Computer Networks 127 (2017) 282–295

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Distributed connectivity restoration in multichannel wireless sensor networks

Samira Chouikhi[a,b,*], Inès El Korbi[a], Yacine Ghamri-Doudane[c], Leila Azouz Saidane[a]

[a] National School of Computer Science, CRISTAL Lab, University of Manouba, 2010 Tunisia
[b] Université Paris-Est, LIGM Lab, 5 Bd. Descrates, 75420 Champs sur Marne, France
[c] L3i Lab, University of La Rochelle, Av. Michel Crépeau, 17042, La Rochelle CEDEX 1, France

ARTICLE INFO

Article history:
Received 31 August 2016
Revised 15 June 2017
Accepted 14 August 2017
Available online 17 August 2017

Keywords:
Articulation node failure
Connectivity restoration
Failure recovery
Fault tolerance
Wireless sensor networks

ABSTRACT

Wireless sensor networks (WSNs) are widely used in various domains. However, the specificity of the nodes deployed in these networks makes them prone to failures. To overcome this problem and guarantee the continuity of the network functioning in the presence of node failures, fault tolerance mechanisms need to be designed and integrated to ensure the correct WSN operation. In addition to node failure, the interferences present a serious problem in WSNs. Such a problem is commonly solved by using multichannel communications. Thus, in this paper, we propose a distributed solution, called Connectivity Restoration for Multi-Channel WSNs (CR-MC), to recover from a connectivity loss for multichannel WSNs. The main task of this approach targets the restoration of the connectivity and the reassignment of the channels in a multichannel network after the failure of an articulation node whose failure leads to the network partitioning. CR-MC uses only the neighborhood information to execute the recovery and the channel reassignment tasks. On the other hand, if we consider multi-hop WSNs, a routing tree is generally constructed to disseminate the information to the sink node. Hence, the radio channels should be assigned with a great care to respect these network particularities. In this context, we propose a second solution, called Connectivity Restoration for Routing based Multi-Channel WSNs (CR-RMC) that exploits the routing tree as well as the vicinity information while allocating the channels. We compare the performance of the two proposed approaches CR-MC and CR-RMC by evaluating them through simulations.

1. Introduction

In the recent years, the deployment of wireless sensor networks (WSNs) [1] have won a great notoriety in different domains going from simple data collection to critical system monitoring and control. More and more researchers have dedicated their work to improve the reliability and efficiency of WSNs regarding the requirements of different applications. However, the characteristics of sensor nodes (limited CPU, limited memory, limited battery, wireless communications, etc.) and the harsh environment, in which they are deployed, negatively affect the reliability of WSNs and even lead to the network failure. On one hand, a node failure, caused by energy depletion or any other problem, may partition the network into disjoint parts; where some of these parts get disconnected from the sink. On the other hand, the collisions/retransmissions phenomena caused by interferences between senor nodes, will demand extra energy consumption, thus causing the premature failure of some nodes.

The problem of the interference ratio minimization has been widely discussed in literature, and the common technique used to handle such a problem is to use multichannel communications. Since the integration of multi-channel radios in wireless sensor devices, many interesting researches have been dedicated to optimally exploit the benefits of multi-channel communications in WSNs [2–7]. The proposed mechanisms minimize the interference ratio and hence the collision/retransmission ratio. However, the probability of node failure remains high. Thus any node may fail due to energy depletion or any other factor. In many applications, the WSNs are deployed in unreachable or dangerous environments. Therefore, a node battery exchange is unfeasible. Moreover, the replacement of a failed node by a new node is impossible. The problem becomes even more critical when the failed node is an articulation node (articulation point AP) -a particular node whose deficiency leads to a connectivity loss- and hence, the network is partitioned into isolated segments as depicted by Fig. 1. To overcome this problem, the WSN needs to be reorganized to restore the connectivity. The main idea in the network reorganization is

* Corresponding author.
E-mail addresses: samira.chouikhi@univ-paris-est.fr (S. Chouikhi), ines.korbi@ensi.rnu.tn (I. El Korbi), yacine.ghamri@univ-lr.fr (Y. Ghamri-Doudane), leila.saidane@ensi.rnu.tn (L. Azouz Saidane).

http://dx.doi.org/10.1016/j.comnet.2017.08.016
1389-1286/

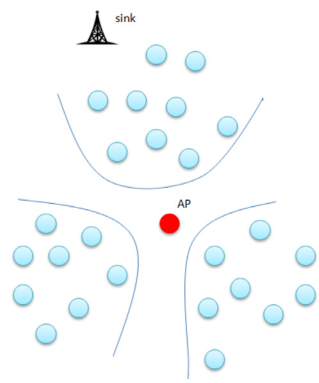


Fig. 1. WSN partition after an articulation node failure.

to relocate some nodes around the failed node with minimum impact on the network initial topology. In this paper, we focus on fault tolerance mechanisms in WSNs while considering the multichannel communications. The main multichannel issue discussed here is the channel allocation when the number of channels is limited. We opt for a strategy based on the routing trees to better exploit the available channels. As a second focus of our work, we propose two mechanisms for fault tolerance: a preventive mechanism and a curative mechanism. The preventive mechanism uses an activity/sleeping technique to minimize the energy consumption, while the curative mechanism relocates some nodes to restore the WSN connectivity after an articulation node failure. This failure can be caused by energy depletion or a crash. The channel allocation and the failure recovery rely on the Graph theory and use graph coloring [8] heuristics for channel (re)allocation and *Steiner Points* [9] heuristic to rearrange the nodes.

The rest of this paper is organized as follows: Section 2 overviews the work related to the topics of multichannel communication and fault recovery. Section 3 describes the problems of multi-channel allocation and articulation node failure recovery. The proposed solution is described in Section 4 and evaluated in Section 5. Section 6 concludes the paper.

2. Related work and contributions

2.1. Related work

The connectivity is one of the most crucial requirements in WSNs. In literature, many interesting works have been dedicated to ensure, improve or restore connectivity in this type of networks [10]. The solutions proposed in the literature can be classified into two main classes: preventive solutions and curative solutions. The preventive solutions aim to prevent the network division into disconnected segments. Generally, these solutions are based on *energy consumption management* (multichannel communications, sensor activity/sleeping schedule, etc.), *deployment strategies* (redundant nodes, relay nodes,etc.) and *multi-path routing* techniques. The curative class includes all the solutions triggered after a failure occurrence. These solutions use different techniques to resume the correct network functioning. The main techniques used are *new nodes' deployment* and *network reorganization*. These techniques are commonly used to improve or to restore the network connectivity to prevent or to recover from network segmentation. The techniques of network reorganization are based on the redeployment of some nodes in the vicinity of the failure to fully/partially restore the network connectivity. It thus assumes that some or all the nodes are movable [11–14].

Many proposed mechanisms use node reorganization techniques to enhance the WSN reliability. In [11], Younis et al. proposed CRR (*Connectivity Restoration through Rearrangement*) which recovers from the connectivity loss by moving some nodes to new locations in the failed node vicinity. CRR defines the new locations as the *Steiner Points* of a *Steiner Tree* connecting the different network segments. The algorithm presented in [13] replaces the failed node by its neighbors one by one. Each node moves back and forth to the failed location for a period of time. In [14], the authors proposed the RIM (Recovery through Inward Motion) approach which performs a local recovery around the failed node. The main idea of RIM is to move the neighbors towards the failed node position until they become connected (i.e. they can directly communicate with each other).. If, at any time, a node becomes disconnected from one of its neighbors because of its relocation, the disconnected node follows this neighbor to restore their lost connection. The relocation procedure is triggered recursively to avoid any node disconnection. However, RIM is a very costly solution in dense networks in terms of number of relocated nodes and traveled distance. In [15], the approach CSDS (a Critical Sensor Determination and Substitution) was proposed for mobile sensor networks. The strategy used by this approach allows identifying the critical nodes and selecting a backup sensor for each of them. The selection of the best candidates as backup nodes is based on the criticality, the hop-count and the Euclidean distances to their neighbors. The main advantage of CSDS is that only one node will be relocated in the case of an articulation node failure.

The LeDiR (*Least-Disruptive topology Repair*) algorithm, proposed in [16] for Wireless sensor-actor networks (WSAN), restores the connectivity between actors (nodes designed to take some actions collaboratively to achieve predefined application tasks). This algorithm relocates a minimum number of nodes without extending the length of the shortest inter-actor path. In LeDiR, the recovery process is restricted to the neighbors belonging to the smallest disjoint network segment after the WSAN partitioning. In [17], Imran et *al.* proposed DCR (Distributed partitioning detection and Connectivity Restoration algorithm) that detects the critical nodes in WSANs and selects a backup non critical node for each articulation one based on *Guardian nomination* principle. An extended version, named RAM, of DCR has been proposed to deal with multi-node failures. In [18], the authors proposed the algorithm DPCRA (Distributed Prioritized Connectivity Restoration Algorithm) that identifies the articulation nodes in a proactive way and elects some Failure Handlers (FHs) to recover from a possible failure. The FHs detect the division of the network after an articulation node failure and the first designed FH tries to handle this failure and restore the connectivity. If this FR is unable to start the connectivity process within a permissible reaction time, the next designed FH starts the recovery process. Moreover, DPCRA adopts a cascaded relocation to handle the connectivity restoration. In [19], Haider et al. proposed the algorithm RACE (Restore Actor Connected Coverage) that recovers from the failure of critical and non-critical actors. If the failed actor is non-critical, its neighbors move to cover the bereaved sensors (uncovered after their actor failure). In the case of the critical actor failure, its neighbors independently determine the best candidate to replace the failed actor. This selection is based on the criticality of the candidate (to avoid a cascaded relocation) and the impact of this candidate relocation on the number of uncovered sensors. Thus a non-critical actor which covers the small number of sensors, uncovered by other actors, will be preferred to substitute the failed actor.

In addition to the mechanisms proposed for the recovery from a single articulation node failure, many interesting algorithms discussed the connectivity restoration after simultaneous failures of multiple nodes. The algorithm DORMS (Distributed algorithm for Optimized Relay node placement using Minimum Steiner tree), proposed in [12], uses *Steiner Trees* to reconnect the WSN after the failure of multiple nodes. DORMS forms a bridge between each two disconnected network segments using mobile relay nodes. In [20], the authors proposed CRAFT (Connectivity Restoration with Assured Fault Tolerance) which establishes a bi-connected inter-partition topology by deploying a minimal number of relay nodes. CRAFT tries to form the largest backbone empty polygon around the center of the failed zone which means that no partition exists inside this polygon. Then, the algorithm determines two disjoint paths for each partition, using the deployed relay nodes, to connect it to the backbone polygon. Another algorithm, called RIR (Recovery algorithm with Increased Robustness), proposed in [21], targets the failure of multiple critical nodes. RIR replaces the failed nodes with non-critical healthy nodes having the highest residual energy in order to maintain the network connectivity as long as possible. Moreover, to minimize the distance traveled by the substituting nodes, RIR considers the problem of failure recovery as a Minimum Cost Flow (MCF) problem. The failed nodes are considered as the destinations in the MCF model while the healthy nodes are considered as source nodes. The MCF model determines the solution generating the minimal cost, where the cost function denotes the total traveled distance.

All the mentioned algorithms recover from node failures and offer mechanisms to restore the connectivity in WSNs. Unfortunately, they did not consider the multichannel communications which are commonly used in WSNs to avoid interferences during the data transmission. Thus, if we consider the multichannel context, the presented algorithms are inapplicable without adding a channel management mechanism to adapt them to this context. The consideration of multichannel communications, when proposing a solution for the connectivity restoration, will considerably simplify and favor the application of such an algorithm in interference prone WSNs. In [22], we proposed a centralized solution for the failure recovery of an articulation node in a multichannel WSN. The proposed approach tries to reorganize a minimum set of nodes around the failed node to restore the network connectivity regarding the multichannel context. As the proposed solution is centralized, we only described the failure recovery mechanisms as they are executed by the sink. In [23], we propose the RNFR Rotating Nodes based Failure Recovery approach that extends the solution proposed in [22] to consider the case of the network partitioning after the articulation node failure. For instance, the articulation node's failure may preclude some part of the network from receiving the recovery information sent by the sink. The main idea of the proposed approach RNFR (*Rotating Nodes based Failure Recovery*) is to select a set of nodes to take in charge the dissemination of the recovery information in the whole network.

2.2. Contributions

In this paper, we propose two distributed solutions to recover from a multichannel WSN partitioning after an articulation node failure. The main contributions of this work are summarized as follows:

- We formulate the problem of the network connectivity restoration in a multichannel context as a Multi-objective Optimization Problem (MOP). This formulation allows minimizing the impact of the failure recovery on the WSN configuration in terms of topology and initial channel assignment while restoring the connectivity.
- We propose a native distributed approach, named Connectivity Restoration for Multi-Channel WSNs (CR-MC), for the recovery from an articulation node failure. This approach uses only the neighborhood information and does not consider the routing information when executing the fault tolerance procedure. It integrates some heuristics from graph theory to perform a localized recovery around the failed node. These heuristics rearrange some neighbors to reconnect the disjoint parts of the WSN and then to reallocate the channels, hence ensuring an optimal channel assignment. CR-MC is executed in a distributed manner by the failed articulation node neighbors.
- Then, we propose the distributed approach Connectivity Restoration for Routing based Multi-Channel WSNs (CR-RMC). This approach uses the same heuristics used by CR-MC for the connectivity restoration. However, CR-RMC also considers the routing trees when assigning the available channels to the WSN nodes.
- We evaluate the performance of the two proposed approaches by simulation. Moreover, we compare them with the existing approach RNFR [23].

The two approaches CR-MC and CR-RMC will be described in detail in Sections 3.2 and 4.2 respectively.

3. CR-MC approach

3.1. Problem formulation

In this Section, we focus on the formulation of the connectivity restoration problem in a multichannel WSN. The multichannel communication is used to reduce the interference ratio between all the sensor nodes in the network. Thus, the initial channel allocation must minimize the number of interferences between the nodes. Besides, if an articulation node failure arises in the WSN, its recovery should restore the connectivity while minimizing the network interferences.

We use the graph theory paradigm to formulate our problem. Thus, we consider the problem of channel allocation/reallocation as a graph coloring problem and the problem of connectivity restoration as a sub-graphs' connection problem. For the latter problem, we will exploit a technique based on Steiner Points [9] used to reconnect sub-graphs with a minimal cost.

Our approach *Connectivity Restoration for Multi-Channel WSNs* (CR-MC) aims to restrict the failure recovery process in the small vicinity of the damaged articulation node hence to limit the changes around the considered node and not in the whole network. Our fault recovery problem objectives can be summarized as follows: i) to minimize the number of interfering links over each node, ii) to reduce the number of relocated nodes, iii) to limit the number of nodes changing their channels when being relocated, , and iv) to minimize the distance traveled by the relocated nodes. Such a problem can be formulated as a constrained multi-objective optimization problem (MOP) with four objectives to be optimized.

As a first step of our formulation, we model the WSN as a colored graph $G_0=(V_0,E_0)$, with V_0 the set of vertices representing the movable sensor nodes that can be relocated to ensure the failure recovery ($|V_0|=n$). E_0 represents the set of edges between the vertices. An edge e, $(u,\ v)\in E_0$, exists if and only if the two nodes interfere (i.e. the distance between nodes u and v $d(u,\ v)\leq 2R_c$, with R_c the communication range). We use m colors to color the vertices where each color represents a channel.

After the failure of an articulation node, we reorganize the nodes to reconnect the network. Let $G_0^{'}=(V_0^{'},E_0^{'})$ be the new colored graph representing the WSN after the failure recovery, where $V_0^{'}=V_0-n_f$ (n_f is the failed node). We define the color of a vertex v in a graph G as a function $c_G : R\longmapsto M$, where $R\subset V-s$ is the set of receivers. We also define a conflicting link as a link where the nodes on its both sides use the same channel. To formulate the first objective, we check if a link is a conflicting link by defining the binary function $\delta : V_0^{'}\times V_0^{'}\longmapsto [0,1]$ as:

$$\delta(v,u)=\begin{cases}1 & \text{if } (v,u)\in E_0^{'}\wedge c_{G_0^{'}}(v)=c_{G_0^{'}}(u)\\ 0 & \text{otherwise.}\end{cases} \tag{1}$$

The number of a node v conflicting links is determined by summing the conflicting links between the node v and all its neighbors:

$$f_1^v = \sum_{u \in V_0'} \delta(v, u). \quad (2)$$

Now, we can define the objective function f_1 as a vector of functions:

$$f_1 = [f_1^v], \forall v \in V_0' \quad (3)$$

To formulate the second objective, we define the function $pos_G : V' \longmapsto \mathbb{R} \times \mathbb{R}$ as the position function that returns the coordinates of a vertex in a graph G. Therefore, to determine if a node is relocated or not, we define $\varphi(v)$ as:

$$\varphi(v) = \begin{cases} 1 & \text{if} \quad pos_{G_0'}(v) \neq pos_{G_0}(v) \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

Hence, the total number of relocated nodes is determined by:

$$f_2 = \sum_{v \in V_0'} \varphi(v). \quad (5)$$

In the same way, we define $\gamma(v) : V_0' \times V_0' \longmapsto [0, 1]$ the function that indicates if a node v is re-colored (a new channel is assigned to v) or not:

$$\gamma(v) = \begin{cases} 1 & \text{if} \quad c_{G_0'}(v) \neq c_{G_0}(v) \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

The total number of re-colored nodes is defined as:

$$f_3 = \sum_{v \in V_0'} \gamma(v). \quad (7)$$

The last objective in our problem is to minimize the distance traveled by the relocated nodes. Let $pos_{G_0}(v) = (x, y)$ and $pos_{G_0'}(v) = (x', y')$ the positions of the relocated node v before and after the network reorganization, respectively. The distance traveled by this node is:

$$dist(v) = \sqrt{(x' - x)^2 + (y' - y)^2} \quad (8)$$

The fourth function is defined as the total distance traveled by all the relocated nodes:

$$f_4 = \sum_{v \in V_0'} dist(v)\varphi(v) \quad (9)$$

After defining the four functions, a vector function f can be defined as:

$$f = [f_1, f_2, f_3, f_4]^T \quad (10)$$

Therefore, our multi-objective problem becomes:

$$minimize \ f \quad (11)$$

s.t.

$$G_0' \text{ is connected} \quad (12)$$

$$1 \leq c_{G_0'}(v) \leq m, \forall v \in V_0' \quad (13)$$

where the constraint (12) ensures the WSN connectivity and the constraint (13) guarantees that each node is assigned a channel from the list of available channels.

Many methods are proposed in the literature to give the exact solutions of multi-objective optimization problems. Unfortunately, these methods cannot be applied in the WSN context due to their exponential execution time and their complexity (NP-hard). The exact solving of this problem cannot be applied in real WSNs because of the limited sensor nodes resources and the size of the network that can reach hundreds and even thousands of nodes. Hence, we propose the Connectivity Restoration for Multi-Channel WSNs approach (CR-MC) which is based on heuristics to solve this problem in a distributed manner and at a polynomial time.

3.2. CR-MC approach's steps

We propose the distributed approach Connectivity Restoration for Multi-Channel WSNs (CR-MC) for the multi-objective problem described in the previous Section.

3.2.1. Failure detection

In CR-MC, a node failure is detected by its 1-hop neighbors. For instance, periodically, each node sends a *heartbeat* message to check the state of its neighbors. The network reorganization algorithm is performed by all the articulation node's 1-hop neighbors.

3.2.2. Node relocation

To reorganize the WSN, we move some nodes to new locations in a way to restore the network connectivity. We use the same technique of Steiner Points (SPs) used in [11] to find the new locations and then move some 1-hop and may be 2-hop neighbors to these locations.

To find the SPs (SP_set), we use a heuristic that limits the reorganization process to the failed node vicinity (1-hop and 2-hop neighbors). We first form a polygon with all the failed node's 2-hop neighbors as vertices. The SPs are determined as follows:

1. For each two nodes i and j of the polygon, a triangle Δ_{ij} is formed with the failed articulation node n_f as its third vertex.
2. for each Δ_{ij}, the SP is the point inside Δ_{ij} that minimizes the distance $d = dist(SP, i) + dist(SP, j) + dist(SP, n_f)$. To find this SP, the proposed approach uses the approximation algorithm *k-LCA* (k-restricted Loss-Contracting Algorithm) [24], with k set to 3.
3. Among all the determined SPs, we remove the SPs covering less than two vertices of the polygon (i.e. $\forall i$ vertex, $dist(i, SP) > R_c$) and the redundant SPs.[1] The remaining SPs are put in SP_set.
4. If some vertices of the polygon (2-hop neighbors) remain uncovered by the SPs, we form a new polygon with these vertices and the SPs of SP_set. We apply the steps 1–3 to find the new SPs.
5. The process is repeated until all the 2-hop neighbors are covered by the SPs.

The second step consists in relocating the failed node's 1-hop and 2-hop neighbors to the selected SPs. We define H_1 and H_2 as the sets of 1-hop and 2-hop neighbors of the failed node. Two cases can take place:

- **Case 1**: $|SP\_set| \leq |H_1|$: each SP y will be occupied by the nearest 1-hop node $u \in H_1$ such as:

$$dist(y, u) = \max_{x \in SP\_set} (\min_{v \in H_1} dist(x, v)). \quad (14)$$

  The process is repeated until all the SPs are replaced by the failed node's 1-hop neighbors.
- **Case 2**: $|SP\_set| > |H_1|$: all the 1-hop neighbors will move to SPs locations, as well as some of the 2-hop neighbors such as the number of the relocated 2-hop nodes is equal to $|SP\_set| - |H1|$. The relocated 2-hop nodes will be replaced using the DARA protocol [15], which keeps the same network topology.

Fig. 2 depicts the steps of the network reorganization, where the green nodes represent the 1-hop neighbors, the blue nodes

[1] A redundant SP covers a vertex covered by another SP

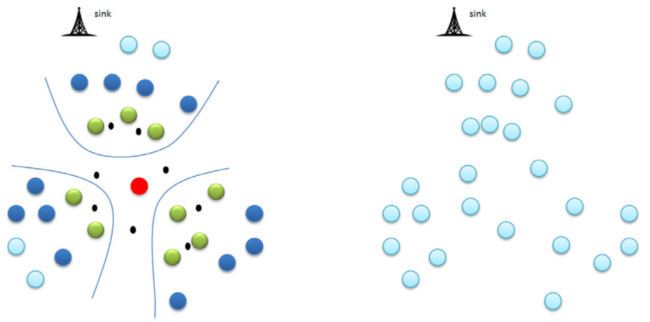


Fig. 2. The WSN reorganization after the failure.

represent the 2-hop neighbors and the black points represent Steiner points (SPs).

It is to mention that the whole recovery process depends on the node mobility. As we said in previous Sections, we assume that all the nodes are movable, while it is not the case in many networks. However, our solution remains valid for WSNs with partial nodes' mobility. In this case, in the SPs' replacement we only use the movable nodes (whatever the number of hops towards the failed node is).

3.3. Channel reallocation

After termination of the relocation process, the neighborhood tables and the number of conflicting links are updated. Then, the process of channel reallocation is triggered for each relocated node. This process tries to assign a channel to the node while minimizing the number of its conflicting links affecting the whole network configuration. As CR-MC is a distributed approach, each node is responsible of choosing its new channel. The process is based on negotiation between the neighbors to take the best decision. The relocated nodes perform the channel reallocation as follows:

- The relocated node v broadcasts a *hello* message.
- Each neighbor u of v, replies with its ID, its channel number, the number of conflicts (conflicting links) generated by this channel and if it is relocated or not (using a reserved flag in the *hello* message).
- The node v checks the *hello* messages received from the relocated neighbors.
- If v has the highest ID among the relocated nodes, it starts the reallocation process as follows:
  1. If a channel is still free, v chooses it as its new channel.
  2. Else v calculates the number of conflicts for each channel and broadcasts a *conflicts* message to its non relocated neighbors.
  3. Each neighbor verifies if it can reduce the number of conflicting links when it changes its channel. If it is the case, it sends a *suggestion* message to v with its ID, its channel number and the new number of conflicting links.
  4. v chooses the suggestion which reduces the number of conflicts as much as possible.
  5. If no suggestion was received, v chooses the channel which generates the smallest number of conflicts.
  6. v broadcasts a *decision* message to inform its neighbors of its new channel.
  7. If a neighbor sent a suggestion and its channel is chosen, it switches to a new channel.
- Else, it receives the decision messages from all its relocated neighbors having IDs higher than its own ID. Then, go to 1.

The pseudo-code of the recoloring procedure is given in Algorithm 1

Algorithm 1 Re-coloring heuristic.

$ch = 0$
send(msg_hello,own-id)
receive(msg_hello, id, c, nb_cf(c))
if own-id=max(id) **then**
  1:
  if $\exists i \in M | C_i = \emptyset$ **then**
    $ch = i$
  else
    for each $i \in M$
    calculate nb-cf(i)
    send (msg_conflicts)
    receive(msg_suggestion, id, c,)
    $ch = best\_suggestion$
    if no suggestion received **then**
      $ch = argmin_{i \in M} nb\_cfi$
    end if
  end if
  send(msg,decision)
else
  receive(msg,decision, neighbors). Then, go to 1.
end if

Even though CR-MC ensures the connectivity restoration and the channel reallocation without disturbing the whole WSN functioning, it presents some drawbacks. In fact, it reallocates the channels blindly which means that it did not consider the routing trees while assigning the channels. Therefore, CR-MC can assign different channels to siblings (nodes having the same parent) while assigning the same channel to neighbors belonging to different parents. However, we know that two siblings cannot transmit simultaneously even if they use different channels. Hence, we propose a new approach CR-RMC (Connectivity Restoration for Routing based Multi-Channel WSNs) that uses the routing information while allocating the channels.

4. CR-RMC approach

In this section, we propose the CR-RMC approach that considers the routing information when assigning the channels to the sensor nodes (initial allocation and reallocation during the failure recovery). The use of routing trees will allow a best exploitation of the available channels by allocating the same channel to all the children of the same parent. For instance, if two nodes send their data to the same parent v, they must switch their radios to the receiving node v's channel. Moreover, this mechanism will minimize the number of nodes switching their radios while transmitting data along the routes from the sources to the sink.

4.1. Problem statement and formulation

Considering the same WSN as in CR-MC, we divide it into layers according to the number of hops toward the sink. Then, a new WSN model needs to be defined where a communication link $e_c(u, v)$ is the link between the node u and its parent v in the routing tree. An interference link $e(u, v)$ indicates that the transmissions of the two nodes v and u interfere. Since a sensor node v can receive data from only one of its descendants at once, the information can either transmit over the link $e_c(u, v)$ or $e_c(w, v)$, but not over the two links at the same time even if u and w use two different channels. Therefore, we will not consider the links between the children of the same parent as conflicting links. We then introduce the concept of Logical Node (LN) as the set of nodes having the same parent ($N_i = \{u | p(u) = i\}$, where $p(u)$ is the parent of u).

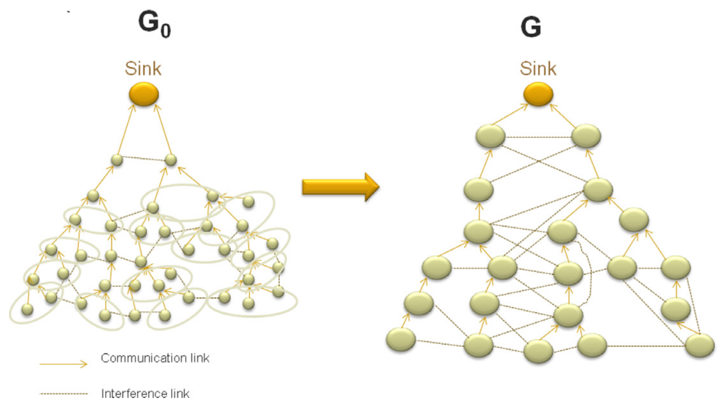


Fig. 3. The graph of logical nodes.

Two LNs, N_i and N_j, are interfering (an interference link exists between the two nodes) in two cases:

- If $\exists u \in N_i$ in the interference range of a node $v \in N_j$.
- If $\exists u \in N_i$ in the interference range of j or vice versa.

We define $c(N_i, N_j)$ as the cost of the interference link of a logical node N_i to the node N_j. We first introduce the function α, that checks if two sensor nodes are interfering, by:

$$\alpha(u,v) = \begin{cases} 1 & \text{if} \quad e(u,v) \in E_0 \\ 0 & \quad\text{otherwise.} \end{cases} \tag{15}$$

If two sensor nodes u and j interfere, u cannot transmit data simultaneously with any descendant of j. Thus, we define the cost of a sensor u's interference link to a logical node N_j as:

$$c(u, N_j) = \alpha(u, j)|N_j| \tag{16}$$

The total cost of a logical node N_i's interference link to another logical node N_j includes all the links between N_i and j (parent of all N_j descendants). Moreover, we add the interference links between the sensor belonging to N_i, not interfering with j, and the sensors belonging to N_j. This cost is then determined by:

$$c(N_i, N_j) = \sum_{u \in N_i} c(u, N_j) + \sum_{w \in N_i \wedge w \neq u, v \in N_j} \alpha(w, v) \tag{17}$$

We extract the modified colored graph $G = (V, E)$ from the graph G_0, where V represents the set of LNs and E the edges' set that represents the interference links between the LNs. Fig. 3 depicts how the original graph G_0 is transformed to obtain G, the graph of logical nodes. The solid lines denote the communication links, while dashed lines represent the interference links. Unlike CR-MC, CR-RMC aims to minimize the number of conflicting links for each logical node instead of minimizing each sensor node 's conflicting links. Then, the Eq. (1) is redefined by:

$$\delta(N_i, N_j) = \begin{cases} c(N_i, N_j) & \text{if} \quad e(N_i, N_j) \in E' \wedge c_{G'}(i) = c_G(j) \\ 0 & \quad\text{otherwise.} \end{cases} \tag{18}$$

In the same way, the Eq. (2) becomes:

$$f_1^{N_i} = \sum_{\forall N_j \in V} \delta(N_i, N_j) \tag{19}$$

The objective function f_1, that represents the number of conflicts for each logical node, can be written as:

$$f_1 = \left[f_1^{N_i}\right], \forall i \in R \text{ (the set of receivers)} \tag{20}$$

The objective functions f_2, f_3 and f_4 are the same as described in Eqs. (5), (7) and (9).

Moreover, for fault recovery purposes, the relocation of sensor nodes may increase the number of hops to the sink which may incur in channel reassignment for all the relocated nodes' sub-trees. Hence, to minimize the topological and configuration changes upon the failure occurrence, we define the function f_5, that calculates the number of nodes belonging to a new layer after the failure recovery, as:

$$f_5 = \sum_{\forall v \in V - n_f} \lambda(v) \tag{21}$$

where

$$\lambda(v) = \begin{cases} 1 & \text{if} \quad h_G(v) \neq h_{G'}(v) \\ 0 & \quad\text{otherwise.} \end{cases} \tag{22}$$

Finally, the function f_6 given by Eq. (24) determines the number of nodes switching their radios to transmit data to their parents. Thus, we define the binary function ω, that checks if a node switch its radio or not, by:

$$\omega(v) = \begin{cases} 1 & \text{if} \quad c_{G'}(v) = c_G(p(v)) \\ 0 & \quad\text{otherwise.} \end{cases} \tag{23}$$

The total number of the nodes assigned different channels from their parents is given by:

$$f_6 = \sum_{v \in V_0'} \omega(v) \tag{24}$$

Hence, the multi-objective optimization problem becomes:

$$\textit{minimize} \;\; f = [f_i]_{i=1..6} \tag{25}$$

s.t.

$$G' \text{ is connected} \tag{26}$$

$$1 \leq c_{G'}(v) \leq m, \forall v \in V' \tag{27}$$

In the next Section, we propose the CR-RMC (Connectivity Restoration for Routing based Multi-Channel WSNs) approach to solve this problem.

4.2. CR-RMC approach's steps

In this Section, we propose a distributed fault tolerance approach for routing based multichannel WSNs.Our approach includes both preventive and curative mechanisms to ensure the fault tolerance in these networks. The preventive mechanisms, using a sleep/awake nodes' strategy, aim to avoid the fault occurrence as long as possible by increasing the whole network lifetime. The curative mechanisms are related to the network recovery after an articulation node failure. This second category of mechanisms represents the solution to the multi-objective problem stated in Section 4.1.

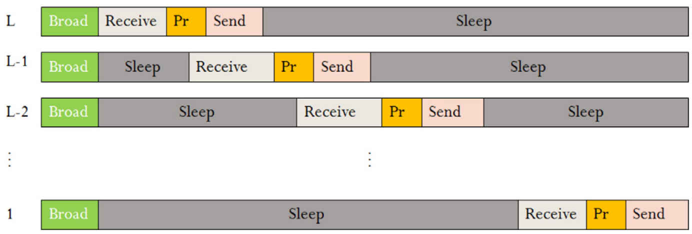


Fig. 4. The sleep/awake frame structure.

4.2.1. Sleeping strategy

As we assume that our WSN collects data from different sensors and periodically transmits them to the sink, the sensors can go to the sleep state when they have no data to transmit or receive. To minimize the energy consumption in the network, we exploit the routing tree to propose a new sleep/awake strategy that ensures that nodes belonging to two consecutive layers in the routing tree do not transmit their data at the same time. The layers are numbered such as the layer *1* (respectively layer *L*) is one-hop (respectively L-hops) away from the sink.

Then, the time is divided in frames depending on the number of layers in the network (the hop number to the sink). Each frame includes different time slots for the broadcasting (*Broad*), receiving (*Receive*), processing (*Pr*) and sending (*Send*) data. Fig. 4 shows the frame structure in each layer. In the first slot (*Broad*), all the sensor nodes are awake to exchange control messages (routing, heartbeat, neighborhood information,...). Then, all sensors go to sleep except those of the last layer *L* that stay awake to collect data, processing them and send them in the *Send* slot. Before the layer *L Send* slot, the layer *L-1* sensors wake up to receive the data from the layer *L*'s nodes. This behavior is propagated through the *L-1* remaining layers by ensuring that at each time the layer l $(2 \leq l \leq L-1)$ nodes can send their data to their upper layer *l-1* (their parents' layer in the routing tree) and then go to sleep. According to this strategy, two nodes belonging to different layers will never transmit data simultaneously.

Once the sleep/awake strategy is defined, we proceed to the initial allocation using the routing information.

4.2.2. Initial routing-based channel allocation

We use the well known graph theory, in particular graph coloring, to assign a channel to each node. As we said in previous Sections, the number of available channels is limited, then we define a channel allocation strategy based on routing trees to optimally minimize the number of interferences for each logical node. The proposed solution exploits the fact that siblings cannot send packets simultaneously even if they use different channels. The network is divided into layers as described in the previous Section. The CR-RMC approach handles each layer independently, which means that the approach tries to minimize the interference ratio between the nodes belonging to the same layer. This technique considerably reduces the number of graph vertices. We extract the reduced graph $G_l = (V_l, E_l)$ from the graph G, where V_l denotes the set of layer l's logical nodes and E_l the set of links between these nodes. Each logical node is represented by its parent which is responsible for the channel allocation.

In the initialization phase, the routes are built and neighborhood information must be exchanged between nodes to detect and recover from articulation nodes' failures. Each node collects information to build a local view of its vicinity. This information includes positions, neighbors, parent and layer of its 1-hop, 2-hop and 3-hop neighbors. Indeed, the parent i of a logical node N_i can determine its degree $c(N_i) = \sum_{\forall N_j} c(N_i, N_j)$ (the sum of interference links with all its LNs neighbors), where $c(N_i, N_j)$ is given by the Eq. (17). This degree, as well as the links costs, are broadcast to the logical nodes belonging to the same layer.

The CR-RMC approach assigns a color to each logical node. This assignment is handled by each LN's parent and follows the process described in Algorithm 2. To minimize the conflicting links, the

Algorithm 2 Routing-based multi-channel allocation.

for $\forall i \in V_l$ **do**
  Lt=Tri$(N_i, c(N_i))$
  ch(N_i)*=random*$(1,|M|)$
end for
$C = \{C_1, \dots, C_M\} = \emptyset$ the set of colored nodes, in which class C_j includes the nodes colored by j.
k=1 (color);
l=L (layer);
while $l \geq 2$ **do**
  $C_k = \{N_i | c(N_i) = max_j c(N_j) \wedge |N_i| = max_j |N_j|, \ \forall N_j \in Lt\}$;
  Remove N_i from Lt;
  $C_k = C_k \cup \left\{N_j | c(N_j, N_i) = 0, \forall N_j \in Lt \wedge \forall N_i \in C\right\}$;
  Remove N_j from Lt;
  repeat
    $k = k+1$;
    $C_k = \left\{N_j = argmax_{N_i \in C}(c(N_j, N_i)) | (N_j, N_i) \in E_l\right\}$;
    Remove N_j from Lt;
  until $Lt = \emptyset$ or $k = L+1$
  if $Lt \neq \emptyset$ **then**
    repeat
      $m = \left\{ch(N_i) | N_i = argmin_{N_j \in C}(\Delta(N_j))\right\}$;
      $C_m = C_m \cup \left\{N_j\right\}$;
    until $Lt = \emptyset$
  end if
  $l = l - 1$;
end while

parent i of the logical node N_i having the highest degree among the logical nodes belonging to layer L starts by coloring its children($\in N_i$). If many logical nodes have the same degree, the parent with the highest identity will be prioritized. Then, i broadcasts the chosen color with its identity to layer L's nodes. All logical nodes not interfering with the node N_i choose the same color as they do not generate any conflict and broadcast their identities and colors. When all possible nodes are colored, the parents of nodes having a link with the colored nodes assign a different color to their children. To avoid any conflict, the nodes are colored one by one, while prioritizing the links with highest costs. As the information about the links' costs is known, each node can easily identify if it is prioritized or not. If no more color is available, the node is assigned the color of the node having the lowest conflicts' number. If all the nodes of the layer L are colored, CR-RMC moves to the next layer (*L-1*) and the coloring process is repeated. Moreover, when moving from a layer l to the layer *l-1*, CR-RMC tries to color a node N_i by the color assigned to the most of its descendants. The steps of logical graph coloring is illustrated by Fig. 5.

Once all the nodes are assigned to channels, the nodes start transmitting data using the assigned channels and according to the sleep/awake strategy until an articulation node failure. As CR-MC approach, CR-RMC uses the same mechanisms of failure detection and connectivity restoration described in Section 3.2.2. However, the mechanism of channel reallocation is different and is based on the routing decision for this latter approach.

4.2.3. New channel reassignment

The articulation node failure recovery requires relocating some nodes. These nodes as well as their children try to find new parents with new channels. We aim to minimize the impact of the

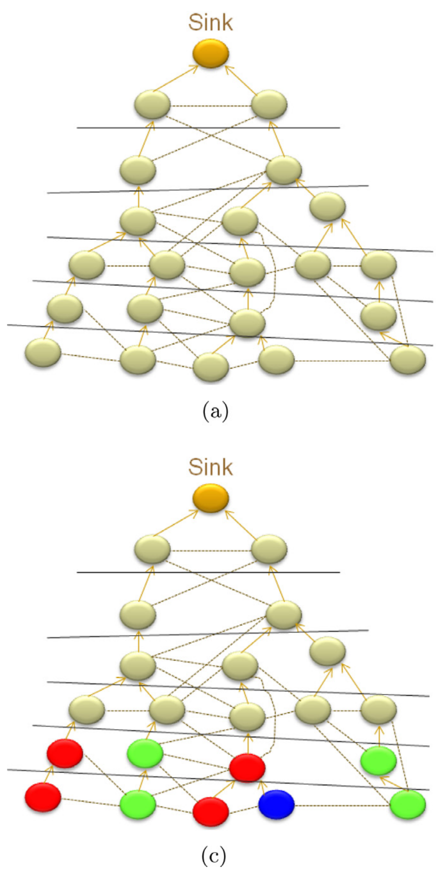


Fig. 5. Initial graph coloring.

recovery process on the whole network by restricting the changes (node parent, layer and channel) in the reorganized zone. Following the previously defined sleep/awake strategy, each node try to maintain its current network layer when looking for a new parent.

The new parent search process performs as follows:

- The node v updates its neighboring table by exchanging *hello* messages.
- v triggers a timer T and broadcasts a *parent_request* message containing its ID, its layer, and its new neighbors.
- When a node u having a parent receives this message, it calculates the new number of conflicts if v becomes its descendant. Then u sends a *parent_candidate* message to v with its ID, its layer, its channel and the number of conflicts.
- If u has no parent, it saves the *parent_request* message and waits for its own *parent_request* answer.
- If v receives a *parent_candidate* message with the same layer as its old parent and zero conflicts, it cancels immediately the timer T and returns a *parent_join* message to the sender. The node v also sends a *parent_candidate* message as an answer to all the saved *parent_request* messages.
- If this is not the case, v waits until the timer end and then selects the best candidate to be its parent. The selection is based first on the candidate layer, then on the number of conflicts.
- If the layer is changed, v must informs its own sub-tree nodes to update their layers and the sleep/awake frame.

At this stage, a problem can arise: what happens if the number of layers increases? If only one layer is added, the problem is solved; the nodes belonging to the layer *L+1* can send their data at the *Receive* slot of the *L* layer. However, if more than one layer is added in the network, the sink is notified and sends a new frame with the updated layer number.

5. Complexity and analysis

In this section, we analyze the approaches CR-MC and CR-RMC to compare their performances in terms of complexity of the algorithms and the number of control messages. We focus on the number of comparison operations and the number of control messages sent when each approach is performing. The fact that the two approaches are distributed implies that the mechanisms of failure recovery only affect a small area around the failed node. Thus, the complexity of CR-MC and CR-RMC is independent of the network size (in term of number of nodes). However, this complexity necessarily depends on the network density and topology. To evaluate the proposed approaches, we introduce the following lemmas.

Lemma 1. *The number of messages generated by CR-MC and CR-RMC is* $O(\Delta^3)$*, where* Δ *is the maximum neighborhood degree.*

Proof. During the network reorganization around the failed articulation node n_f, each of its 1-hop neighbors runs the recovery algorithms independently of each other. Thus, no message needs to be exchanged between these nodes. However, the 2-hop neighbors must be informed of the decision taken by the 1-hop neighbors. Let $H_1(n_f)$ and $H_2(n_f)$ be the 1-hop and the 2-hop neighbors sets and n_d the number of relocated nodes. The number of control messages sent during the reorganization phase is:

1. $|H_1(n_f)|$ if $N_d \le |H_1(n_f)|$.
2. N_d if $N_d > |H_1(n_f)|$.

Moreover, the number of Steiner Points (n_d) cannot exceed $|H_2(n_f)| - 2$. As each relocated node informs its neighbors of its relocation, in both cases (1) and (2), the number of the exchanged messages cannot exceed $max(|H_1(n_f)|, |H_2(n_f)| - 2)$. Therefore, if we denote the maximum neighborhood degree of a node by Δ, the number of emitted messages is bounded by $\Delta^2 - 2$. This is true for every $\Delta > 1$. Consequently, the number of exchanged messages during the reorganization phase is $O(\Delta^2)$.

During the channel reallocation phase, using CR-MC, each relocated node i sends three control message (*hello, conflicts* and *decision*) and receives in the worst case $2\Delta_i + 1$ messages (*hello* and *suggestion*) with Δ_i is the node i neighborhood degree. The total number of messages generated by all the relocated nodes (called set V_d) is given by:

$$N_m = \sum_{\forall i \in V_d} (2\Delta_i + 4) \tag{28}$$

We use the same method as for the reorganization phase to get:

$$\sum_{\forall i \in V_d} (2\Delta_i + 4) \leq N_d (2\Delta_i + 4) \tag{29}$$

As $N_d < \Delta^2 - 2$ and $\Delta_i \leq \Delta$, (29) can be written as:

$$\sum_{\forall i \in V_d} (2\Delta_i + 4) < (\Delta^2 - 2)(2\Delta + 4) \tag{30}$$

Then

$$\sum_{\forall i \in V_d} (\Delta_i + 2) < \Delta^3 + 2\Delta^2 - 4\Delta - 4 \tag{31}$$

Consequently, the number of control messages exchanged during the channel reallocation phase is $O(\Delta^3)$ for CR-MC. In the same way, we can demonstrate that the number of messages exchanged during the same phase is $O(\Delta^3)$ for CR-RMC.

The upper bound Δ^3 of the exchanged messages' number is very satisfactory for both small and large scale WSNs. For instance, in a small or sparse WSN, the node's degree Δ is very small. Hence, if we are in the worst case, the total number of messages remains small. However, in dense WSNs, where Δ increases, the replacing nodes trend to be selected only among the 1-hop neighbors hence resulting in a number of generated messages $<< \Delta^3$.

We consider now the complexity of each approach in term of the comparison operations' number. We focus on the operations executed during the channel reassignment step. □

Lemma 2. *During channel reallocation, CR-MC and CR-RMC execute $O(\Delta)$ operations of comparison.*

Proof. In CR-MC, each node i looks for an available channel availability by checking the *hello* messages received from its Δ_i neighbors. In the worst case, the node executes Δ_i comparisons by checking all the *hello* messages. The node i can also receives at most Δ_i *suggestion* messages if no channel is available. Then, i executes Δ_i other comparisons. The total number of comparison operations cannot exceed $2\Delta_i$. If we consider $\Delta = \max_{i \in V} \Delta_i$, then the complexity of the channel reallocation algorithm is $O(\Delta)$.

In the same way, using CR-RMC, each node j receives at most Δ_j *parent_candidate* messages. The node executes two comparisons to check the layer and the number of conflicting links before selecting its new parent. Then, as for CR-MC, the complexity of the channel reassignment algorithm, used by CR-RMC, is $O(\Delta)$ □

6. Performance evaluation

In this section, the proposed solution is evaluated through simulation using Omnet++ and compared to an exact solving method Min-Max method [25] and the centralized approach RNFR proposed in [14] where all the recovery process is executed by the sink.

Table 1
Simulation parameters.

Parameter	Value
Area size	1000m × 1000m
Number of sensors	[15,250]
Number of channels	4
R_c	100m
Number of runs	≥ 30

6.1. Simulation parameters

In all the simulation scenarios, the sensor nodes are randomly scattered in the simulation area. The simulation parameters are summarized in Table 1.

To evaluate the performance of the two approaches CR-MC and CR-RMC, we focus on the following metrics:

1. *The number of conflicts*: denotes the number of conflicting links causing collisions and retransmission.
2. *The number of re-colored nodes*: reports the number of nodes assigned a new channel after the failure recovery.
3. *The throughput*: reports the percentage of useful data regarding the all transmitted data.
4. *The number of relocated nodes*: depicts the impact of the connectivity restoration process on the network topology.
5. *The total traveled distance*: this metric denotes the distance traveled by all moved nodes to recover from the failure.
6. *The routes' stability*: this metric reports the routing tree changes after the failure recovery in terms of the channel used by each node and its layer (the number of hops towards the sink).

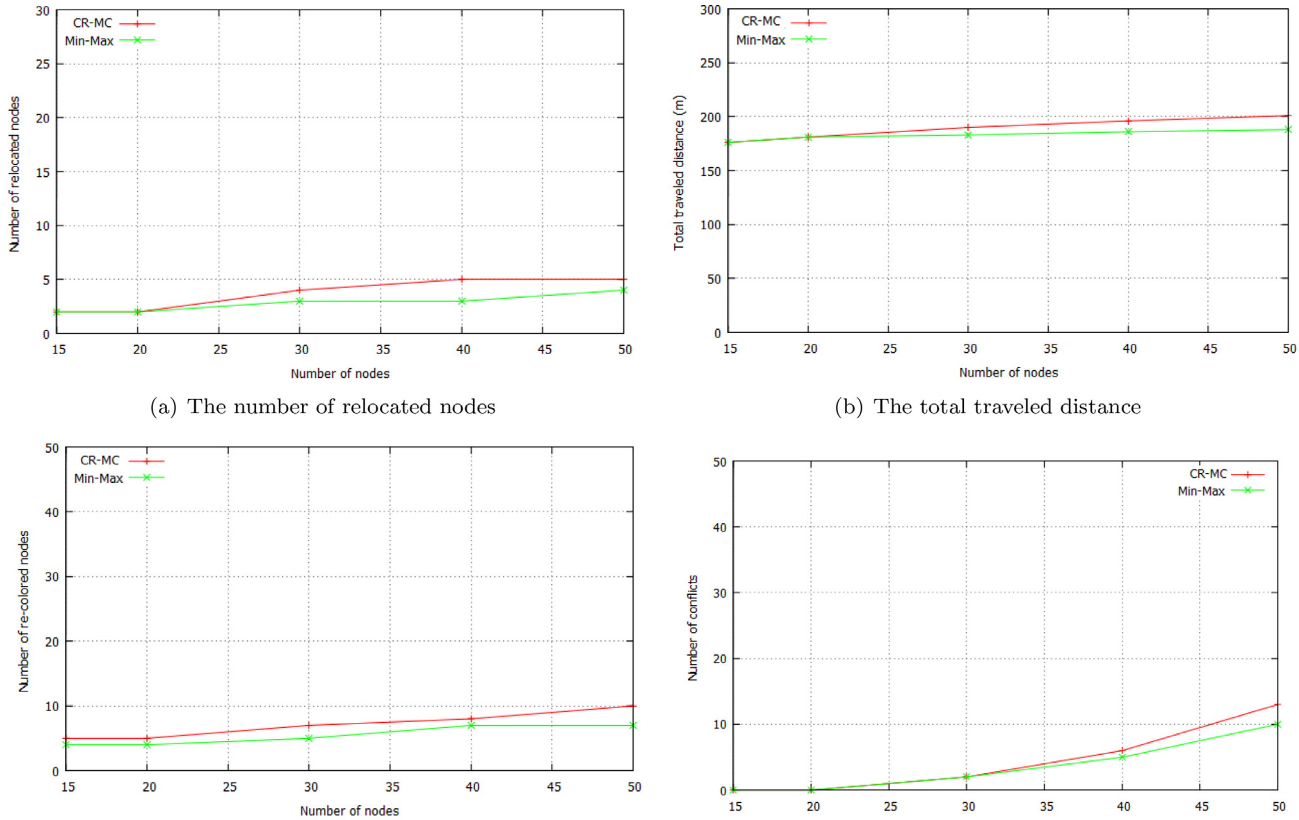
6.2. Results

Fig. 6 shows the performance of the CR-MC approach compared to an exact solving of the presented multi-objective optimization problem using Min-Max method [25]. This figure depicts that our proposed heuristics give a near optimal results, in particular in small WSNs. For instance, CR-MC generates the same number of relocated nodes as the Min-Max method in a WSN with 20 nodes, as well as the optimum number of conflicts (number of neighbor nodes having the same channel) in WSNs with 15 and 20 and 30 nodes. The results deviate slowly from the optimum when increasing the number of nodes.

Fig. 6(a) shows that the number of relocated nodes increases with the number of nodes in the network. This result is due to the fact that the number of 2-hop neighbors used to find new locations for the relocated nodes increases with the network density. As a consequence, the total traveled distance becomes more important too as depicted in Fig. 6(b). However, the average distance traveled by each node decreases from 88 m in a WSN with 15 nodes to 40.2 m in a WSN with 50 nodes. We also notice that the percentage of relocated nodes, regarding the total number of nodes, remains quasi unchanged (between 10% and 13%).

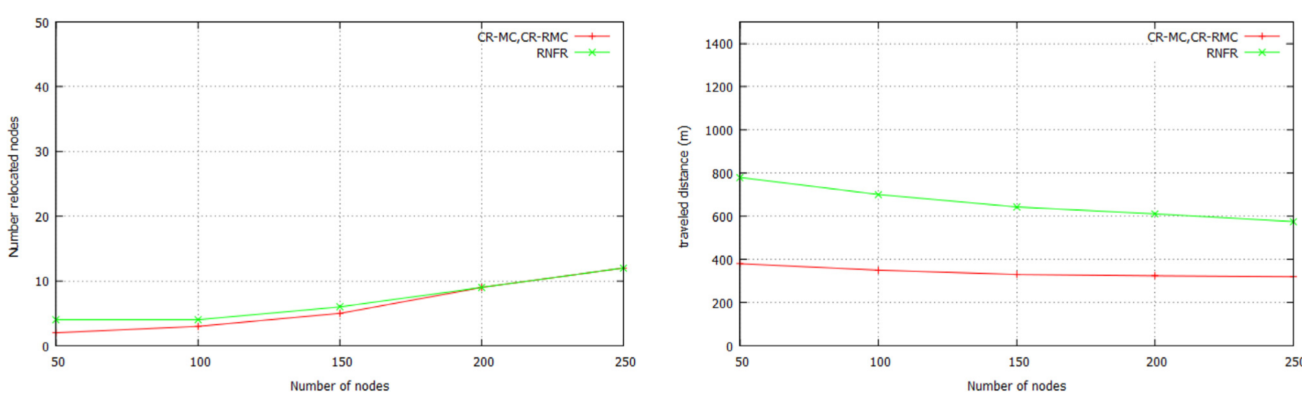
Fig. 6(c) and (d) depict that even if more nodes change their channels, the number of conflicts increases with the total number of nodes in the network. This result is explained by the fact that more nodes may interfere with each other when the network density becomes more important. The CR-RMC approach uses the same node replacement techniques as CR-MC. Then, these results concerning the total traveled distance and the number of relocated nodes are still the same for CR-RMC.

Fig. 7(a) depicts the number of relocated nodes as a function of total the nodes' number. As shown in this figure, the two distributed approaches CR-MC and CR-RMC outperforms the centralized approach RNFR [23]. The additional relocated nodes in RNFR



(a) The number of relocated nodes
(b) The total traveled distance
(c) The number of recolored nodes
(d) The number of conflicts

Fig. 6. Comparison of the CR-MC approach to the Min-Max method.



(a) The number of relocated sensor nodes
(b) The total traveled distance (m)

Fig. 7. The number of relocated nodes and the total traveled distance.

are the rotating nodes used to communicate the recovery information from one segment to another. However, the three approaches use the same number of relocated nodes in dense networks as all the rotating nodes are participating in the connectivity restoration process.

In Fig. 7(b), we notice that CR-MC and CR-RMC generate the same distance as they use the same mechanism to select the nodes occupying the Steiner Points. The longest distance among the three approaches is one generated by RNFR due to the extra rotation distance traveled by the nodes to disseminate the recovery information to the isolated network segments.

Fig. 8 illustrates that CR-RMC offers a very high stability degree, compared to CR-MC and RNFR, in particular in dense WSNs. This is explained by the fact that in the first approach, each node prioritizes the candidates with the same layer and the same channel when looking for a new parent. The two other approaches RNFR and CR-MC only consider the number of conflicts when executing; while the routes are updated by the routing protocol. Moreover, the routes become more stable when the network density increases. This stability degree is a result of the high probability that a node find a new parent with the same layer and channel as its old parent. For instance, CR-RMC offers a stability degree equal to 85% in a WSN with 50 nodes (compared to 62% for CR-MC and RNFR). This degree reaches 98% in a network with 250 nodes (80% and 81% for CR-MC and RNFR respectively).

Fig. 9(a) represents the number of generated conflicting links. We notice that CR-RMC offers better performance than CR-MC and RNFR especially in dense networks. The number of conflicting links

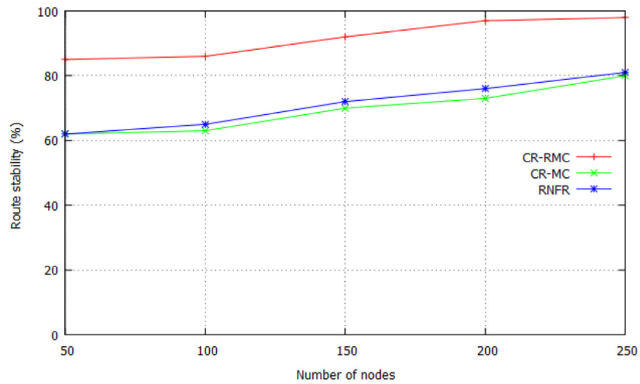


Fig. 8. Stability of routes after the failure recovery.

is still lower than 200 links in a WSN with 250 nodes for CR-RMC while the two other approaches generate more than 500 conflicting links in the same network. Indeed, the introduction of logical nodes allows better exploitation of the available channels by assigning the same channel to the nodes having the same parent. Moreover, RNFR generates less conflicting links than CR-MC. This can be explained by the distributed aspect of this latter approach that only considers the sensor nodes' small vicinity when assigning the channels.

As in Fig. 9(a), (b) shows that CR-RMC offers a better performance than Cr-MC and RNFR in term of number of nodes reassigned new channels. This efficiency becomes more remarkable in dense networks. For instance, after the failure recovery, CR-RMC reassigns new channels to only 45 nodes in a WSN with 250 nodes compared to 75 nodes and 89 nodes for CR-MC and CR-RMC respectively.

Fig. 10(b) compares the throughput generated by the three approaches as a function of the total number of nodes. The throughput is defined as the rate of bits transmitted during a time period (second). We consider that a 100% of theoretical throughput is achieved (over all the network) if all the senders and the receivers use the same channels and do not need to switch their radio between channels. A node takes 200 µs to switch its radio from a channel to another witch decreases the time dedicated to transmission and then the number of bits that can be transmitted by second. This figure shows that CR-RMC offers a better throughput than CR-MC and RNFR due to its small number of radios' switching compared to RNFR and CR-MC (Fig. 10(a)). CR-RMC generated only 150 radios' switching in a WSN with 250 nodes compared to 593 and 653 radios' switching generated by CR-MC and RNFR, respectively. With CR-RMC, the throughput is still important (> 92%) even in dense networks; while it decreases to reach 78% and 76% for RNFR and CR-MC, respectively. This is explained by the increasing number of radios' switching that leads to more wasted time.

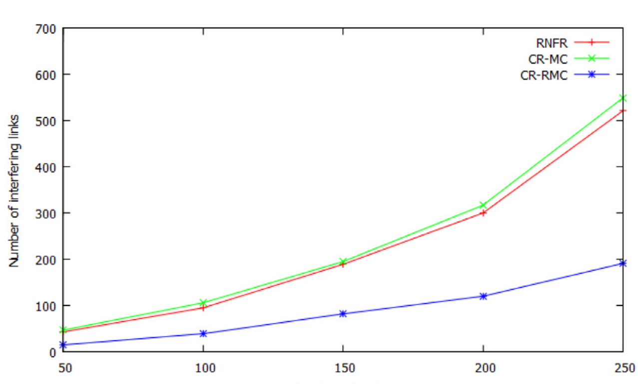
Fig. 11 depicts the coverage of the network after the failure recovery compared to the coverage before the failure as the number of nodes increases. As CR-RMC and CR-MC use the same recovery techniques, their results are the same. However, the area coverage is enhanced after the nodes' relocation. This result is explained by the fact that we replace one node by many other node and we distribute them over all the articulation node vicinity. In addition, the relocation process is restricted in a localized area which minimize the impact of the recovery on the remaining network coverage.

Fig. 12 compares CR-RMC and CR-MC overheads in terms of the number of control messages during the recovery and channel reallocation phases. The obtained results show that CR-RMC generates more messages than CR-MC. The difference between the two approaches is due to the message exchange between the nodes and their parents, in CR-RMC. This difference becomes more important with the increasing network size. For instance, CR-RMC generates 40 messages in a WSN with 50 nodes and 121 in a network with 250 nodes. These values respectively decrease to 39 and 81 when CR-MC is used.

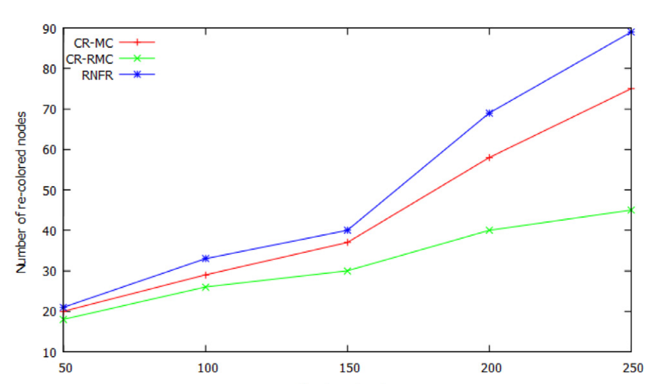
The last Fig. 13 shows the impact of the overhead on the lifetime average of sensor nodes in function of network size. We compared the node lifetime when CR-MC and CR-RMC regarding the lifetime average when no recovery technique is used. We notice that when the number of nodes increases the lifetime average decreases. For instance, with 50 nodes, the two approaches offered lifetime averages very close to the no recovery case. However, CR-MC slightly outperforms CR-RMC in sparse networks (97.6% for CR-MC and 96.12% with 100 nodes). CR-RMC becomes more efficient in dense networks (82.76% and 87.3% for respectively CR-MC and CR-RMC with 250 nodes). This is a direct result of the smaller ratio of interferences and retransmissions when using CR-RMC. On other hand, the network lifetime is enhanced as WSN becomes nonfunctional after the failure of an articulation node. Then, the application of a recovery technique will extend the network lifetime even if they generate some overhead. In our opinion, the reduction in the node lifetime average is still minimal regarding the advantages of the recovery approaches.

7. Conclusion

In this paper, we discussed the failure of an articulation node in multichannel WSNs which leads to the network partitioning into several disconnected segments. We first proposed a distributed fault recovery solution, called Connectivity Restoration for Multi-

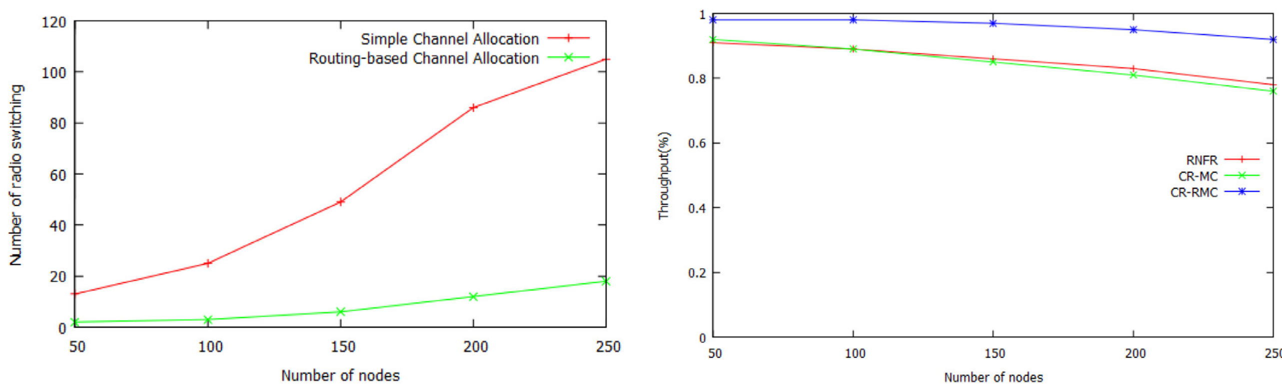


(a) The number of conflicting links



(b) The number of nodes reassigned new channel

Fig. 9. The number of conflicting links and reassigned nodes.



(a) The number of radios' switching (b) The WSN throughput

Fig. 10. The number of radios' switching and its impact on the throughput.

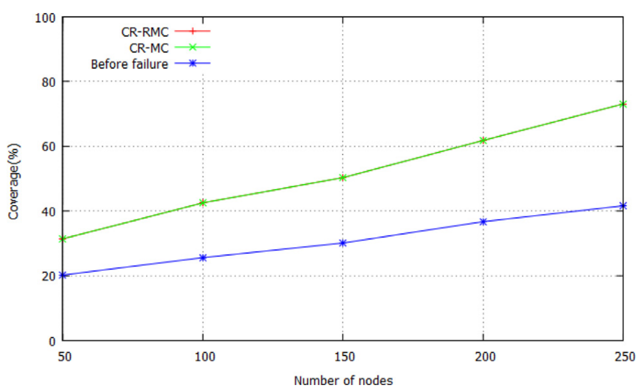


Fig. 11. The coverage percentage of the network.

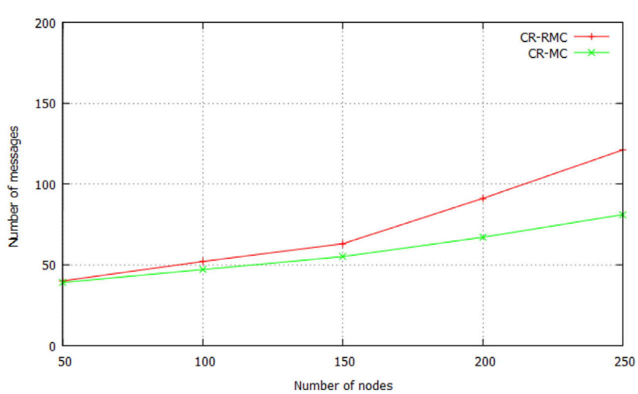


Fig. 12. The number of control messages (overhead).

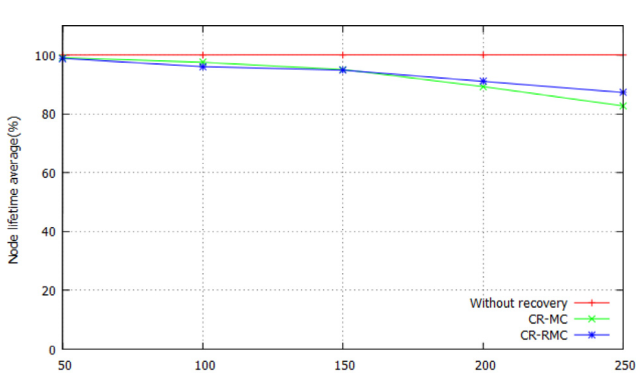


Fig. 13. The lifetime average of a node.

Channel WSNs (CR-MC). Therefore, we formulate the problem of connectivity restoration in the multichannel context as a multi-objective optimization problem. Then, we introduced some mechanisms dedicated to reconnect the different network segments by reorganizing the network around the failed node and reassigning new channels to the nodes. The CR-MC approach only uses the neighborhood information while executing the channel reallocation and does not consider the routes followed by the data towards the sink. Thus, we proposed a second solution, called Connectivity Restoration for Routing based Multi-Channel WSNs (CR-RMC), that exploits the routing tree as well as the neighborhood information during the channel allocation/reallocation phases. In this approach, we introduce the logical node concept denoting the nodes having the same parent. This will reduce the network size and then allow a better exploitation of the available channels. The performance of CR-MC and CR-RMC is evaluated by simulation and compared to a centralized approach. All the performance results showed that CR-RMC offers better performance than the two other approaches thank to the logical nodes and the routing tree consideration which ensure an efficient channel reallocation while executing the recovery process. As a next step, we will focus on large scale WSNs where we will try to extend our solution to hierarchical sensor networks where simultaneous failures can occur.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Comput. Netw. 38 (2002) 393–422.

[2] Y. Wu, J. Stankovic, T. He, S. Lin, Realistic and efficient multi-channel communications in wireless sensor networks, in: In Proceedings INFOCOM, 2008, pp. 1867–1875.

[3] K.R. Chowdhury, N. Nandiraju, D.P.A. P. Chanda, Q.A. Zing, Realistic and efficient multi-channel communications in wireless sensor networks, Ad Hoc Netw. 7 (2009) 307–321.

[4] O.D. Incel, L.V. Hoesel, P. Jansen, P. Havinga, Mc-lmac: a multi-channel mac protocol for wireless sensor networks, Ad Hoc Netw 9 (2011) 73–94.

[5] G. Zhou, C. Huang, T.H. T. Yan, J.A. Stankovic, T.F. Abdelzaher, Mmsn: multi--frequency media access control for wireless sensor networks, in: Proceedings IEEE International Conference on Computer Communications INFOCOM, 2006, pp. 1–13.

[6] Y. Kim, H. Shin, H. Cha, Y-Mac: an energy-efficient multi-channel mac protocol for dense wireless sensor networks, in: Proceedings IPSN, 2008, pp. 53–63.

[7] Y. Yang, W. Yi, Rainbow: reliable data collecting MAC protocol for wireless sensor networks, in: In Proceedings Wireless Communications and Networking Conference WCNC , 2010.

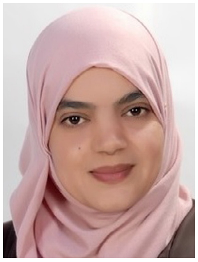
[8] D. Brelaz, New methods to color the vertices of a graph, Mag. Commun. ACM 22 (1979) 251–256.

[9] L. Wang, Z. Li, An approximation algorithm for a Bottleneck K- Steiner tree problem in the euclidean plane, Inf. Process. Lett. 81 (2002) 151–156.

[10] S. Chouikhi, I.E. Korbi, Y. Ghamri-Doudane, L.A. Saidane, A survey on fault tolerance in small and large scale wireless sensor networks, Comput. Commun. Elsevier 69 (2015) 2237.

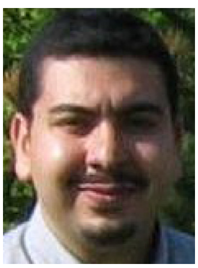
[11] M. Younis, R. Waknis, Connectivity restoration in wireless sensor networks using Steiner tree approximations, in: IEEE Global Communications Conference (GLOBECOM), 2010, pp. 1–5.

[12] S. Lee, M. Younis, Recovery from multiple simultaneous failures in wireless sensor networks using minimum Steiner tree, J. Parallel Distrib. Comput. 70 (2010) 525–536.
[13] N. Tamboli, M. Younis, Coverage-aware connectivity restoration in mobile sensor networks, in: International Conference on Communications ICC , 2009.
[14] M. Younis, S. Lee, S. Gupta, K. Fisher, A localized self-healing algorithm for networks of moveable sensor nodes, in: Proc. of the IEEE Global Communications Conference GLOBECOM , 2008.
[15] Z. Mi, R.S. Hsiao, Z. Xiong, Y. Yang, Graph-theoretic based connectivity restoration algorithms for mobile sensor networks, Int. J. Distrib. Sens.Netw. (2015).
[16] A. Abbasi, M. Younis, U. Baroudi, Recovering from a node failure in wireless sensor-actor networks with minimal topology changes, IEEE Trans. Veh. Technol. 62 (2013) 256–271.
[17] M. Imran, M. Younis, A.M. Said, H. Hasbullah, Localized motion-based connectivity restoration algorithms for wireless sensor and actor networks, J. Netw. Comput. Appl. 35 (2012) 844856.
[18] V. Ranga, M. Dave, A.K. Verma, A hybrid timer based single node failure recovery approach for WSANs, Wirel. Pers. Commun. 77 (2014) 21552182.
[19] N. Haider, M. Imran, M. Younis, M.G. N. Saad, A novel mechanism for restoring actor connected coverage in wireless sensor and actor networks, in: IEEE International Conference on Communications (ICC), 2015, pp. 6383–6388.
[20] S. Lee, M. Younis, M. Lee, Connectivity restoration in a partitioned wireless sensor network with assured fault tolerance, Ad Hoc Netw. 24 (2015) 1–19.
[21] H. Essam, M. Younis, E. Shaaban, Minimum cost flow solution for tolerating multiple node failures in wireless sensor networks, in: IEEE International Conference on Communications (ICC), 2015, pp. 6475–6480.
[22] S. Chouikhi, I.E. Korbi, Y. Ghamri-Doudane, L.A. Saidane, Fault tolerant multi-channel allocation scheme for wireless sensor networks, in: Proceedings Wireless Communications and Networking Conference (WCNC), 2014, pp. 2438–2443.
[23] S. Chouikhi, I.E. Korbi, Y. Ghamri-Doudane, L.A. Saidane, Articulation node failure recovery for multi-channel wireless sensor networks, in: IEEE Global Communications Conference GLOBECOM, 2015, pp. 1–7.
[24] G. Robins, A. Zelikovsky, Tighter bounds for graph Steiner tree approximation, SIAM J. Discrete Math. 19 (2005) 122–134.
[25] C.A. Coello, G.B. Lamont, D.A.V. Veldhuizen, Evolutionary algorithms for solving multi-objective problems, Springer , 2007.



Samira Chouikhi received the engineering and the MS degrees in computer science from the National School of Computer Science, Tunis, Tunisia, in 2006 and 2009, respectively. She is currently a PhD student jointly at the National School of Computer Science, Tunisia and Universit Paris-Est, France. She is a researcher at CRISTAL Laboratory, Tunisia and LIGM Laboratory, France. Her research work focuses on the fault tolerance in Wireless sensor networks.

Ines El Korbi received the engineering, masters, and PhD degrees in computer science in 2003, 2005, and 2009, respectively, from the National School of Computer Science, University of Manouba, Tunisia. She is currently an assistant professor at High Institute of Computer Science, Tunisia and a senior researcher at National School of Computer Science, Tunisia. She is preparing the associate professor degree in the computer networks domain. She supervises several masters and PhD theses, especially in the wireless networks field. Her researches are conducted in cooperation with many scientific research teams (INRIA Rocquencourt, France, LIGM Paris Est, France, LIP-6 UPMC Paris VI, France, and University of the District of Columbia (UDC)). She is a member of the IEEE.



Yacine Ghamri-Doudane is currently Full Professor at the University of La Rochelle (ULR) in France, and member of its Laboratory of Informatics, Image and Interaction (L3i). Yacine received an engineering degree in computer science from the National Institute of Computer Science (INI), Algiers, Algeria, in 1998, an M.S. degree in signal, image and speech processing from the National Institute of Applied Sciences (INSA), Lyon, France, in 1999, a Ph.D. degree in computer networks from University Pierre and Marie Curie, Paris 6, France, in 2003, and a Habilitation to Direct Research (HDR) in Computer Science from Universit Paris-Est, in 2010. His current research interests lays in the area of wireless networking and mobile computing with a current emphasis on topics related to the Internet of Things (IoT), Wireless Sensor Networks (WSN) and Vehicular Networks.

Leila Azouz Saïdane is Professor at the National School of Computer Science (ENSI), at The University of Manouba, in Tunisia. She was the Director of this school, the Chairperson of the PhD Commission and the supervisor of the Masters Degree program in Networks and Multimedia Systems at ENSI. She is the responsible of RAMSIS group of CRISTAL Research Laboratory (Center of Research in Network and System Architecture, Multimedia and Image Processing) at ENSI. She collaborated on several international projects. She is author and co-author of several papers in refereed journals, magazines and international conferences.