J Inf Process Syst, Vol.11, No.2, pp.205~228, June 2015
http://dx.doi.org/10.3745/JIPS.02.0019

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

A Virtual Laboratory to Practice Mobile Wireless Sensor Networks: A Case Study on Energy Efficient and Safe Weighted Clustering Algorithm

Amine Dahane*, Nasr-Eddine Berrached*, and Abdelhamid Loukil*

Abstract

In this paper, we present a virtual laboratory platform (VLP) baptized Mercury allowing students to make practical work (PW) on different aspects of mobile wireless sensor networks (WSNs). Our choice of WSNs is motivated mainly by the use of real experiments needed in most courses about WSNs. These experiments require an expensive investment and a lot of nodes in the classroom. To illustrate our study, we propose a course related to energy efficient and safe weighted clustering algorithm. This algorithm which is coupled with suitable routing protocols, aims to maintain stable clustering structure, to prevent most routing attacks on sensor networks, to guaranty energy saving in order to extend the lifespan of the network. It also offers a better performance in terms of the number of re-affiliations. The platform presented here aims at showing the feasibility, the flexibility and the reduced cost of such a realization. We demonstrate the performance of the proposed algorithms that contribute to the familiarization of the learners in the field of WSNs.

Keywords

Clustering, Energy Efficiency, Practical Work, Security Attacks, Virtual labs, Wireless Sensor Networks

1. Introduction

Laboratory experiences play a central role in an advanced technical education. They allow students to see how concepts can be put into practice, enabling them to appreciate the real-world implications of what can seem at first very abstract. They also have the potential for increasing the synergy between research and education [1]. Practical work in laboratory to study WSNs is a very challenging task for students in computer science and electronics. A sensor is an electronic device with limited resources (processing, storage, battery power and bandwidth).Sensor nodes are randomly and densely deployed in a sensed environment [2, 3]. If we exclude systems which are dedicated to wire line networks such as those presented in [1], no proposal of a virtual platform devoted to study the practical aspects of WSNs has been made up to now. There are only expensive and cumbersome simulators such as OMNet++ (Objective Modular Network Testbed in C++) [4], Castalia [5], NS2 (Network Simulator_2) [6], and Opnet [7], for which practical works require the physical presence of the students as well as the availability of a great number of nodes in the classroom. For example, to make a simulation study on

※ This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/3.0/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Manuscript received May 23, 2014; first revision September 18, 2014; accepted November 20, 2014; onlinefirst April 20, 2015.
Corresponding Author: Amine Dahane (amine_usto.info@yahoo.fr)
* Intelligent Systems Research Laboratory, University of Sciences and Technology of Oran, Algeria (amineusto.laresi@gmail.com, laresi_usto_2012@hotmail.fr, loukil_ah@yahoo.fr)

Copyright© 2015 KIPS

WSNs using NS2 environment, it is necessary to have skills in both Tcl (Tool Command Language) [8] and C++.

Using virtual laboratories is a possible alternative to overcome these difficulties and contribute to make a PW on different aspects of WSNs like clustering, routing, medium access control, aggregation, and security. A virtual laboratory is defined, according to [9], as a digital work space where distance collaboration is allowed and research experimentation is conducted to produce and deliver results using distributed information of the new communication technologies.

This paper proposes a VLP named "Mercury" for practicing WSNs by studying the clustering concept. This environment is developed by using a unified modeling language (UML) meth odology. Clustering means grouping nodes which are closed to each other and it has been widely studied in ad-hoc networks [10-17]. More recently, it has been used in WSNs [18-21] where the purpose in general is to reduce useful energy consumption and routing overhead. Fig. 1 illustrates how inside the cluster [27], two kinds of nodes can be found: one node called cluster head (CH) or coordinator (in Fig. 1: CH1, CH2 and CH3) which is responsible for coordinating the cluster activities and several ordinary nodes called cluster members (CMs) (in Fig. 1: CM1 and CM2) that have direct access only to one CH. An ordinary node which is able to hear two or more CHs, is called a gateway (G) (in Fig. 1: the gateway G2 can hear CH1, CH2 and CH3, while the gateway G1 can hear CH1 and CH2) instead. So, each communication initiated by a cluster member to a destination inside the cluster must pass by CH. If the destination is outside the cluster, the communication must be forwarded by a gateway. Recent research studies recognize that organizing mobile WSNs, in the sense defined above, into clusters by using a clustering mechanism is a challenging task [16-22]. This is due to the fact that CHs carry out extra work, and consequently consume more energy compared to CMs during the network operations and this will lead to untimely death causing network partition and therefore failure in communication link. For this reason, one of the most frequently encountered problems in this mechanism is to search for the best way to elect CH for each cluster. Indeed, a CH can be selected by computing the quality of nodes. This may depend on several metrics: connectivity degree, mobility, residual energy and the distance of a node from its neighbors. Significant improvement in performance of this quality can be achieved by combining these metrics [10-12, 17, 19].

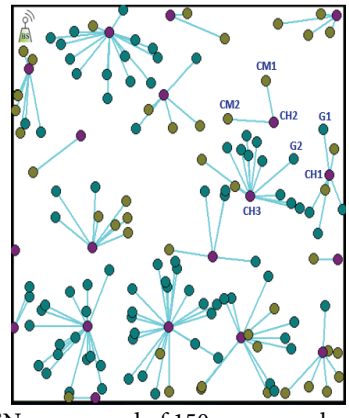


Fig. 1. Clustering formation of WSNs composed of 150 sensor nodes deployed in a 570 m × 555 m space area with a radio range=100 m [27].

Accordingly, we propose energy efficient and safe weighed clustering algorithm for mobile WSNs using a combination of the above metrics with the behavioral level metric which we have added. This metric is decisive as it allows the proposed clustering algorithm to avoid any malicious node in the neighborhood to become a CH, even if the remaining metrics are in its favor.

The election of CHs is performed using weights of neighboring nodes which are computed on the basis of selected metrics. So this strategy ensures the election of legitimate CHs with high weights. The preliminary results obtained through PW study demonstrate the effectiveness of our algorithm in terms of the number of equilibrate clusters and the number of re-affiliations, compared to WCA (Weighted Clustering Algorithm) [10] and DWCA (Distributed Weighted Clustering Algorithm) [11].

These results also reveal that our approach is suitable if we plan to use it in network layer reactive routing protocols instead of proactive ones once the clustering mechanism is launched.

We can enumerate the contributions of our paper as follows:

- The Design and development of a VLP allowing learners to study and make PW on different aspects of mobile WSNs, with lower costs by using UML methodology.
- Teach the learners how to proceed in:
  - Maintaining stable clustering structure and offering a better performance in terms of the number of re-affiliations using the proposed algorithm ES-WCA (Energy Efficient and Safe Weighted Clustering Algorithm).
  - Detecting common routing problems and attacks in clustered WSNs, based on behavior level.
  - Showing clearly the interest of the routing protocols in terms of energy saving processes and therefore maximizing the lifetime of the global network.

This paper is organized as follows: in section 3, we emphasize on the security problems in WSNs. Section 4 introduces and explains the selected metrics for the proposed approach of clustering. A special attention is devoted to this last aspect in this research. More details on the proposed algorithm are provided in Section 5. Section 6 presents the developed platform for evaluation. PW results are provided to show the effectiveness of the proposed algorithm. Section 7 concludes the paper and outlines directions for future research.

2. Related Works

In this section, we outline some approaches of clustering used in ad-hoc networks and WSNs. Research studies on clustering in ad-hoc networks involve surveyed works on clustering algorithms[13, 23] and cluster head election algorithms [14, 24]. A single metric based on clustering as in paper [25] shows that the node with the least stability value is elected as CH among its neighbors. However, the choice of CH which has a lower energy level could quickly become a bottleneck of its cluster. Other proposals use a strategy based on computed weight in order to elect CHs [10-12, 17].

The main strategy of these algorithms is based mainly on adding more metrics such as the connectivity degree, mobility, residual energy and the distance of a node from its neighbors, corresponding to some performance in the process of electing CHs that have a greatest weight.

Although the algorithms which use this strategy allow to ensure the election of better CHs based only on their high computed weight from the considered metrics, they unfortunately do not ensure that the elected CHs are legitimated nodes, i.e., whether the election process of CHs is safe or not. In Section 3, we show that WSNs are vulnerable to various types of attacks [26]. In the last decade, several studies proposed solutions to solve attacks in WSNs by using cryptography, such as SPINS [28]. However, cryptography alone is not enough to prevent node compromise attacks and novel misbehavior in WSNs [29]. Khalil et al. [30] propose a protocol called DICAS which uses local monitoring and mitigates the attacks against control traffic by detecting, diagnosing and isolating the malicious nodes. Marti et al. [31] use a watch-dog technique or local monitoring for ad-hoc networks in order to improve the detection of mischievous nodes. They use a technique called path rater to help routing protocols to avoid them. A self-monitoring mechanism that pays more attention to the system-level fault diagnosis of the network was proposed by Hsin et al. [33], especially for detecting node failures. However, they did not deal with malicious behaviors.

Little effort has been made to include the security aspect in the clustering mechanism. Yu and Zhang [34] try to secure the clustering mechanism against wormhole attack in ad-hoc networks (communication between CHs). However, this is done after forming clusters, not during the election procedure of CHs. Liu [35] surveyed the clustering algorithms available for WSNs but that was done from the perspective of data routing.

In the context of these surveyed research works about clustering in both ad-hoc networks and WSNs, we situate our contribution in the approaches based on the computing of the weight of each node in the network as this approach focuses on a strategy of distributed resolution which introduces a new metric (the behavioral level metric) which promotes a safe choice of a cluster head in the sense that this last one will never be a malicious node. The monitor node watches its neighbors to know what each one of them does with the messages it receives from another neighbor. If the neighbor of the monitor changes, delays, replicates or simply keeps a message that should be retransmitted, the monitor signals a failure. None of the works mentioned above sought to give more importance to the election criteria of nodes responsible for monitoring the network. Moreover, WSNs include limited energy resources (batteries) due mainly to their small size. Our algorithm shows clearly the interest of the routing protocols in energy saving which therefore maximize the lifetime of the network by coupling it with AODV then DSDV protocols [36, 37].

3. Security in WSNs

Wireless sensor networks are susceptible to multiple types of attacks because they are randomly deployed in open and unprotected environments [38-40]. For securing WSNs, it is necessary to address the potential attacks on such networks. These can be classified as either passive attacks or active ones [41, 42]. It is known that routing protocols in sensor networks are simpler and more vulnerable to attacks than the other two types of wireless networks: ad-hoc and cellular. The first serious discussion and analyses on secure routing were performed by Karlof and Wagner [43]. They studied multiple types of attacks on routing protocols in detail and the effects on common routing protocols in WSNs. The assumption is that there are two types of attacks, outer attacks and inner attacks [1, 2]. Other researchers have used a decentralized approach to monitor network nodes with fault detections through

the coordination of neighboring nodes [44] or the use of watchdogs to detect misbehavior in neighbors [30, 31, 45, 46]. Da Silva et al. [38] adopted local monitoring between neighboring nodes. Among the studies that have been conducted in the related works, no research has intended to use a monitoring mechanism with a cluster-based architecture except scheme in [56]. However authors focused only on the misbehavior of malicious nodes and not on the nature of attacks. We thus propose a mechanism that assures the distributed monitoring of WSNs security issues. This mechanism uses a cluster-based architecture together with a new set of metrics and rules for diagnosing the state of the sensors. Reducing the flow of communication and providing a stable surveillance environment are the most significant advantages of this solution. In this paper, we only examine inner attacks and more precisely active attacks. Outer attacks are prevented by the use of link layer security mechanisms [47].

In our current work the focus is on the misbehavior of malicious nodes and the nature of attacks. In the following section, we review the most common network layer attacks on WSNs we selected and we highlight the characteristics of these attacks [1, 2, 43, 48].

3.1 Sinkhole

For this type of attack, almost the totality of the traffic from a particular area is redirected via a malicious node. Consequently this creates a metaphorical sinkhole [43, 49]. The laptop-class adversary may use higher computational resources and communication power than a legitimate node to advertise itself as the shortest path to the base-station or, in our case, to the CH. A CH aggregates the data of member nodes in a cluster and relays them to another CH or the sink node [1, 2].

3.2 Black Hole

In a black hole attack, an attacker stops sending the entering packages of his/her nodes to which he/she is connected, in order to remain unperceived. This preserves the sending of the auto-generated packages and thus, the malevolent node may seem normal with other nodes. This makes it difficult for the sink to detect the cause of why certain nodes logged out at the base.

3.3 Hello Flood Attack

Many routing protocols use 'hello' broadcast messages to announce themselves to their neighbor nodes. The nodes that receive this message assume that source nodes are within range and add source nodes to their neighbor list. The laptop-class adversary can spoof 'hello' messages with sufficient transmission power to convince a group of nodes that it is its neighbor [1, 2, 49].

3.4 Node Outage

If a node acts as an intermediary, an aggregation point, or a cluster head, what happens if the node stops working? Protocols used by the WSNs must be robust enough to mitigate the effects of failures by providing alternate routes [50].

4. Metrics for CHs Election

This section introduces the different metrics used for CHs election by focusing on the behavior level metric.

4.1 The Behavior Level of Node (n_i (BL_i))

The behavioral level of a node is a key metric in our contribution. Initially, each node is assigned an equal static behavior level 'BL_i=1'. However, the anomaly detection algorithm can decrease this level if a node misbehaves. For computing the behavior level of each node, nodes with a behavior level that is less than the threshold behavior then they will not be accepted as CH candidates, even if they have other interesting characteristics, such as high energy, a high degree of connectivity, or low mobility. However, abnormal nodes and suspect nodes may belong to a cluster as a CM, but never as a CH. So, we define the behavior level of each sensor node n_i, noted BL_i, in any neighborhood of the network as follows. BL_i is classified by the following mapping function:

$$\mathrm{Mp}(\mathrm{BL_i}) = \begin{cases} Normalnode: & 0.8 \le BL_i \le 1 \\ Abnormalnode: & 0.5 \le BL_i < 0.8 \\ Suspectnode: & 0.3 \le BL_i < 0.5 \\ Maliciousnode: & 0 \le BL_i < 0.3 \end{cases} \quad (1)$$

The values in Formula (1) are chosen on the basis of several reputed models of WSNs that have been adopted by numerous researchers like Shaikh et al. [51] and Hai et al. [1].

4.2 The Mobility of Node (n_i (M_i))

Our objective is to have stable clusters. So, we have to elect nodes with low relative mobility as CHs. To characterize the instantaneous nodal mobility, we use a simple heuristic mechanism as presented in the formula below (2) [52, 53]:

$$M_i = \frac{1}{T}\sum_{t=1}^{T}\sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2} \quad (2)$$

Where $(x_t, y_t)$and $(x_{t-1}, y_{t-1})$are the coordinates of node $n_i$at time t and t-1, respectively. T is the period for which this parameter is estimated.

4.3 The Distance between Node n_i and its Neighbors (D_i)

This is likely to reduce node detachments and enhance cluster stability. For each node i, we compute the sum of the distance D_i with all of its neighbors j.
This distance is given, as in [10, 11], by:

$$D_i = \sum_{j \in N(i)}\{dist(i,j)\} \quad (3)$$

4.4 The Residual Energy of Node n_i(Er_i)

After transmitting a message of k bits at distance d from the receiver, this energy is calculated according to [54]:

$$Er_i = E - \left(E_{Tx}(k,d) + E_{Rxelec}(k)\right) \quad (4)$$

Where:

- E: The node's current energy;
- $E_{Tx}(k,d) = k.E_{elec} + k.E_{amp}.d^2$: refers to the required energy to send a message; where $E_{amp}$is the required amplifier energy.
- $E_{Rxelec}(k) = kE_{elec}$: refers to the energy consumed while receiving a message.

4.5 The Degree of Connectivity of Node n_i at Time $t(C_i)$

It represents the number of n_i ś neighbors given by Eq. (5), according to [10, 32]:

$$C_i = |N(i)| \quad (5)$$

With: $N(i) = \{n_i / dist(i,j) < tx_{range} with i \neq j\}$
Where:

- $dist(i,j)$: outdistance separating two nodes n_i and n_j
- tx_{range} : the transmission radius.

For each node, we must calculate its weight P_i, according to the equation:

$$P_i = w_1 * BL_i + w_2 * Er_i + w_3 * M_i + w_4 * C_i + w_5 * D_i \quad (6)$$

Where $w_1, w_2, w_3, w_4,$ and w_5 are the coefficients corresponding to the system criteria, so that:

$$w_1 + w_2 + w_3 + w_4 + w_5 = 1 \quad (7)$$

We propose to generate homogeneous clusters whose size lies between two thresholds: $Thresh_{Upper}$ and $Thresh_{Lower}$. These thresholds are arbitrarily selected or they depend on the topology of the network. Thus, if their values depend on the topology of the network, they are calculated as follows, according to [55]:

- u: the node that has the maximum number of neighbors with one jump:

$$\delta_{12}(u) = \min(\delta_{12}(u_i): \ u_i \in U) \quad (8)$$

- v: the node that has the minimal number of neighbors with one jump:

$$\delta_{12}(v) = \min(\delta_{12}(v_i): \ v_i \in U) \quad (9)$$

We denote AVG as the average cardinal of the groups with one jump of all the nodes of the network:

$$AVG = \frac{\sum_{i=1}^{n} \delta_{12}(u_i)}{N} \quad (10)$$

Where: N represents the number of nodes in the network. Thus, the two thresholds are calculated as:

$$Thresh_{Upper} = \frac{1}{2}(\delta_{12}(u) + AVG) \quad (11)$$

$$Thresh_{Lower} = \frac{1}{2}(\delta_{12}(v) + AVG) \quad (12)$$

The calculated weight for each sensor is based on the above parameters ($BL_i, M_i, D_i Er_i$, and C_i). The values of coefficients w_i should be chosen depending on the basis of the importance of each metric in considered WSNs applications. For instance, we can assign a greater value to the metric BL_i compared to other metrics if we promote the safety aspect in the clustering mechanism. We can also assign the

same value for each coefficient w_i in the case where all metrics are considered as having the same importance. An approach based on these weight types will enable us to build a self-organizing algorithm that is able to form a small number of homogenous clusters in size and radius by geographically grouping close nodes. The resulting weighted clustering algorithm reduces energy consumption and guarantees the choice of legitimate CHs.

5. Weighted Clustering Algorithm (ES-WCA)

In this section, we first present some assumptions about our proposed ES-WCA. Then we present ES-WCA in detail, followed by an illustrative example to demonstrate our weighted clustering algorithm with the help of Figs. 4-6.

5.1 Assumptions

This paper is based on the following assumptions:

- The network formed by the nodes and the links can be represented by an undirected graph G = (U, E), where *U* represents the set of nodes n_i and *E* represents the set of links e_i [10].
- All sensor nodes are deployed randomly in a 2-dimensional (2D) plane.
- A node interacts with its one-hop neighbors directly and with other nodes via intermediate nodes using multi-hop packet forwarding based on a routing protocol, such as an ad-hoc on-demand distance vector [36] or DSDV [37].
- The radio coverage of sensor nodes is a circular region centered on this node with radius *R*.
- Two sensor nodes cannot be deployed in exactly the same position (x, y) in a 2D space.
- All sensor nodes are identical or homogeneous. For example, they have the same radio coverage radius *R*;
- Each node can determine its position at any moment in a 2D space.
- Each cluster is monitored by only one CH;
- Each CM communicates directly with its CH for the transmission of security metrics.
- A CH communicates directly with the base station for the transmission of security information and alerts.

5.2 Proposed Algorithm

The ES-WCA that we present below is based on the ideas proposed by Chatterjee et al. [10], Lehsaini et al. [55] and Zabian et al. [12], with modifications made for our application. This algorithm runs in three phases: the Set up phase, the Re-affiliation phase, and the Monitoring phase.

5.2.1 The Set up Phase

ES-WCA uses three types of messages in the set up phase. The message '**CHmsg**' is sent in the

network by the sensor node that has the greatest weigh. The second one is the '**JOINmsg**' message, which is sent by the neighbor of CH if it wants to join this cluster. Finally, a CH must send a response '**ACCEPTmsg**' message, as shown in Fig. 2 [27].

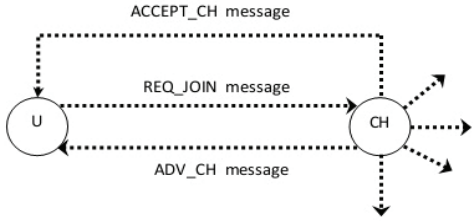


Fig. 2. Procedure of affiliation of node 'U' to a cluster [27].

Algorithm 1: Set Up Phase Algorithm

Begin

1: Assign values to the coefficients w_1, w_2, w_3, w_4, w_5;
2: **For** any node $n_i \in G$ **make**:
3: n_i forms a list of its neighbors $N(i)$ through the Message {who_are_neighbors};
4: $N(i) = \emptyset$;
5: Calculate its weight P_i:
6: $P_i = w_1 * BL_i + w_2 * Er_i + w_3 * M_i + w_4 * C_i + w_5 * D_i$;
7: Initialize Time Cluster and the state vector of all nodes $n_i \in G$ Vector_State (Id, CH, Weight, List_Neighbors, Size, Nature)
8: CH = 0, Size = 0;
9: Nature =" None";
10: **Repeat**
11: Any node $n_i \in G$ Broadcasts a message" Hello";
12: **If** $N(i) <> \emptyset$ **Then**
13: Choose $v \in N(i)$;
14: $Weight(v) = \max\{weight(w) / w \in N(i)\}$;

15: the node that have the same maximum weight, the CH is the node that has the best criteria ordered by their importance (BL_i, Er_i, C_i, D_i and M_i) if all criteria of nodes are equal, the choice is random.

15: **Else** n_i is a CH of itself.
EndIf
16: Update the state vector of the elected CH;
17: CH = ID;
18: Size = 1;
19: Nature = CH;
20: Send the message "CHmsg" by CH to its neighbors $N(\mathrm{CH})$;
21: J = Count ($N(\mathrm{CH})$);
22: **For** I = 1 to J **Do**
23: **If** ($n_i \in N(\mathrm{CH})$ receives the message && $n_i \rightarrow$ CH = 0)
24: **Then** n_i sends a message "JOINmsg" to CH
25: **If** (CH → Size < $Thresh_{Upper}$)
26: **Then** CH sends a message "ACCEPTmsg" to Node n_i;
27: CH executes the accession process;

28: CH → Size = CH → Size + 1;
29: n_i executes the accession process;
30: n_i→ CH = CH →Id;
31: **Else** go to 10;
EndIf
EndIf
End For
32: **Until expired** (TimeCluster);
End

5.2.2 The Re-affiliation phase

ES-WCA uses four types of messages in the Re-affiliation phase. The message '**RE_AFF_CH**' is sent in the network by the CH whose cluster size is less than $Thresh_{Upper}$. The second one is the '**REQ_RE_AFF**' message, which is sent by the neighbors of CH if it wants to join this cluster. Finally a CH must send a response '**ACCEPT_RE_AFF**' message or '**DROP_AFF'** message, as illustrated by Fig. 3.

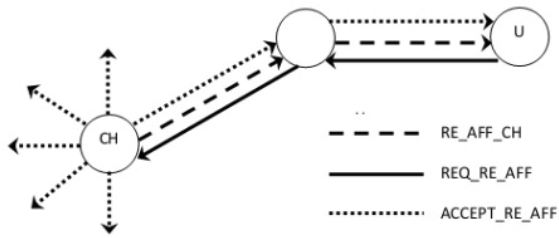


Fig. 3. Procedure of re-affiliation of node 'U' to a cluster.

Algorithm 2: Re-affiliation Phase Algorithm

Inputs: $Thresh_{Upper}$, $Thresh_{Lower}$;
Outputs: set of clusters
Begin
1:For num_cl = 1 to Count (Cluster) **Do**
2:If (Size (Cluster [num_cl]) <$Thresh_{Upper}$)
Then
3: CH sends a message "RE_AFF_CH" to its neighbors (N(CH));
4: J = Count (N(CH));
EndIf

5:For I = 1 to J **Do**
6:If ($n_i \in N$(CH) receives the message)
&& ($n_i \in$(Size(Cluster[num_cl]) <$Thresh_{Lower}$)
Then
7: n_i sends a Select message "REQ_RE_AFF" to the CH;
8: **If** (Size (Cluster [num_cl]) <$Thresh_{Upper}$)
Then
9: CH sends a message "ACCEPT_RE_AFF" to n_i;
10: CH updates its state vector;
11: CH → CH → Size = Size + 1;
12: n_i updates its state vector;

13: n_i → CH → ID = ID;
14: **Else** CH sends a "FIN_ AFF" message to n_i;
15: Go to 2;
EndIF
16:Else n_i sends a "DROP_AFF" message to CH;
EndIf
End For
End For
End

Our set up phase algorithm is demonstrated with the help of three figures (Figs. 4–6). Table 1 shows the values of the different criteria for the nodes that have behavior BL_i>0.8 (Normal nodes). Table 2 shows the weights P_i of neighbors for each node that have behavior BL_i>0.8. Nodes in Fig. 4 are presented by circles containing their identity IDs at the top and the levels of behavior at the bottom. According to Table 2, node 1 has a choice between CH11 and CH8 (they have the same weight), but the behavior level of node 11 is greater than that of node 8 (BL_{11}>BL_8). So, node 1 will be attached to CH11. For the other nodes, we have various conditions. Node 4 declares itself as a CH. Node 5 will be attached to CH4. Node 6 declares itself as a CH, because it is an isolated node. Node 8 will be attached to CH4. Node 10 is connected to CH5, but node 5 is attached to CH4. Thus, node 10 declares itself as a CH. Node 11 declares itself as a CH. These results give us the representation shown in Fig. 5. Node 2 is connected to CH4 and CH10. Node 2 will be attached to CH4, because CH4 has the maximum weight (968.133). Node 3 is connected to CH4, which implies that node 3 will be attached to CH4. Node 7 is not connected to any CH, so node 7 declares itself as CH. Node 9 is connected to CH4, and then node 9 will be attached to CH4. Node 12 is not connected to any CH, which implies that node 12 declares itself as a CH.

Table 1. Values of the various criteria of normal nodes

Ids	BL_i	Er_i	C_i	D_i	M_i	P_i
1	0.86	3842.12	3	1.15	1.20	769.632
4	0.81	4832.54	5	2.30	0.30	968.133
5	0.88	4053.25	3	1.30	0.55	811.829
6	0.85	4620.43	0	0.00	0.20	924.361
8	0.81	4816.80	4	1.05	1.40	964.753
10	0.95	3650.25	2	0.55	0.10	730.805
11	0.91	4819.60	1	0.70	2.20	964.753

Table 2. Weights of neighbors

Ids	1	4	5	6	8	10	11
1	769.632	-	-	-	964.753	-	964.753
4	-	968.133	811.829	-	964.753	-	-
5	-	968.133	811.829	-	-	730.805	-
6	-	-	-	924.361	-	-	-
8	769.632	-	-	-	964.753	-	-
10	-	968.133	811.829	-	-	730.805	-
11	769.632	-	-	-	-	-	964.753

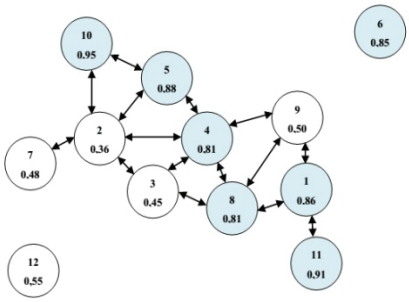


Fig. 4. Topology of the network.

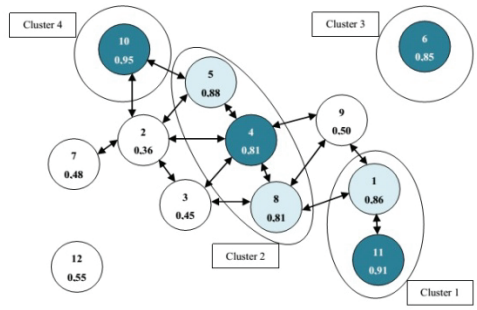


Fig. 5. Identification of clusters node.

At the end of this example, we obtained a network of six clusters (as shown in Fig. 6).

There are five situations that require the maintenance of clusters:

- Battery depletion of a node.
- Behavior level of a node less than or equal 0.3.
- Adding, moving, or deleting a node.

In all of these cases, if a node n_i is CH, then the set up phase will be repeated.

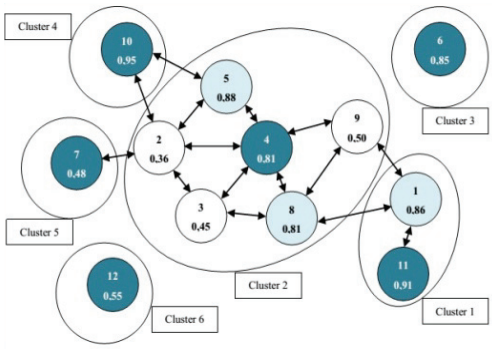


Fig. 6. The final identification of clusters.

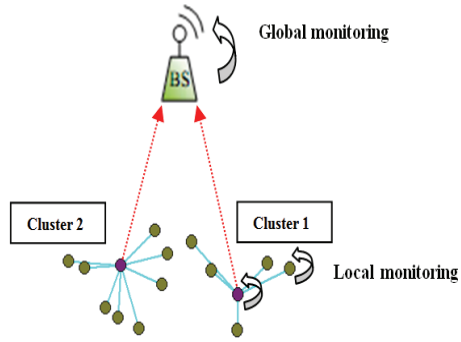


Fig. 7. Monitoring phase architecture.

5.2.3 The Monitoring phase

Monitoring in WSNs can be both local and global. The local monitoring can be with respect to a node and the global monitoring can be with respect to the network, but in sensor networks, the local monitoring is insufficient for detecting some types of errors and security anomalies [32]. For this reason, we adopted a hybrid approach that is global monitoring based on distributed local monitoring. The general architecture of our approach is illustrated in Fig. 7. 'Mercury' detects the internal misbehavior nodes during distributed monitoring process in WSNs by following up on the messages exchanged between the nodes. We assume that the network already has a prevention mechanism to avoid the external attacks. All the received messages are analyzed by using a set of rules. A similar approach is used by Da Silva et al. [38] and Benahmed et al. [56].

Algorithm 3 : Monitoring Phase Algorithm

Step1: This step runs in each CH_i:
Each CH_i becomes the monitor node of its cluster members and broadcasts a 'Start Monitoring' message with its Id_i to its entire cluster of CMs.

Step2: Calculation of security metrics performed by each member n_i of the cluster i.
Each node n_i (i<> j) receives the message "START MONITORING" and calculates its security metrics as follows:

- Number of packets sent by$n_i$at time interval $\Delta t = [t_0, t]$: $Nbp\_Send(ni, \Delta t)$.
- Number of packets received by noden_i at time interval$\Delta t = [t_0, t_0]$: $Nbp\_Received(n_i, \Delta t)$.
- Delay between the arrivals of two consecutive packets:

$$Delay\_BP(n_i, t) = Arrival\_PT_i - Arrival\_PT_{i-1} \quad (13)$$

- Energy consumption: the energy consumed by the node j in receiving and sending packets is measured using the following equation:

$$Ec(n_i, \Delta t) = Er(n_i, t_0) - Er(n_i, t_1) \quad (14)$$

Where: Δt is the time interval $[t_0, t_1]$; $Er(n_i, t_0)$ is the residual energy of node n_i at time t_0 ; $Er(n_i, t_1)$ is the residual energy of node n_i at time t_1 and $Ec(n_i, \Delta t)$ is the energy consumption of node n_i at time interval Δt.

Step3: Sending all metrics to the CH.
After each consumption of the security metrics, the state of a noden_i at time *t* is denoted as state(n_i, t_i). For storage volume economy, each node only keeps the latest calculation state.

- In the initial deployment of nodes, each CM in cluster 'i' sends some states (state(n_i, t_i)) to the CH_i for making a normal behavior model of node n_i by using a learning mechanism.
- Each state contains the following information:

$$(Id, Nbp\_Send(ni, \Delta t), Nbp\_Received(n_i, \Delta t), Delay\_BP(n_i, t), Ec(n_i, \Delta t)).$$

- If:(state(n_i, t_i) – state(n_i, t_{i-1})> ϵ)
  then: node $n_i$sends a message (ϵ a given threshold):

$$Msg=(Id, Nbp\_Send(ni, \Delta t), Nbp\_Received(n_i, \Delta t), Delay\_BP(n_i, t), Ec(n_i, \Delta t))$$

  to its CH_i for monitoring purposes.
  Otherwise, no information is sent to the CH.
- The message received by CH_i will be stored in a table Tmet, for future analysis.
- If a sensor node n_i does not respond during this monitoring period, it will be considered as misbehaving. The behavior level of sensor node n_i is computed using the following equation:

$$BL_i = BL_i - \text{rate} \quad (15)$$

  The 'rate' is fixed on the basis of the nature of the application. For example, if it is fault tolerant or not. In our case, we took: rate=0.1.

Step4: Misbehavior detection, which is performed by CH_i.

- For each node n_i in the cluster 'i', the state in time slot '*t*' is expressed by the three-dimensional vector:

$$S = (S_{t1}, S_{t2}, S_{t3})$$

  Where:
  - S_{t1} is the number of packets dropped by n_i, defined as follows:

$$S_{t1} = \sum Ps_{Received_{byn_i}} - \sum Ps_{Sent_{byn_i}} - \sum Ps_{destined_{byn_i}} \quad (16)$$

  With:

$$\sum Ps_{Received_{byn_i}} = \sum Ps_{Sent_{byn_i}} + \sum Ps_{destined_{byn_i}} + \sum Ps_{lost_{byn_i}} \quad (17)$$

For a normal node, of:$S_{t1} \approx 0$.

- S_{t2} is the delay between the arrival of two consecutive packets:

$$S_{t2} = Delay\_BP(n_i, t)$$

- S_{t3} is the energy consumption:

$$S_{t3} = Ec(n_i, \Delta t)$$

Here: $t \in [t_0, t] = \Delta t$;

- In our case, the first interval is used for the training data set of n time slots.

We calculated the mean vector $\bar{S}$ of S by using (18).

$$\bar{S} = \frac{\sum_{t=t_0}^{t_{n-1}} S_t}{n} \quad (18)$$

- After modeling a normal behavior model for each sensor node, the behaviors of all nodes are sent to the base station for further analysis. We then computed the deviation $d(S)$ by using Eq. (19).

$$d(S) = |S - \bar{S}| \quad (19)$$

- When the deviation $d(S)$ is larger than threshold T_h, which means that it is out of the range of normal behavior, it will be judged as a misbehaving node. In this case, the level of behavior is $BL_i \approx 0$. This is called the punishing algorithm.

$$\begin{cases} d(S) > T_h: n_i \text{ is an abnormal node} \\ d(S) \leq T_h: n_i \text{ Is a normal node.} \end{cases} \quad (20)$$

Algorithm 4: Punishing Algorithm

Begin
1:I:=0;
2: I: = I+1;
3:If ((I = Rate) && (BL_i<=0.1))
// Rate: parameter of maximum number of faults
defined by the administrator.
4: $BL_i = BL_i$ - Rate;
// Classification of the node according to its BL_i
5: $\mathrm{Mp(BL_i)} = \begin{cases} \text{Normal node:} & 0.8 \leq \mathrm{BL_i} < 1 \\ \text{Abnormal node:} & 0.5 \leq \mathrm{BL_i} < 0.8 \\ \text{Suspect node:} & 0.3 \leq \mathrm{BL_i} < 0.5 \\ \text{Malicious node:} & 0 \leq \mathrm{BL_i} < 0.3 \end{cases}$
6:If (BL_i≤0.3) **Then**
7:If (n_i is CM) **Then**
8: Suppression of the node of the list of the members;
9: Addition of the node to the blacklist;
EndIf
10:If (n_i is CH) **Then**// CH: Cluster Head
11: Addition of the node to the blacklist;
12: Set up Phase;
EndIf
EndIf
EndIf
End.

6. Implementation and Results

In this section, we present the implementation of our platform using the C Borland language.

6.1 The Virtual Laboratory 'Mercury'

We attempted to complete the theoretical study by implementing our own wireless sensor network virtual laboratory 'Mercury'. It is based on an object-oriented design and a distributed approach, such as a self-organization mechanism, which is distributed at the level of each sensor. Knowing that the sensors are too expensive and not available, we developed 'Mercury' to simulate network partitioning into a number of clusters that are more homogeneous in a combination of metrics to produce a virtual topology. To determine and evaluate the results of the execution of the algorithms introduced above, the number of sensors (N) to deploy must be less or equal to 1,000. There are two types of sensor node deployments in the sensor field, which are random and manual. The laboratory 'Mercury' offers the possibility to the learners to select a type of sense from five predefined types (see Table 3). Each type has its characteristics (radius, energy, etc.). The student can also introduce his/her own characteristics. The unity of the energy used is the nano-joule (1 Joule = 10^9 nJ).

Table 3. Different types of 'Mercury' sensors

Type	Range (m)	Energy (nJ)
1	75	0.2
2	100	0.4
3	125	0.6
4	150	0.8
5	175	1

6.2 Discussion and Results

In all the experiments, N varies between 10 and 1,000 sensor nodes. The transmission range (R) varies between 10 m and 175 m, and the used energy (E) is equal to 50,000 NJ. The sensor nodes are randomly distributed in a 570×555 m^2space by the following function:

```
for (intn = 0; n < node_tobe_deployed; n++)
        {
        X_ = rand ( )  %  image_Field_Of_Collecting-> width;
        Y_ = rand( )   %  image_Field_Of_Collecting-> Height;
        }
```

To measure the performance of the ES-WCA, we considered the following four metrics:

a. The number of clusters.
b. The number of re-affiliations.
c. The choice of ES-WCA with AODV or DSDV.
d. The detection of misbehavior nodes and the nature of attacks during the distributed monitoring process.

The values of weighting factors used for simulation were:

$$w_1 = 0.3,\ w_2 = 0.2,\ w_3 = 0.2,\ w_4 = 0.2 \text{ and } w_5 = 0.1$$

It is noted that these values are arbitrary at this time for this reason that they should be adjusted according to the system requirements. To evaluate the performance of the ES-WCA with other algorithms, we studied the effect of the density of the networks (number of sensor nodes in a given area) and the transmission range on the average number of formed clusters. Then we compared it with a DWCA proposed in [11] and WCA proposed in [10].

Fig. 8 shows the variation of the average number of clusters with respect to N. The results are shown for varying transmission range (R) between 75 m and 175 m. We observed that the number of clusters increases with the increase in the number of nodes. The experiments with a transmission range of 175 m gives the best result, where the average number of clusters is very close to its value in the interval 400–500. The number of clusters remains stable in the interval of 700–900 and is equal to 12. Fig. 9 illustrates the variation of the average number of clusters, with respect to the transmission range. The results are shown for N, which varies between 200 and 1,000. We observe that the number of clusters decreases with the increase in the transmission range. This is due to the fact that a CH with a large transmission range will cover a large area.

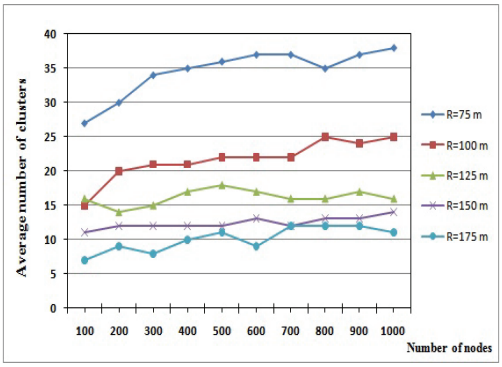


Fig. 8. Average number of clusters vs number of nodes.

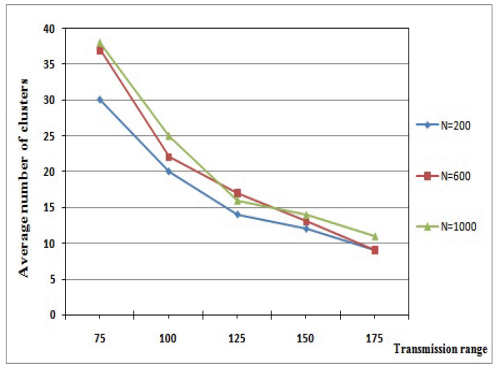


Fig. 9. Average number of clusters vs transmission range (R).

Fig. 10 depicts the average number of clusters that are formed, with respect to the total number of nodes in the network [27]. The communication range used in this experiment is 200 m. As shown in Fig. 10, the proposed algorithm produced the same number of clusters than DWCA when the node number is equal to 20 nodes. If the node density had increased, ES-WCA would have produced constantly less clusters than DWCA, regardless of the node number. The result of ES-WCA is unstable between 60 and 90 because we used a random deployment. So, if the distance between the nodes increases, the number of clusters increases too. When there were 100 nodes in the network, the proposed algorithm produced about 61.91% less clusters than DWCA [11]. As a result, our algorithm gave better performance, in terms of the number of clusters, when the node density in the network is high. This is due to the fact that ES-WCA generates a reduced number of balanced and homogeneous clusters, whose size lies between the two thresholds of: $Thresh_{Upper}$ and $Thresh_{Lower}$ (Re-affiliation phase) in order to minimize the energy consumption of the entire network and prolong the lifetime of the sensors. Fig. 11 shows the variation of the average number of clusters, with respect to the transmission range. The results are shown for varying N. We observed that the average number of clusters decreases with the increase in the transmission range. As shown in Fig. 11, the proposed

algorithm produced 16% to 35% fewer clusters than WCA when the transmission range of nodes was 10 m. When the node density increased, ES-WCA constantly produced less clusters than WCA, regardless of the node number. With 70 nodes in the network, the proposed algorithm produced about 47% to 73% less clusters than WCA. The results show that our algorithm gave a better performance, in terms of the number of clusters, when the node density and transmission range in the network are high.

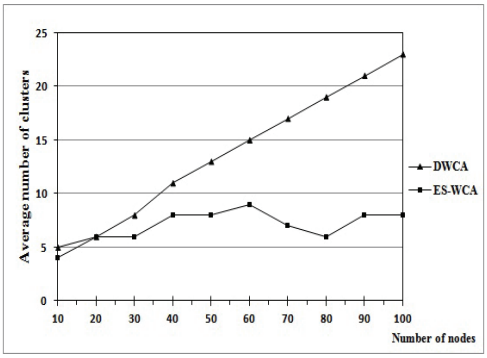


Fig. 10. Average number of clusters vs number nodes (N) for ES-WCA and DWCA [27].

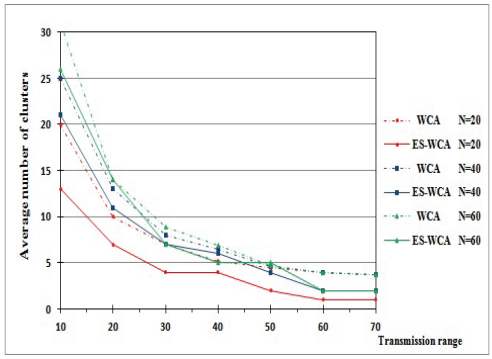


Fig. 11. Average number of clusters vs transmission range ES-WCA and WCA.

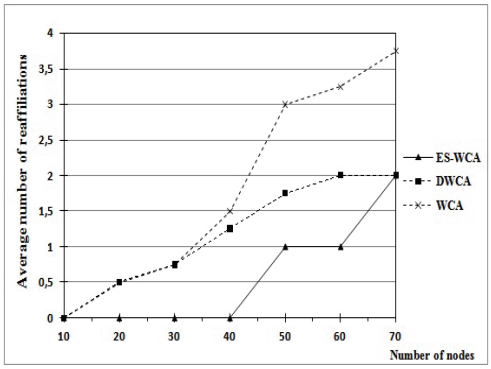


Fig. 12. Average number of re-affiliations.

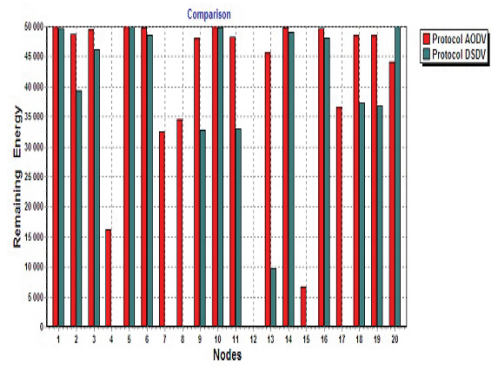


Fig. 13. Remaining energy per node using ES-WCA [27].

Fig. 12 depicts the average number of re-affiliations that are formed, with respect to the total number of nodes in the network. We propose to generate homogeneous clusters whose size lies between two thresholds: $Thresh_{Upper} = 18$ and $Thresh_{Lower} = 9$. The number of re-affiliations increased linearly when there were 30 or more nodes in the network for both WCA and DWCA. However, for our algorithm, the number of re-affiliations increased starting from 50 nodes. According to the results, our algorithm gave a better performance, in terms of the number of re-affiliations. The benefit of decreasing the number of re-affiliations mainly comes from the localized re-affiliation phase in our algorithm. The result of the remaining amount of energy per node for each protocol of AODV and DSDV is presented in Fig. 13, such as R is equal to 35 m [27]. As shown in the above-mentioned figure, the remaining energy for each node in the AODV protocol is greater than that in the DSDV protocol, such as AODV, which consumes 22.74% less than DSDV.

According to the results, the network consumes 19.23% of the total energy when we used an AODV

protocol (192,322,091 nJ). However, it consumes 41.97% with a DSDV protocol (419,740,129 nJ). We also observed that the network lost six nodes with DSDV, but only one node with AODV because of the depletion of its battery. This result clearly shows that AODV outperforms DSDV. This is due to the tremendous overhead incurred by DSDV when exchanging routing tables and because of the periodic exchange of the routing control packets. Consequently, our algorithm gave a better performance, in terms of saving energy when it is coupled with AODV.

In Fig. 14, we evaluate the lifetime of the network by varying the number of nodes, such as R being equal to 70m. We considered that the network will be invalid when the nodes in the neighborhood of the sink exhaust their energy, as illustrated in Fig. 15 with the color red. There are nine nodes in an active state, but the network is in valid. We observed that the increase in the number of nodes does not have a significant impact on the lifetime of network, except between N=60 and N=80. When there were 20 nodes in the network, the AODV increased the network duration by about 88.47% more than DSDV and about 57.9% for N=100. Also, this is due to the fact that in a DSDV protocol each node must have a global view of the network. This in turn increases the number of the exchanged control packets (overhead) in the whole network and it decreases the remaining energy of each node, which has a direct effect on the lifetime of the network.

For the experiment on abnormal behavior in the network, we generated 100 nodes with 5 malicious nodes. The states of the malicious nodes moved from a normal node (show in yellow) to an abnormal node (shown in blue), to a suspicious node (shown in gray), and finally, to a malicious node (shown in black). All of the states of the CMs are detected by their CH. Malicious CHs are detected by the base station.

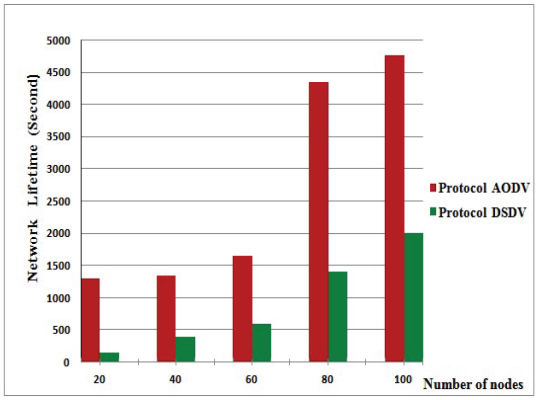


Fig. 14. Network lifetime depending on number of nodes using ES-WCA [27].

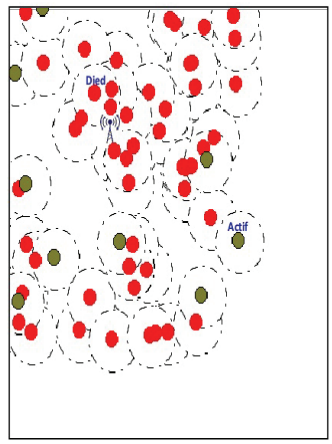
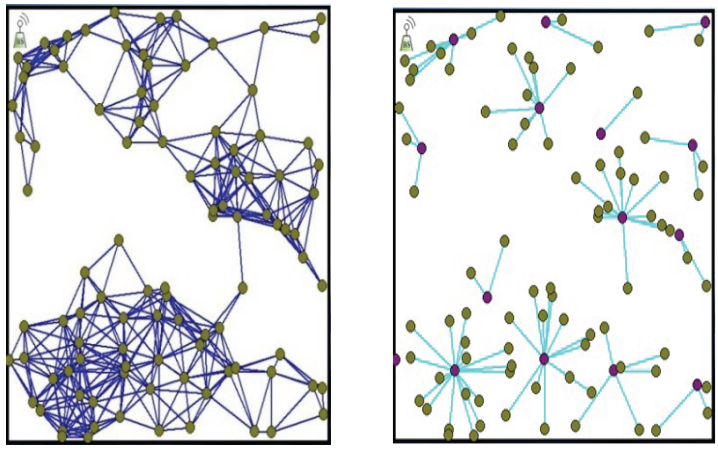


Fig. 15. Snapshot showing the neighborhood of the sink exhaust their energy (N=60, R=30 m).

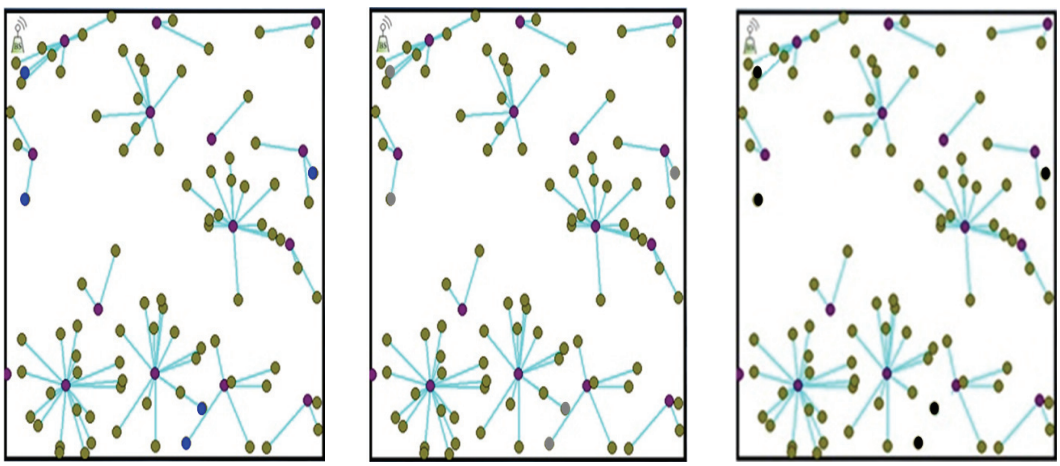
Fig. 16(b) shows the number of clusters formed according to the transmission range. Fig. 17(a)–(c) show the results of the experiment for a scenario with malicious nodes that are generated by the generator of bad behavior in Fig. 18. The generated attacks are described in Section 3.



(a) (b)

Fig. 16. (a) Graph connectivity of 100 nodes; (b) Network after clustering formation.

We can see that these nodes move from a normal state to an abnormal or suspicious state and finally, to a malicious states as expected. Table 4 shows the Ids of malicious nodes and their types of attacks during the distribution of a monitoring process in the network by the follow-up of the messages exchanged between the nodes. When Packets_sent (N_1,N_2), Packets_received (N_3,N_4). Thus N_1: is the number of packets sent before attacks, and N_2: is the number of packets sent after attacks. While N_3: is the number of packets received before attacks, and N_4: is the number of packets received after attacks. We see that these malicious nodes increases by N_1, as the sensors (16, 62), decrease by N_1, like the sensor (3), increases by N_3, as the sensor (41) and finally stop sending information like the node (94). We note from Fig. 19 that the sensor nodes (3, 16) are malicious and have a behavior level that is less than 0.3.



(a) (b) (c)

Fig. 17. (a) Sensors with a blue color are abnormal but not malicious. (b) The grey sensors have a suspect behavior. (c) The sensors with a black color are compromised and are exhibiting malicious behavior.

Table 4. Detection of the nature of attacks

Ids	Packets_sent	Packets_received	Attack
16	*(27,100)*	*(20,11)*	*Hello_flood*
3	*(13,1)*	*(5,8)*	*Black hole*
41	*(5,4)*	*(4,76)*	*Sinck hole*
94	*(5,4)*	*(4,4)*	*Node_outage*
62	*(26,115)*	*(21,6)*	*Hello flood*

We can see that these nodes move from a normal state to an abnormal or suspicious state and finally, to a malicious states as expected. Table 4 shows the Ids of malicious nodes and their types of attacks during the distribution of a monitoring process in the network by the follow-up of the messages exchanged between the nodes. When Packets_sent (N_1,N_2), Packets_received (N_3,N_4). Thus N_1: is the number of packets sent before attacks, and N_2: is the number of packets sent after attacks. While N_3: is the number of packets received before attacks, and N_4: is the number of packets received after attacks. We see that these malicious nodes increases by N_1, as the sensors (16, 62), decrease by N_1, like the sensor (3), increases by N_3, as the sensor (41) and finally stop sending information like the node (94). We note from Fig. 19 that the sensor nodes (3, 16) are malicious and have a behavior level that is less than 0.3.

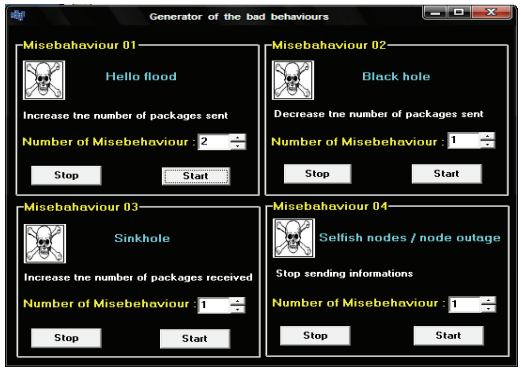


Fig. 18. Generator of the bad behaviors.

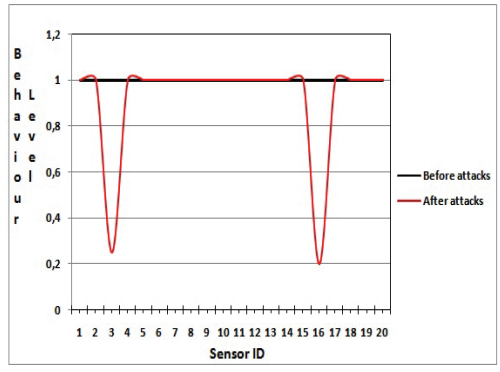


Fig. 19. Behavior level of some sensors before and after attacks.

7. Conclusions and Future Works

This paper presents a specification of our VLP 'Mercury'. It is based on an object-oriented design of a wireless sensor network PW using UML methodology. We proposed a new algorithm called ES-WCA for the self-organization of mobile sensor networks. The results obtained from simulations show that our algorithm outperforms WCA and DWCA in every sense. It yields a low number of clusters and it preserves the network structure better than WCA and DWCA by reducing the number of re-affiliations. The proposed algorithm chooses the most robust and safe CHs with the responsibility of monitoring the nodes in their clusters and maintaining clusters locally. Our third algorithm analyzes and detects specific misbehavior in the WSNs. The results show that in scenarios in which mobile WSNs with a low density or with a small size, the choice of ES-WCA with AODV is comparable to ES-WCA with DSDV

to clearly show the interest of the routing protocols in saving energy. However, the difference in favor between ES-WCA and AODV becomes very important in the case of a high node density. This is due to the tremendous overheads incurred by ES-WCA with DSDV when exchanging routing tables and exchanging routing control packets.

As a result of this work, we plan to explore further the concept of redundancy, in order to enhance results that are related to energy conservation. This can be done by using the 'sleep' and 'wake up' mechanisms in case of node failure due to a lack of energy, a software crash, or external attacks. In these cases, a redundant node is able to replace the failed node automatically. Moreover, we plan to provide in-network processing by aggregating correlated data in the routing protocol. Thus, the aggregator nodes can be identified in the network to receive multiple streams of correlated data. It can aggregate them (Average, Min, and Max) and then send a representative value, instead of the different flows. This will reduce both the energy consumption (the transmission of the representative value) and congestion. This considerably reduces the amount of data that is transported in the network. Another interesting avenue of exploration would be to use the same cluster-based architecture in order to examine passive attacks and to provide a stable and reliable surveillance environment.

References

[1] K. Wong, T. Wolf, S. Gorinsky, and J. Turner, "Teaching experiences with a virtual network laboratory," *ACM SIGCSE Bulletin*, vol. 39, no. 1, pp. 481-485, 2007.

[2] T. H. Hai, E. N. Huh, and M. Jo, "A lightweight intrusion detection framework for wireless sensor networks," Wireless Communications and Mobile Computing, vol. 10, no. 4, pp. 559-572, 2010.

[3] E. N. Huh and T. H. Hai, *Lightweight Intrusion Detection for Wireless Sensor Networks*. Rijeka, Croatia : INTECH Open Access Publisher, 2011.

[4] Omnet official site [Online], Available: http://www.omnetpp.org.

[5] Castalia official site [Online]. Available: https://castalia.forge.nicta.com.au/index.php/en/index.html.

[6] The Network Simulator (ns-2) [Online]. Available: http://www.isi.edu/nsnam/ns.

[7] OPNET official site [Online]. Available: http://www.riverbed.com/products/performance-management-control/opnet.html? redirect=opnet.

[8] Tcl SourceForge Project [Online]. Available: http://tcl.sourceforge.net/.

[9] D. Mechta, S. Harous, M. Djoudi, and A. Douar, "A collaborative learning environment for a biology practical work," in *Proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services (iiWAS2010)*, Paris, 2010, pp. 389-394.

[10] M. Chatterjee, S. K. Das, and D. Turgut, "WCA: a weighted clustering algorithm for mobile ad hoc networks," *Cluster Computing*, vol. 5, no. 2, pp. 193-204, 2002.

[11] W. Choi and M. Woo, "A distributed weighted clustering algorithm for mobile ad hoc networks," in *Proceedings of Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services (AICT-ICIW'06)*, Guadeloupe, French Caribbean, 2006, pp. 73-73.

[12] A. Zabian, A. Ibrahim, and F. Al-Kalani, "Dynamic head cluster election algorithm for clustered Ad-Hoc networks," *Journal of Computer Science*, vol. 4, no. 1, pp. 42-50, 2008.

[13] M. Chawla, J. Singhai, and J. L. Rana, "Clustering in mobile ad hoc networks: a review," *International Journal of Computer Science and Information Security*, vol. 8, no. 2, pp. 293-301, 2010.

[14] S. Mehta, P. Sharma, and K. Kotecha, "A survey on various cluster head election algorithms for MANET," in Proceedings of *2011 Nirma University International Conference on Engineering (NUiCONE)*, Ahmedabad,

India, 2011, pp. 1-6.

[15] H. Kim, "An efficient clustering scheme for data aggregation considering mobility in mobile wireless sensor networks," *International Journal of Control and Automation*, vol. 6, no. 1, pp. 221-234, 2013.

[16] M. Chatterjee, S. K. Das, and D. Turgut, "A weight based distributed clustering algorithm for mobile ad hoc networks," in Proceedings of the 7th International Conference on High Performance Computing (HiPC2000), Bangalore, India, 2000, pp. 511-521.

[17] R. Agarwal, R. Gupta, and M. Motwani, "Review of weighted clustering algorithms for mobile ad hoc networks," *Computer Science & Telecommunications*, vol. 33, no. 1, pp. 71-78, 2012.

[18] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Communications*, vol. 30, no. 14, pp. 2826-2841, 2007.

[19] K. A. Darabkh, S. S. Ismail, M. Al-Shurman, I. F. Jafar, E. Alkhader, and M. F. Al-Mistarihi, "Performance evaluation of selective and adaptive heads clustering algorithms over wireless sensor networks," *Journal of Network and Computer Applications*, vol. 35, no. 6, pp. 2068-2080, 2012.

[20] V. Geetha, P. V. Kallapur, and S. Tellajeera, "Clustering in wireless sensor networks: performance comparison of LEACH & LEACH-C protocols using NS2," *Procedia Technology*, vol. 4, pp. 163-170, 2012.

[21] Y. Wang, X. Wu, J. Wang, W. Liu, and W. Zheng, "An OVSF code based routing protocol for clustered wireless sensor networks," *International Journal of Future Generation Communication and Networking*, vol. 5, no. 3, pp. 117-128, 2012.

[22] E. Ekici, Y. Gu, and D. Bozdag, "Mobility-based communication in wireless sensor networks," *IEEE Communications Magazine*, vol. 44, no. 7, pp. 56-62, 2006.

[23] J. Y. Yu and P. H. J. Chong, "A survey of clustering schemes for mobile ad hoc networks," *IEEE Communications Surveys & Tutorials*, vol. 7, no. 1, pp. 32-48, 2005.

[24] B. Sun, C. Gui, Y. Song, and C. Hu, "Stable clusterhead selection algorithm for ad hoc networks," *International Journal of Future Generation Communication and Networking*, vol. 6, no. 3, pp. 95-105, 2013.

[25] I. I. Er and W. K. G. Seah, "Mobility-based d-hop clustering algorithm for mobile ad hoc networks," in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, Atlanta, GA, 2004, pp. 2359-2364.

[26] T. Kavitha and D. Sridharan, "Security vulnerabilities in wireless sensor networks: a survey," *Journal of Information Assurance and Security*, vol. 5, no. 1, pp. 31-44, 2010.

[27] A. Dahane, N. Berrached, and B. Kechar, "Energy Efficient and Safe Weighted Clustering Algorithm for Mobile Wireless Sensor Networks," *Procedia Computer Science*, vol. 34, pp. 63-70, 2014.

[28] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "SPINS: security protocols for sensor networks," *Wireless Networks*, vol. 8, no. 5, pp. 521-534, 2002.

[29] S. Ganeriwal and M. B. Srivastava, "Reputation-based framework for high integrity sensor networks," in *Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Network (SASN2004)*, Washington, DC, 2004, pp. 66-77.

[30] I. Khalil, S. Bagchi, and N. B. Shroff, "LITEWORP: a lightweight countermeasure for the wormhole attack in multihop wireless networks," in *Proceedings of International Conference on Dependable Systems and Networks (DSN2005)*, Yokohama, Japan, 2005, pp. 612-621.

[31] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom)*, Boston, MA, 2000, pp. 255-265.

[32] K. Benahmed, H. Haffaf, and M. Merabti, "Monitoring of wireless sensor networks," in *Sustainable Wireless Sensor Networks*, W. Seah and Y. K. Tan, Eds. Rijeka, Croatia: InTech, 2010.

[33] C. Hsin and M. Liu, "Self-monitoring of wireless sensor networks," *Computer Communications*, vol. 29, no. 4, pp. 462-476, 2006.

[34] Y. Yu and L. Zhang, "A secure clustering algorithm in mobile ad hoc networks," in *Proceedings of 2012 IACSIT Hong Kong Conferences (IPCSIT vol. 29)*, 2012, pp. 73-77.

[35] X. Liu, "A survey on clustering routing protocols in wireless sensor networks," *Sensors*, vol. 12, no. 8, pp. 11113-11153, 2012.

[36] S. Taneja and A. Kush, "A Survey of routing protocols in mobile ad hoc networks," *International Journal of Innovation, Management and Technology*, vol. 1, no. 3, 279-285, 2010.

[37] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," *ACM SIGCOMM Computer Communication Review*, vol. 24, no. 4, pp. 234-244, 1994.

[38] A. P. R. da Silva, M. H. Martins, B. P. Rocha, A. A. Loureiro, L. B. Ruiz, and H. C. Wong, "Decentralized intrusion detection in wireless sensor networks," in *Proceedings of the 1st ACM International Workshop on Quality of Service & Security in Wireless and Mobile Networks*, Montreal, Canada, 2005, pp. 16-23.

[39] M. S. Islam and S. A. Rahman, "Anomaly intrusion detection system in wireless sensor networks: security threats and existing approaches," *International Journal of Advanced Science and Technology*, vol. 36, no. 1, pp. 1-8, 2011.

[40] S. Marchesani, L. Pomante, M. Pugliese, and F. Santucci, "A middleware approach to provide security in IEEE 802.15. 4 wireless sensor networks," in *Proceedings of International Conference on Mobile Wireless Middleware, Operating Systems and Applications (Mobilware),* Bologna, Italy, 2013, pp. 85-93.

[41] H. Ghamgin, M. S. Akhgar, and M. T. Jafari, "Attacks in Wireless Sensor Network," *Journal of Applied Sciences Research*, vol. 7, no. 7, pp. 954-960, 2011.

[42] D. Martins and H. Guyennet, "Security in wireless sensor networks: a survey of attacks and countermeasures," *International Journal of Space-Based and Situated Computing*, vol. 1, no. 2, pp. 151-162, 2011.

[43] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," *Ad Hoc Networks*, vol. 1, no. 2, pp. 293-315, 2003.

[44] A. Sheth, C. Hartung, and R. Han, "A decentralized fault diagnosis system for wireless sensor networks," in *Proceedings of IEEE International Conference on Mobile Adhoc and Sensor Systems Conference*, Washington, DC, 2005.

[45] I. Khalil, S. Bagchi, N. F. Shroff, "MOBIWORP: mitigation of the wormhole attack in mobile multihop wireless networks," *Ad Hoc Networks*, vol. 6, no. 3, pp. 344-362, 2008.

[46] S. Kaplantzis, A. Shilton, N. Mani, and Y. A. Sekercioglu, "Detecting selective forwarding attacks in wireless sensor networks using support vector machines," in Proceedings of *3rd International Conference on Intelligent Sensors, Sensor Networks and Information (ISSNIP2007)*, Melbourne, Australia, 2007, pp. 335-340). IEEE.

[47] S. A. Camtepe and B. Yener, "Key distribution mechanisms for wireless sensor networks: a survey," Rensselaer Polytechnic Institute, Troy, New York, *Technical Report* 05-07, 2005.

[48] P. Berwal, "Security in wireless sensor networks: issues and challenges," *International Journal of Engineering and Innovative Technology*, vol. 3, no. 5, pp.192-198, 2013.

[49] W. Stallings, Cryptography and Network Security: Principles and Practices, 5th ed. Harlow: Pearson Education, 2010.

[50] A. S. K. Pathan, H. W. Lee, and C. S. Hong, "Security in wireless sensor networks: issues and challenges," in *Proceedings of the 8th International Conference Advanced Communication Technology (ICACT2006)*, Phoenix Park, Korea, 2006, pp. 1048-1054.

[51] R. A. Shaikh, H. Jameel, S. Lee, S. Rajput, and Y. J. Song, "Trust management problem in distributed wireless sensor networks," in *Proceedings of 12th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, Sydneyt, 2006, pp. 411-414.

[52] A. H. Hussein, A. O. Abu Salem, and S. Yousef, "A flexible weighted clustering algorithm based on battery power for Mobile Ad hoc Networks," in Proceedings of *IEEE International Symposium on Industrial Electronics (ISIE2008)*, Cambridge, UK, 2008, pp. 2102-2107.

[53] C. Li, Y. Wang, F. Huang, and D. Yang, "A novel enhanced weighted clustering algorithm for mobile networks," in *Proceedings of the 5th International Conference on Wireless Communications, Networking and Mobile Computing (WiCom'09)*, Beijing, China, 2009, pp. 1-4.

[54] S. Soro and W. B. Heinzelman, "Cluster head election techniques for coverage preservation in wireless sensor networks," *Ad Hoc Networks*, vol. 7, no. 5, pp. 955-972, 2009.

[55] M. Lehsaini, H. Guyennet, and M. Feham, "An efficient cluster-based self-organisation algorithm for wireless sensor networks," *International Journal of Sensor Networks*, vol. 7, no. 1, pp. 85-94, 2010.

[56] K. Benahmed, M. Merabti, and H. Haffaf, "Distributed monitoring for misbehaviour detection in wireless sensor networks," *Security and Communication Networks*, vol. 6, no. 4, pp. 388-400, 2013.

Amine Dahane http://orcid.org/0000-0001-6998-208X

He received Master degree in Computer Systems and Networks from the University of Bechar in 2011. Currently he is pursuing his Ph.D. degree in the Department of Electronics from the University of Sciences and Technology of Oran (USTO, Algeria) and member of LARESI Laboratory (Research in Intelligent Systems Laboratory) from 2011. His main research interest includes wireless sensor networks, their security, routing and management, Intrusion detection.

Nasr-Eddine Berrached

He received Dr. Eng. Degree in computer science from the Tokyo Institute of Technology (TIT, Japan) in 1992. He joined USTO in 1982, where he is professor at electronic department. From 1986, he was leading the laboratory of robotics and from 2000 he is leading the Research laboratory in intelligent systems (LARESI). His interest includes man-machine interface, telerobotics, machine vision, pattern recognition and inverse problem.

Abdelhamid Loukil http://orcid.org/0000-0003-4936-5599

He obtained Ph.D. degree at Paris12 University (France) in 1993. Currently, he is associate professor at Electronic Department (USTO, Algeria) and member of LARESI Laboratory (Research in Intelligent Systems Laboratory) where he leads the research team 'Mobile Robot and Artificial Vision'. His research interests focus on Robotics, Artificial Vision, Image Processing, Design of Human-Machine Interfaces (HMI), Virtual Reality and Augmented Reality.