-Hop Clustering in Wireless Networks. *Journal of Telecommunication Systems*, *22*(1), 205–220.
22. Nguyen, D., Minet, P., Kunz, T. & Lamont, L. (2011). New findings on the complexity of cluster head selection algorithms. In IEEE international symposium on world of wireless, mobile and multimedia networks. pp. 1–10, ISBN: 978-1-4577-0352-2 (2011).
23. Hussain, K., Abdullah, A. H., Awan, K. M., Ahsan, F., & Hussain, A. (2013). Cluster head election schemes for WSN and MANET: a survey. *World Applied Sciences Journal*, *23*(5), 611–620.
24. Heinzelman, W., Chandrakasan, A., & Balakrishnan, H. (2000). Energy-efficient communication protocol for wireless microsensor networks, In Proceedings of 33rd HICS, vol 2, pp. 1–10 (2000).

Abdelmalek Boudries is an associate professor at Bejaia University. He is member of LMA Laboratory (Laboratory of Applied Mathematics). He received the Engineer degree from the University of Tizi-ouzou, Algeria, in 2000, and the Magister degree from the University of Bejaia, Algeria, in 2007. Since June 2015, he is a PhD at the University of Bejaia. His main research interests ad hoc and wireless sensor networks (*Routing, connectivity maintenance*,...).

Mourad Amad is an associate professor at Bejaia University. He is member of LaMOS Research Unit (*Modelling and Optimisation of Systems*). He received the engineer degree from the National Institute of Computer Science (*INI-Algeria*) in 2003 and the magister degree from the University of Bejaia (*Algeria*) in 2005. Since January 2012, he is a PhD at the University of Bejaia, In may 2016, he obtained the HDR in computer sciences from Bejaia University. His research interests include: Peer to peer networks (*Lookup acceleration, Security, Performance evaluation, Mathematiques Modelling*, ...), Sensor Networks (*Clusterisation, Routing*, ...), Media Conferencing (*architecture, ALM, signalling protocol, P2P-SIP*, ...), Social Networks, Internet of Things (*Architecture, Fault Tolerance*, ...).

Patrick Siarry was born in France in 1952. He received the Ph.D. degree from the University Paris 6, in 1986 and the Doctorate of Sciences (Habilitation) from the University Paris 11, in 1994. He was first involved in the development of analog and digital models of nuclear power plants at Electricité de France (E.D.F.). Since 1995 he is a professor in automatics and informatics. His main research interests are computer-aided design of electronic circuits, and the applications of new stochastic global optimization heuristics to various engineering fields. He is also interested in the fitting of process models to experimental data, the learning of fuzzy rule bases, and of neural networks.