Egyptian Informatics Journal 20 (2019) 179–191

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

Dynamic cluster algorithm for improving percolation of targets in a sensor network (DC-AIPT)

Mohamed Toumi [*], Abderrahim Maizate, Mohammed Ouzzif

RITM-ESTC/CED-ENSEM, University Hassan II, Casablanca, Morocco

ARTICLE INFO

Article history:
Received 2 March 2018
Revised 14 October 2018
Accepted 15 April 2019
Available online 20 May 2019

Keywords:
Clustering algorithm
Percolation
Wireless sensor networks
Target tracking
Dynamic clustering

ABSTRACT

A target tracking in wireless sensor networks consists of two main functions: The detection and the tracking of the target along its trajectory by means of sensors deployed in an area of interest. Generally, these sensors are not maintainable after deployments. Dynamic clustering algorithms seem to be an effective mechanism for increasing the network's lifetime. Indeed, this type of algorithms only activates the nodes that are on the trajectory of the target when the latter is at their reach. All other sensors must be in sleep mode. The effectiveness of a monitoring solution must take into account the quality of monitoring, connectivity, and the power consumption that are directly affected by the distribution and density of the nodes. We propose to construct optimal dynamic clusters on the target trajectory based on a probabilistic model integrating two fundamental parameters: energy consumption and accuracy. This last metric is evaluated, for the first time in the target tracking algorithms, by the notion percolation.

1. Introduction

Tracking by wireless sensor networks involves detecting and locating a target over its entire trajectory across an area of interest. Wireless sensor networks are used to retrieve information about these targets in sometimes hostile environments that are not always accessible to humans. That's why; it is considered that once they are deployed, the sensors are autonomous. The energy factor is therefore at the center of all researchers concerns in the field of wireless sensors.

Dynamic clustering algorithms appear to be promising mechanisms for increased energy efficiency because they only activate nodes that are on the target's trajectory when the target is at their reach. All other sensors must be in a sleep mode. The choice of this type of architecture is motivated, on one hand, by its efficiency of collaboration, processing and sharing of data and, on the other hand, by the ease of scaling. The coverage of the moving target is one of the most important problems to guarantee the quality of service in a wireless network. This coverage is closely linked to the preliminary deployment of the sensors in the area of interest.

The strategy of deploying sensor nodes in a control zone is a decisive phase that has a major impact on the operation and quality of service of the network.

The problem of deploying or positioning sensor nodes in a WSN is a strategy used to define the topology of the network, and therefore the number and position of the sensor nodes. Different problems are studied in the deployment of sensor nodes in a WSN.

Studies in [5–11], which will be presented in the related work, mainly concern the stationary and mobile cases, the mono and multi-objective cases, and finally the deterministic and probabilistic aspects. However; these deployment strategies cannot be adapted to hostile areas of interest that prevent the organized deployment of sensors. In addition, these sensors can also be deprived of the capacity of mobility in order to be able to execute a totally decentralized strategy of deployment. In such cases, the use of a random deployment strategy is mandatory. This network creates the challenge of the network connectivity's quality, which is significantly deteriorated with a poor distribution of sensor nodes in the surveillance zone, or even the loss of the network service's continuity. The problem to solve is how to have a better quality of the monitoring of the moving target with a random distribution of the sensor nodes in the field of interest. The quality of monitoring is measured for the first time by the notion of percolation.

[*] Corresponding author.
E-mail address: m_toumy@yahoo.fr (M. Toumi).

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

ELSEVIER Production and hosting by Elsevier

https://doi.org/10.1016/j.eij.2019.04.002

For the phenomenon of percolation to appear, two ingredients are necessary: on one hand a solid and hollow structure and on the other hand an agent capable of evolving in this structure. By similarity, we consider that the agent is the target in motion and the hollows are the structure of the clusters formed by the algorithm. In Fig. 1 it can be observed that it is not because a rock has a certain porosity that it necessarily leaves water to flow. This porosity must be well structured so that the holes (clusters) connect the top of the rock to the bottom of the rock. In other words, there must be a percolation. It is then said that there is a percolation when the agent who moves in the hollows of the structure is able to cross it. In the "no-percolation" regime, the target is lost a moment before proceeding to its detection again, it is thus manifested by cover holes in the detection field, These holes are not only usually caused by sensor node failures or by the random deployment of nodes that can leave entire areas uncovered, but also by the structure of the clusters formed by the algorithm during the displacement of the target. A poor design of these structures can create a regime of non-percolation even with sufficient node density as explained above by the porosity of soils. The "no-percolation" regime is therefore very expensive in terms of the accuracy of coverage of the moving target and the energy consumption of the WSN network, since the re-detection process is Very energy-intensive (Ref 2001). We propose a dynamic clustering algorithm for the target tracking that achieves energy efficiency and a better percolation rate regardless of the node deployment strategy. for the first time we adopt the evaluation metric "Percolation" to measure the monitoring quality of the target tracking algorithm in an area of interest with a random distribution of the sensors Fig. 2.

This article is organized as follows: in Section 2, we present related work on tracking of the targets.

Then in Section 3, we will detail the proposed protocol. In Section 4, we present the simulation results. Finally, we end this article with a conclusion in Section 5.

2. Related works

In target tracking applications, achieving high accuracy of tracking along with preserving the energy efficiency is considerably and essentially influenced by two parameters:

- The nature of the tracking algorithm: The quality of monitoring, connectivity and the power consumption that are directly affected by the nature of the tracking algorithm.

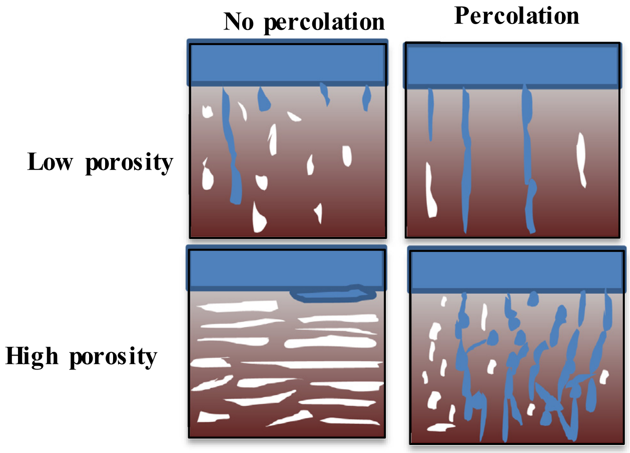

Fig. 1. The phenomenon of percolation.

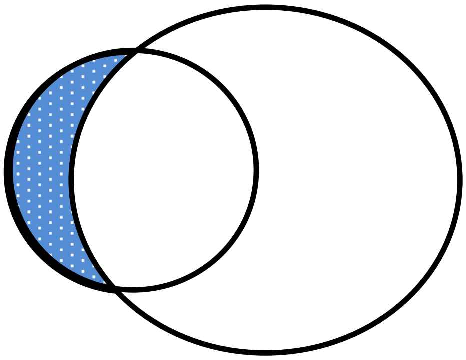

Fig. 2. The surface of the lunula.

- The deployment of the sensor nodes: The problem of deploying or positioning sensor nodes in a WSN is a strategy used to define the topology of the network and therefore the number and position of the sensor nodes.

We present in the following the recent contributions taking into account the two parameters mentioned above.

2.1. The nature of the tracking algorithm

Several classifications for monitoring algorithms and protocols have been proposed in the literature [1–6]. Four predominant categories:

Predictive solutions.

- Solutions based on a hierarchical architecture.
- Solutions based on dynamic cluster architecture.
- Hybrid Solutions

This classification takes into account both the type of the network architecture and the successive wake-up process of the sensors

2.1.1. Predictives solutions

In this category, the trajectory of the target is predicted in order to activate just the sensors on the way and thus save energy. This prediction can be made using predictive models among such as: Kalman filters [6–9], or using probabilistic mechanisms such as markov chains [10,11].

The Kalman filter (KF) is a prediction model with two stages: prediction and correction. It allows estimating recursively the process status based on its earlier statements, it aims to estimate the future target position based on the current position.

To get more accurate prediction, an Interactive Multiple Model (IMM) and Kalman filters (KFs) may be used [6]. IMM are suitable for nonlinear and non-Gaussian systems. KFs are linear algorithms that use a series of measurements observed over time, contain noise (random variations) and other inaccuracies, and produce estimates of unknown variables. Due to the non-linear motion of the target, the proposal in [7] uses an extended version of KF (EKF). One of the drawbacks of EKF is that the predicted state may seriously diverge from the actual state because of the EKF's disruptions during the linearization process.

Contribution [10] addresses the problem of efficiency efficient distributed detection of predicates capturing the motion trends of mobile objects evaluated with respect to a (boundary of a) polygonal region, in the settings in which the (location, time) data is obtained via tracking in Wireless Sensor Networks (WSN).

Predictive solutions have the advantage of making the best use of available information on both the target and the environment. However, they remain inadequate for certain critical contexts (military applications, intrusion detection, etc.) where even the momentary loss of the target is unthinkable.

2.1.2. Solutions based on a hierarchical architecture

In tree-based tracking approaches, the moving target is surrounded by sensor nodes. As the target moves, the tree is dynamically configured to add and remove some nodes. The root node of the spanning tree is chosen to be the closest node to the target's predicted location. In addition, the root collects the sensed data, computes all necessary information about the target, and ultimately sends it to the base station [15]. The dynamic lookahead spanning tree based tracking algorithm (DLSTA), proposed in [16], forms look ahead trees along the predicted target direction that aim to reduce the times of missing the target that moves very fast. In DLSTA, when the target is detected by some nodes, they collaborate with each other and select the closest node to the target as a root node.

The main disadvantage of solutions based on a hierarchical architecture is that the processing of the data is centralized at the level of the root node; this generates a not negligible cost of the information routing. Moreover, since the distance between the root node and the target increases, the rate at which the tree must be reconfigured also Increases. As a result, the tree structure is not efficient for tracking objects at high speed.

2.1.3. Solutions based on dynamic cluster architecture

The solution with dynamic clustering sensors tries to form different clusters at different times according to the movement of the target, so a node can be a member of different clusters at different time. Unless static clustering, dynamically it forms the cluster with sensors. Only one cluster is active at each time in accordance to target moving manner. Thus it reduces the energy consumption of sensors in the cluster. It is more reliable than static clustering method. Also the sensor nodes sensing and forwarding redundant information has been reduced. In [12], the authors have mentioned that Dynamic cluster-based approaches provide better bandwidth utilization and higher scalability than other approaches in target tracking.

2.1.4. The hybrid solutions

Hybrid solutions combine several approaches. As an example, a predictive mechanism (markov chain) implemented on a clustered network is proposed in [3*]. By analyzing the intrinsic relationship between tracking performance and energy consumption, the contribution cast the scheduling problem of WSN as the optimal policy problem of partially observable Markov decision process, and proposes a dynamic cluster members scheduling (DCMS) algorithm to solve the tradeoff between tracking performance and energy consumption. The authors of [13,14] propose a dynamic clustering algorithm coupled with the Kalman filter to predict the position of the single moving target. It is described using a state evolution model and a measurement model that assumes linear with Gaussian errors.

This type of solution makes it possible to make the best use of the advantages of the different schemes. The aim is always to have maximum energy efficiency.

2.2. The deployment of sensor nodes

*The s*ensor deployment is a critical issue since it reflects the cost and the surveillance capability of a wireless sensor network. A lot of research has been proposed for better node deployment efficiency.

Virtual force algorithms (VFA) are popular approaches for solving the problem of coverage and deployment of nodes. In this type of algorithm, the nodes are considered points subjected to a force of repulsion and attraction exerted between them and which can move as a function of the calculated force. In [15], the authors proposed a virtual force-directed co-evolutionary particle swarm optimization (VFCPSO) algorithm to maximize sensing coverage while minimizing the energy consumption in a hybrid wireless sensor network that consists of static and mobile sensors.

The research in [18,19] determined the densities of sensor nodes that achieve a desired area coverage based on Hall's asymptotic analysis. They defined the area coverage as the fraction of the geographical area and determined the minimum number of sensors to be deployed in the infinite plane using homogeneous Poisson point processes.

The authors in [20] present a genetic algorithm that seeks a solution to the problem of coverage holes in the network. The proposed algorithm determines the minimum number and the best locations of the mobile nodes that must be added after the initial deployment of the fixed nodes. The performance of the proposed genetic algorithm was evaluated by using several indicators; the simulation results show that this algorithm optimizes network coverage in terms of the overall coverage ratio and the number of additional mobile nodes.

In [21], the issue of estimating the number of sensors required to achieve a complete coverage for a desired region was studied. The coverage holes were introduced in [22] as a metric to evaluate the performance of deployment strategies with the presence of failures and placement errors in sensor networks.

In [23], the authors proposed a distributed algorithm that uses Voronoi diagrams in the Laguerre geometry to solve the deployment problem in heterogeneous mobile sensor networks, where sensing radii of sensors are not identical. A *Voronoi diagram* describes the spatial relationship between points that are near each other, or their nearest neighbours. It is a set of connection polygons derived from points or locations. Each line of a Voronoi "region" is halfway between two points. These deployment strategies have the disadvantage of being centralized approaches that necessarily require an organized deployment of sensors. Distributed deployment approaches are limited to the case where nodes do not have the mobility capability to execute a fully decentralized deployment strategy. It is important to have an algorithm capable of obtaining the best of the perforations with a random distribution of the nodes. In the following table, we propose a classification of target tracking algorithms in a way that frees itself from the network architecture and focuses on the following criteria:

a. The detection of the target
b. The location of the target
c. Activation mechanisms of the nodes
d. Support for the density of network nodes

Solution	Detection	Localisation	Activation mechanisms	the density of the nodes
Predictives solutions: Kalman filters [6–9],		✓	✓	

(continued on next page)

(*continued*)

Solution	Detection	Localisation	Activation mechanisms	the density of the nodes
Predictives solutions: Markov Chain [12,13]		✓	✓	
Solutions based on a hierarchical architecture. [15,16]	✓	✓		
Solutions based on dynamic cluster architecture. [14]	✓	✓	✓	
The Hybrid solutions [17]	✓	✓		✓
Virtual force algorithms (VFA) [18–20]			✓	✓

3. The proposed algorithm

3.1. Percolation concept

The percolation theory was developed to mathematically deal with the disordered media, in which the disorder is defined by a random variation in the *degree of connectivity* [24]. The main concept of percolation theory is the existence of a percolation threshold, defined in the following way. Suppose p is a parameter that defines the average degree of connectivity between various sub-units of some arbitrary system. When p = 0, all sub-units are totally isolated from every other sub-unit. When p = 1, all sub-units are connected to some maximum number of neighboring sub- units. At this point, the system is connected from one side to the other, since there are paths that go completely across the system, linking one sub-unit to the next along the spanning cluster. In the following some examples of systems using percolation:

- Coffee percolation, where the solvent is water, the permeable substance is coffee grounds, and the soluble constituents are the chemical compounds that give coffee its color, taste, and aroma
- Movement of weathered material down on a slope under the earth's surface
- Cracking of trees with the presence of two conditions, sunlight and under the influence of pressure
- Robustness of networks to random and targeted attacks
- Transport in porous media
- Epidemic spreading
- The spread of fire in a forest

The concept of percolation is therefore widely used in many applications; our algorithm will use this formulation for the first time to measure the accuracy of the target tracking in wireless sensor networks.

3.2. The system model and assumptions

Before heading into the technical details of our contribution, we first give some definitions and notations that will be used later in our paper.

A lune is a plane figure bounded by two circular arcs of unequal radii, i.e., a crescent. (By contrast, a plane figure bounded by two circular arcs of equal radius is known as a lens.) For circles of radius 'ɑ' and 'ß > ɑ' whose centers are separated by a distance C, the area of the lune is given by the Eq. (1) [25]:

$$A = a^2\left[\tan^{-1}\left(\tfrac{a^2-b^2+c^2}{4\Delta}\right)+\cos^{-1}\left(\tfrac{b-c}{a}\right)+\tan^{-1}\left(\tfrac{b-c}{\sqrt{(a+b-c)(a-b+c)}}\right)\right] - b^2\left[\tan^{-1}\left(\tfrac{a^2-b^2-c^2}{4\Delta}\right)+\tfrac{\pi}{2}\right]+2\Delta$$

$$= 2\Delta + a^2\sec^{-1}\left(\frac{2ac}{a^2-b^2-c^2}\right) - b^2\sec^{-1}\left(\frac{2bc}{b^2+c^2-a^2}\right)$$

where

$$\Delta = \frac{1}{4}\sqrt{(a+b+c)(b+c-a)(c+a-b)(a+b-c)} \tag{1}$$

In general, the signal received by a node Ni from a node Nj is decreased with the distance between the two nodes.

We are adopting the mitigated disk detection model [17] to estimate the distance d (Ni, Nj) from the received signal. The model equation is written as follows:

$$r_i = \begin{cases} \frac{\beta}{d^{\alpha}(N_i,N_j)} & \text{if } d(N_i,N_j) \leqslant r_s \\ 0 & else \end{cases} \tag{2}$$

$\mathbf{r_i}$: is the signal received from the node $\boldsymbol{N_j}$
$\boldsymbol{\beta}$: is the original strength of the signal transmitted by a node.
$\boldsymbol{\alpha}$: attenuation coefficient depending on the environment.
$\mathbf{r_s}$: is the detection range of the node.
$\boldsymbol{d}(\boldsymbol{N_i},\boldsymbol{N_j})$: is the Euclidean distance between the two nodes Ni and Nj

To facilitate the description of the protocol, we adopt the following notations:

i. R_t: The node communication range.
ii. r_s: The node detection radius.
iii. $R(n_i, r_s)$: the detection region of the node Ni with the detection ranger$_s$
iv. G: The wireless sensor Network **G = (V, E)**
v. E: The set of all links between nodes. The set E is defined by:
vi. $E = \{e = (v_i, v_j)| \{v_i, v_j \in V^2 \cap d(v_i,v_j) \leq R_t \cap i \neq j\}$
vii. **V:** The set of all Nodes: $V = \{\boldsymbol{v}_1, \boldsymbol{v}_2 ..., \boldsymbol{v}_n\}$
viii. The cluster $\boldsymbol{C_i}$ is defined by:
$\boldsymbol{C_i}$**:{**$\boldsymbol{v_j}$**|d (**$\boldsymbol{v_i},\boldsymbol{v_j}$**)**$\leqslant\boldsymbol{R_t}$**},** $\boldsymbol{v_i}$ is the Cluster Head Node.
ix. $\boldsymbol{L}_{(\boldsymbol{t})}$: The location of the target at time t

We assume that nodes are initially in the state of sleep which guarantees minimal energy consumption. Indeed, in this state, all the equipment units, which make up the sensor, are off, except for a processing unit and a low power channel for receiving the activation-up messages. Upon receiving a wake-up message, each node has to start all these hardware units.

It is also assumed that the nodes have knowledge of their geographical positions and the first target detection is done. This task is beyond the scope of this project.

The wireless sensor network is abstracted as an undirected graph $G = (V,E)$, where V represents the set of wireless nodes and $E \subseteq V^2$ is the set of edges that gives the available communications: an edge $e = (u,v)$ belongs to E if and only if u is physically able to transmit messages to v and vice versa. Each sensor $u \in V$ is assigned a unique value to be used as an identifier $Node_{id}(u)$ and all links in the graph are bidirectional. The neighbourhood set of a node u is represented by $N_1(u)$ as in equation (2). The size of this set is known as the degree of u, denoted by $\delta_1(u)$.

$$N_1(u) = \{v \in V \mid v \neq u \wedge (u, v) \in E\} \tag{3}$$

The set of 2-hop neighbors is represented by $N_2\,(u)$ as in equation (3).

$$N_2(u) = \{w \in V \mid (v, w) \in E \; where \; v \in N_1(u) \wedge \boldsymbol{w} \wedge \boldsymbol{w} \in N_1(v)\} \tag{4}$$

3.3. Enabling sleep nodes (Prediction)

In our contribution, we propose an algorithm that uses the activation of sensors located on the way to the target by over- listening nodes belonging to the active cluster. The activation process is described as follows:

Let $\boldsymbol{C_k}$**:{**$\boldsymbol{v_j}$**, d (**$\boldsymbol{CH}$**,**$\boldsymbol{v_j}$**)**$\leqslant$ $\boldsymbol{R_t}$ **}** be the set of the members of the active cluster, is defined

$$\boldsymbol{N_{Neig}} = \left\{\boldsymbol{v_i} \in \boldsymbol{V} \cap_{\boldsymbol{K}}^{\boldsymbol{C}} \mid \; \exists \boldsymbol{v_j} \in \; \boldsymbol{C_K} \;, \mathrm{d}\left(\boldsymbol{v_i}, \; \boldsymbol{v_j}\right) \leqslant \boldsymbol{R_t}\right\}$$

$\mathrm{N_{Neig}}$ is the set of all neighboring nodes to the active cluster $\mathrm{C_K}$.

$\boldsymbol{D_k} = \{\boldsymbol{v_j} \in \boldsymbol{C_k} | \exists\, \boldsymbol{t}_0, \mathbf{d}\,(\boldsymbol{L}_{(\boldsymbol{t}_0)}, \boldsymbol{v_j}) \leqslant \boldsymbol{r_s}\}$ is the set of the nodes belonging to the active cluster that detected the target at some point. It also defines the set of the nodes that will be activated by the prediction process based on there over-hearing messages by:

$$\boldsymbol{N_{Activ}} = \{\boldsymbol{v_i} \in \boldsymbol{V} \cap_{\boldsymbol{K}}^{\boldsymbol{C}} \exists \boldsymbol{v_j} \in \; \boldsymbol{D_K} \;, \; \mathrm{d}\;(\boldsymbol{v_i}, \boldsymbol{v_j}) \leqslant \boldsymbol{R_t}\}$$

It is clearly that $\mathrm{N_{Activ}} \sqsubset \mathrm{N_{Neig}}$ that is to say, the set of the nodes that will be activated is only a subset of the active cluster neighbors. This improves the energy efficiency by activating just the nodes that potentially will detect the target. In Fig. 3, the nodes in red form the set $\boldsymbol{D_k}$, by raising the collected data of the target; will create by over-listening the wake-up wave (The chopped part).

The green nodes that form the set $\boldsymbol{N_{Activ}}$ will be immediately activated since they are within the range of the wake-up wave. Prediction schemes have been proposed in recent years to predict the position of the target and thus to activate only the nodes that are on the trajectory of the target. The extended Kalman filter [8] combined with a direction change detection mechanism such as CuSum [17] make it possible to efficiently calculate the future coordinates of the target and accordingly to awaken the sensors.

The prediction filters require message exchanges and sometimes complex computations performed by a central entity, which is time-consuming. Moreover these algorithms use a correction step and lead to additional calculations and therefore a Precious time wasted needed to effectively respond to the real-time constraint imposed especially by the high-speed targets. This can be very penalizing in terms of a percolation of the target.

The proposed activation model is based on the overhearing of the active cluster member sensors, which makes it possible to wake the other nodes on the target route and increase the percolation of the target by having a high probability to get an active cluster waiting for him.

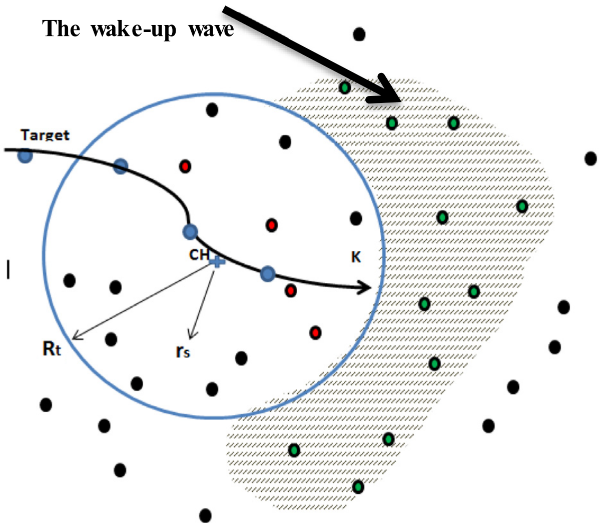

Fig. 3. The Activation of inactive nodes.

3.4. Selection of CHs

After the prediction phase described in Section 3 B, next comes the target tracking phase by constructing optimal clusters once the target detected. We start this paragraph by an estimation of the percolation in a cluster in D-1, after proposing the algorithm of the selection of CH and the construction of clusters in D-2.

3.4.1. The estimation of the percolation

The deployment of the sensor networks can follow regular or arbitrary topologies. In case of a random topology, several works, such as those proposed in [18,19], have shown that the random graphs according to a Poisson distribution offer a better representation of the deployment of nodes in a WSN.

In a network following the Poisson distribution with an intensity λ, each Node N has a number of random neighbors β taking the value k has a probability P such that:

$$\boldsymbol{P}(\boldsymbol{\beta} \;=\; \boldsymbol{k}) = \frac{\boldsymbol{\lambda^k}}{\boldsymbol{k}!}\boldsymbol{e}^{-\lambda} \tag{5}$$

In the Eq. 5, the parameter λ of the Poisson law represents the average degree of the network.

To estimate the percolation, two clustering schemes are defined to track the target:

1. A schema with overlap (Fig. 4.a)
2. A schema without overlap (Fig. 4.b)

We define the percolation ∂_i in a $\mathrm{Cl_i}$ cluster as:

$$\partial_i = \frac{\int_0^\tau (\boldsymbol{P}(\beta \geqslant 1))dt}{\tau} \tag{6}$$

$\boldsymbol{P}(\beta \geqslant 1)$: The probability that it has at least one node to cover the target.

τ : The average duration of the target's presence in the Cli cluster.

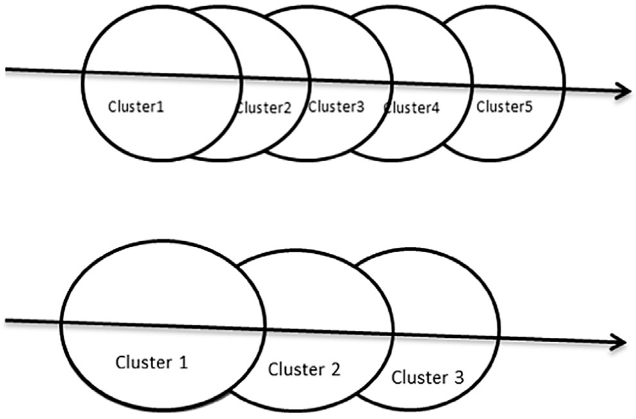

Fig. 4. Overlapping of the clusters.

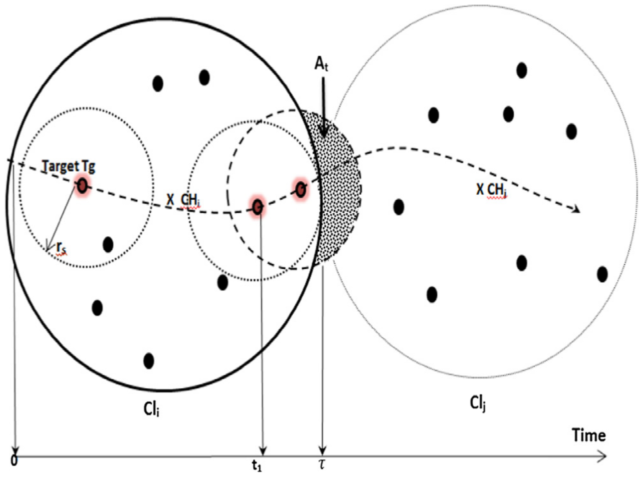

Fig. 5. The evolution of the target in a clustering scheme without overlap.

3.4.2. *Estimation of percolation in the case of a non-overlapping schema*

To estimate this percolation, consider the scheme of Fig. 5. The intensity of the Poisson law is autonomous when the target moves through the cluster CL_i, but once it reaches the border with the cluster CL_j, the calculation of the intensity must take into account that the cluster CL_j is not still active (Eq.9).

$$\partial_i = \frac{\int_0^\tau (1 - \boldsymbol{P}(\beta = 0))dt}{\tau} \tag{7}$$

$$\partial_i = \frac{\tau - \int_0^\tau e^{-\lambda_t} \quad dt}{\tau} \tag{8}$$

$$\partial_i = 1 - \frac{\left(\int_0^{t_1} e^{-\lambda} \quad dt + \int_{t_1}^\tau e^{-\lambda_t} \quad dt\right)}{\tau} \tag{9}$$

t_1 : Is the instant from which the target field C_d reaches the boundary between the two clusters Cl_i and Cl_j (see Fig. 5). From the instant t_1 to the instant τ the intensity λ of the Poisson distribution will depend on the time (this parameter will be denoted by λ_t)

$$\partial_i = 1 - \frac{e^{-\lambda} \quad t_1 + \int_{t_1}^\tau e^{-\lambda_t} \quad dt \;)}{\tau} \tag{10}$$

We have $\boldsymbol{\lambda_t} = \frac{(\boldsymbol{\pi r_s^2} - \boldsymbol{A_t} \;)}{\boldsymbol{\pi r_s^2}} \boldsymbol{\lambda}$ where A_t Is the area of the lunule formed by the evolution of the detection field of the target C_d and the radius formed by the cluster CL_i. The expression of A_t is written as:

$$\begin{aligned} A_t \; &= r_s^2 \\ &* \left(\tan^{-1}\left(\frac{r_s^2 - R^2 + C_t^2}{4\Delta}\right) + \cos^{-1}\left(\frac{R - C_t}{r_s}\right) + \tan^{-1}\left(\frac{R - C_t}{\sqrt{(r_s + R - C_t)(r_s)}}\right)\right) \\ &- R^2\left(\tan^{-1}\left(\frac{r_s - R^2 - C_t^2}{4\Delta}\right) + \frac{\pi}{2}\right) + 2\Delta \end{aligned} \tag{11}$$

C_t : The Euclidean distance separating the center of the CL_i cluster and the target.

$$\begin{aligned} A_t \; &= 2\Delta + r_s^2 \sec^{-1}\left(\frac{2 \; r_s * C_t}{R^2 - r_s^2 - C_t^2}\right) \\ &- R^2 \sec^{-1}\left(\frac{2 \; R * C_t}{R^2 + C_t^2 - r_s^2}\right) \end{aligned} \tag{12}$$

With:

$$\Delta = \frac{1}{4}\sqrt{(r_s + R + C_t)(R + C_t - r_s)(C_t + r_s - R)(r_s + R - C_t}$$

The value of the percolation must therefore be written:

$$\partial_i = 1 - \frac{e^{-\lambda} \quad t_1 + \int_{t_1}^\tau e^{\frac{-\left(\pi r_s^2 - \; A_t \;\;\right)}{\pi r_s^2} \lambda} \quad dt}{\tau} \tag{13}$$

$$\partial_i = 1 \; - \; \frac{e^{-\lambda} \quad t_1 + e^{-\lambda} \quad \int_{t_1}^\tau e^{\frac{2\Delta + r_s^2 \sec^{-1}\left(\frac{2 \;\; r_s * C_t}{R^2 - r_s^2 - C_t^2}\right) \; - \; R^2 \sec^{-1}\left(\frac{2 \;\; R * C_t}{R^2 + C_t^2 - r_s^2}\right)}{\pi r_s^2}} \quad dt}{\tau} \tag{14}$$

We suppose that:

- $R = 2 * r_s$ the radius of transmission is twice the radius of detection.
- The speed of the target is substantially constant within the same cluster, which amounts to saying that: $\frac{t_1}{\tau} = \frac{2R - r_s}{2R}$.

$$A_t = \; 2\Delta + r_s^2 \; \sec^{-1}\left(\frac{4r_s^2}{-r_s^2}\right) \; - \; 4r_s^2\left(\frac{8r_s^2}{7r_s^2}\right) \tag{15}$$

$$\Delta = \frac{\sqrt{15}}{4} r_s^2$$

$$A_t = r_s^2\left(\frac{\sqrt{15}}{4} r_s^2 + sec^{-1}(-4) + 4sec^{-1}(8/7)\right)$$

$$\partial_i = 1 - 3/4e^{-\lambda} \quad - 1/2 \frac{A_t}{\pi r_s^2}\left(\frac{\tau - t_1}{\tau}\right) e^{-\lambda}$$

$$\partial_i = 1 - (3/4e^{-\lambda} \quad + 3/100 \; \lambda e^{-\lambda} \quad) \tag{16}$$

3.4.3. *The estimation of percolation in the case with an overlapping scheme*

The rate of percolation is written as:

$$\partial_i = \frac{\int_0^\tau (\boldsymbol{P}(\beta \geqslant 1))dt}{\tau}$$

$$\partial_i = \frac{\int_0^\tau (\boldsymbol{1} - \boldsymbol{P}(\beta = \boldsymbol{0}))dt}{\tau}$$

On the basis of the hypothesis which ensures the homogeneity of the intensity of the Poisson distribution, we can deduce that in case of an overlapping clustering scheme we have:

$\lambda_t = \lambda \forall t \in [0 - \tau]$

The target, when approaching the border area of the cluster Cli, can be taken by the new cluster Clj which must be in the operational mode.

The percolation is then written:

$$\partial_i = 1 - \boldsymbol{e}^{-\boldsymbol{\lambda}} \tag{17}$$

The Fig. 6 illustrates the percolation rate as a function of the average network degree and therefore the number of sensors deployed in a cluster.

At low a density (up to $\lambda = 1.7$ now called critical intensity λ_c) the adoption of a clustering scheme with a high overlap ratio is better in terms of a percolation compared to a non-overlapping clustering scheme that approaches a VORONOI diagram. From λ_c,

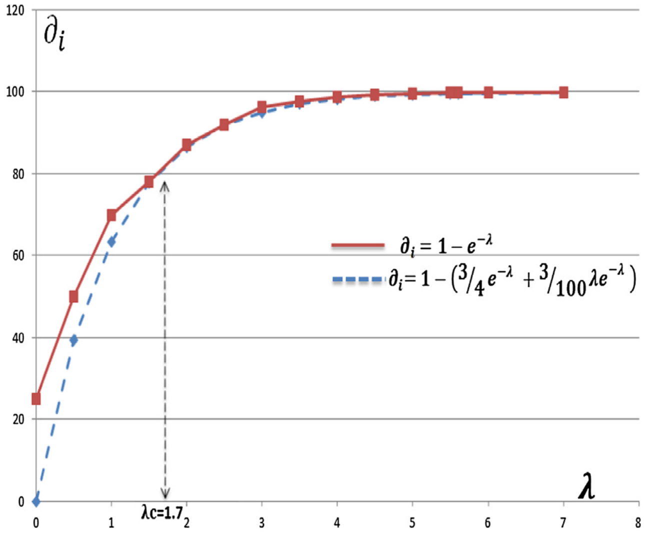

Fig. 6. The curve of the two percolation functions.

both strategies have the same percolation rate regardless of the number of the sensors deployed.

It is therefore a question of finding a compromise between the percolation rate and the energy consumption. Indeed the overlapping of the clusters generates excessive energy consumption for the same displacement of the target compared to a clustering without overlaps [25], at the same time, clustering with overlap is much better in the case of low density, in order to have a better rate of percolation.

3.5. The selection of CHs

The cluster -building process is illustrated in Fig. 8. Each node v detecting the target must generate a ΔT timer competition (Eq. 18) after which it passes to the node Cluster-Head state and subsequently sending the messages MSG_Inv to recruit member's nodes.

ΔT is comprised of two parts, as shown in Eq. (4). The first term of ΔT aims to create a repulsion effect between adjacent clusters and therefore allows distances between clusters each other.

Indeed, increasing the distance between the candidate node to be CH and the active cluster nodes, decreases its competition timer, and therefore greater probability of being a cluster - Head. This metric reduces overlap and at the same time the number of the generated cluster, and therefore better energy efficiency.

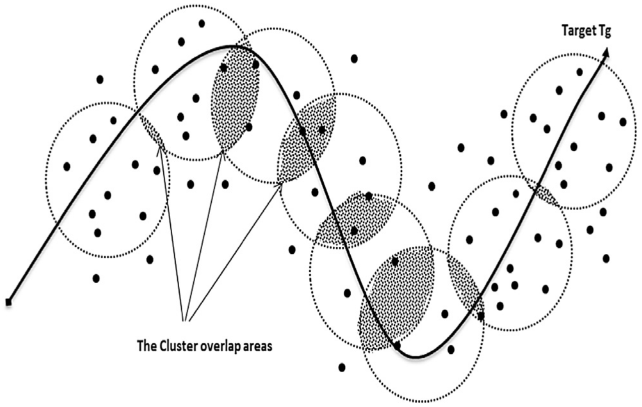

Fig. 7. The Cluster overlap area.

$$\Delta \boldsymbol{T}_v = \begin{cases} \frac{r_s}{MIN(v)} & if \;\; \partial(v) > \;\lambda_c \\ random\;() & \quad else \end{cases} \tag{18}$$

$\partial(v)$: *The density of the node v*

- λ_c : Critical density.
- MIN(v)= {Min{d($\boldsymbol{v_i}$,$\boldsymbol{v}$)} , $\boldsymbol{V}$ $\boldsymbol{v_j} \in \boldsymbol{N_{Activ}}(\boldsymbol{v})$}

$\boldsymbol{C_k}$**:{**$\boldsymbol{v_j}$**, d (**$\boldsymbol{CH}$**,**$\boldsymbol{v_j}$**)**$\leqslant$$\boldsymbol{R_t}$**}** be the set of the members of the active cluster, is defined

$$\boldsymbol{N_{Neig}} = \left\{ \boldsymbol{v_i} \in \boldsymbol{G} \cap^{\boldsymbol{C}}_{\boldsymbol{K}},\; \mathrm{d}\;\left(\boldsymbol{v_i},\; \boldsymbol{v_j}\right)\;\leqslant \boldsymbol{R_t}\;\right\}$$

G: The set of all nodes
N_{Neig} is the set of all neighboring nodes to the active cluster C_K.

$\boldsymbol{D_k}$ **= {**$\boldsymbol{v_j} \in \boldsymbol{C_k}$**|**$\exists$ $\boldsymbol{t}_0$**, d (**$\boldsymbol{L}_{(\boldsymbol{t}_0)}$**,**$\boldsymbol{v_j}$**)**$\leqslant$ $\boldsymbol{r_s}$**}** is the set of the nodes belonging to the active cluster that detected the target at some point. It also defines the set of the nodes that will be activated by the prediction process based on the over-hearing messages by:

$$\boldsymbol{N_{Activ}}(\boldsymbol{v})\left\{\boldsymbol{v_i} \in \boldsymbol{G} \cap^{\bar{C}_K}\; \exists \boldsymbol{v_j} \in \;\boldsymbol{D_K}\;\mathrm{d}\left(\boldsymbol{v_i},\; \boldsymbol{v_j}\right)\;\;\leqslant \boldsymbol{R_t}\right\}$$

Most dynamic clustering algorithms for tracking target build dynamic clusters along the path of the moving target; this dynamic aspect is relative to time since temporal clusters are built on demand. Our algorithm tries to acquire another dynamism dimension, by designing clusters whose topological form changes according to the density of the network as the target moves, to improve the rate of the percolation.

Our algorithm forms more compact clusters with considerable overlap since the density of the sensor nodes is less than the critical density λ_c and tries to acquire a repulsive effect between the adjacent clusters in order to minimize their overlaps because the density is considerable, as shown in Fig. 7.

The case of the formation of two or more clusters at the same time is possible with the counter used in the formula Eq. 18; however this case has a very low probability to be realized. In any case, the algorithm must behave in a fluid way, ensuring that data collection continues even when there is a need to destroy a cluster. Fig. 8 shows how the algorithm behaves in the case that a CH receives an invitation message MSG_INV(ID'_CH) from another CH having

The Id ID'_CH. In this case, the CH with the lowest ID must send MSG_Dest (ID'_CH) destruction messages for those members, which immediately change membership to the winning cluster.

4. Performance evaluation

In this section, we present the simulation settings, evaluation metrics and we discuss the obtained results. We have used the NS3 simulator to perform this task. In the next experiments, the S -MAC IEEE protocol is used as a model for the data link layer. We are also fixing the bandwidth of all nodes to 2 Mbit/s and Rt scope nodes to 200 m while sensing to 50 m. The parameter values of the simulation are summarized in Table 1.

The results presented in this section are corresponding to an average value of the results obtained by simulation of several scenarios chosen in a random manner.

To highlight the contributions of this algorithm, we will focus on the following metrics:

- The percolation of the target in the observation field.
- Tracking accuracy: This represents the probability of having the right target's location. The average of the Euclidean distances, between the estimated location and the real location, obtained is used to measure the tracking error.

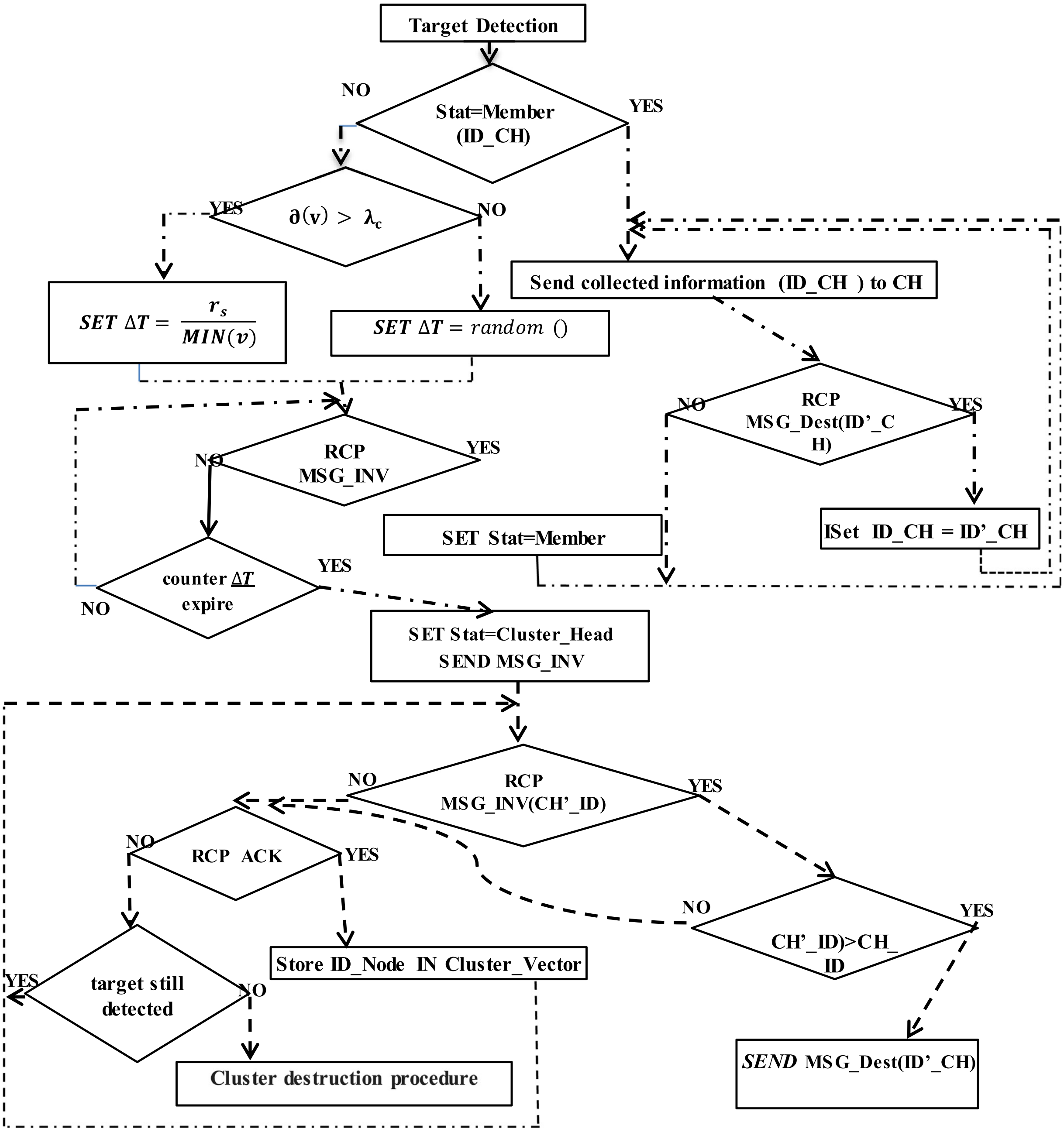

Fig. 8. The flowchart of the algorithm.

- Energy consumption: which represents the total energy consumed as a result of communication of the sensor nodes with each other after the simulation begins.

Table 1
The simulation Parameters.

parameter	values
deployment surface	(0,0) X (10000 m, 10000 m)
Displacement of the target	Random
Eelec	50nJ/bit
Eamp	10pJ/bit/m2
Eamp2	0.0013pJ/bit/m4
Initial energy of a sensor	10 J
Sleep mode	0.005 J/seconde
Active mode	0.078 J/seconde

The main difference between the rate of the percolation of the target and the tracking accuracy is that percolation expresses probability value in relation to time, whereas the tracking accuracy is related to the tracking space. Most of the tracking algorithms try to best express this metric, whereas the percolation metric is implemented for the first time in our algorithm. We are comparing our algorithm DC-AIPT with two dynamic clustering schemes for the target tracking cited in Section 2: CHEW [18] is using, as the prediction method, the overhearing messages from the active clusters, the second solution DKF DC [20] is using Kalman filters to prevent the trajectory of the target.

4.1. The mobility model chosen for the target

This part presents the mobility model used for the target to obtain the simulations. The Markov Gauss model can be totally random, partly random or deterministic according to a parameter of the degree of chance adjustment α, with $0 \leq \alpha \leq 1$. Initially, a speed and a direction are assigned to different mobile nodes. For fixed time intervals, movement occurs by updating the speed and direction. More precisely, the speed values and the next position at the moment are calculated from those of (n-1). The values of speed, direction and position at every moment, depend on the values of the previous moment which creates regular movements of the nodes. Fig. 9 shows the movement of a node according to the Markov Gauss model with three different values of the parameter α. When α is equal to 1, the motion is linear. The node keeps them values of initial speed and direction

4.2. The target percolation rate

We begin the evaluations of the mean performances of our algorithm **DC-AIPT** by studying the impact of the density and the speed of the target on the Rate of percolation. Then we study the impact of the critical density λ_c on this rate.

4.2.1. The percolation versus the density and speed of the target

We vary the density of the network at the same time as the speed of the target. We change the number of nodes from 100 to 1000 in steps of 100. For each network size, we change the target speed from 10 m/s to 900 m/s in steps of 100. We obtain the curve representing the average percolation rate as a function of the density and the speed of the target (Fig. 10).

We note that, except for high speeds of the target (from 700 m/s), the rate of the percolation achieved by the algorithm is 100% for high node density, and remains very acceptable displaying up to 65% for a very low density. With the increase in speed the percolation rate decreases. In the case of low densities, we observe a negative peak which drops the percolation rate to 2%. We observe the same phenomenon as with large densities with a more moderate drop slope than with low density. This can be explained by the fact that for high velocity, the target simply escapes from most trained clusters, and there is a relapse of the rate of percolation.

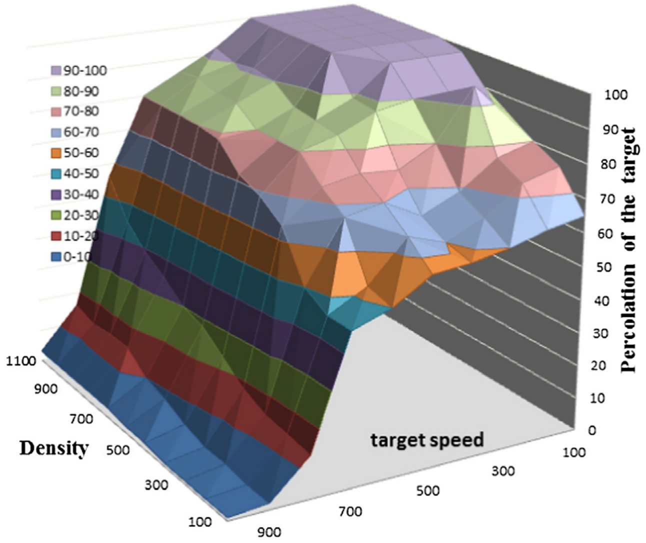

Fig. 10. The percolation versus the density and the target speed.

4.2.2. The percolation versus the density and the critical density λ_c

Remember that our algorithm uses the ΔT counter below to build the clusters.

$$\Delta \boldsymbol{T}_{v} = \begin{cases} \frac{r_s}{MIN(v)} & \text{if } \partial(v) > \lambda_c \\ random\ () & else \end{cases}$$

The graph in Fig. 11 clearly shows that the percolation of the target increases as the density of the network (number of sensor nodes) increases, which is quite natural given the very definition of percolation. On the other hand, the percolation rate is much better for low values of the critical density for any value of the network density. This can be explained by the fact that with low values of λ_c the algorithm tends to build adjacent clusters with a considerable overlap rate which increases the rate of percolation at the expense of energy performance. The energetic aspect will be studied in the following section.

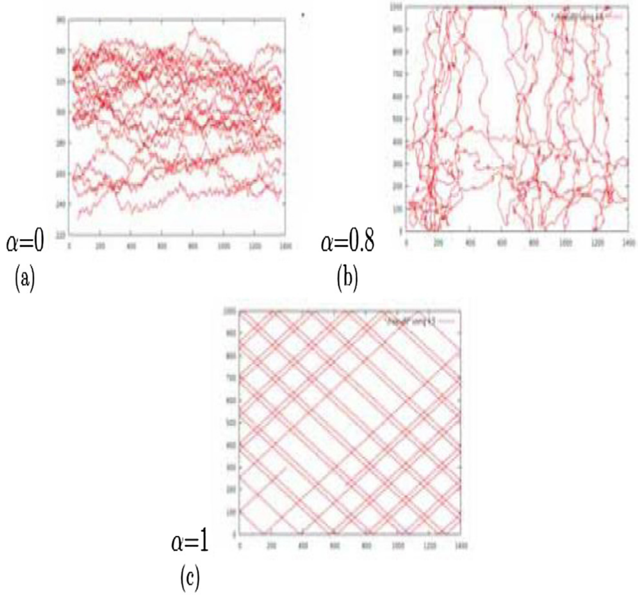

Fig. 9. Markov Gauss motions with three values of α.

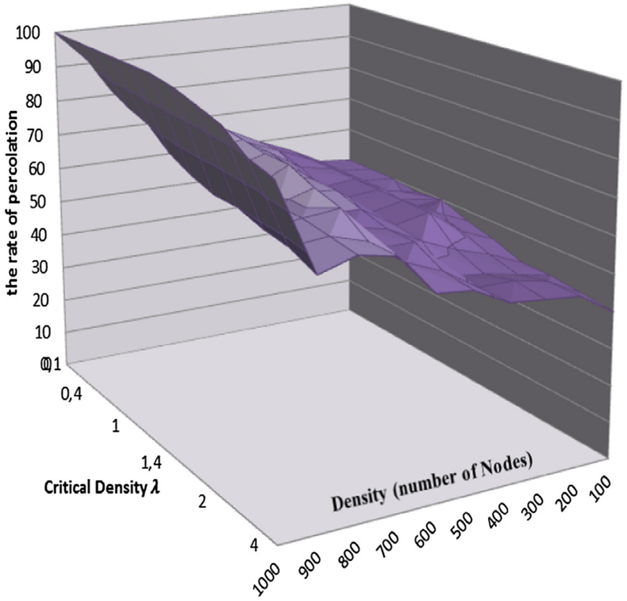

Fig. 11. The percolation versus the density and the critical density.

The list of abbreviations of the graph:

- **(ID_CH):the identifier of the Cluster Head.**
- $\partial(\boldsymbol{v})$: **The density of the node v.**
- λ_c **: Critical density.**
- **MSG_Dest(ID'_CH): destruction message from ID'_CH.**
- **MSG_INV: invitation message to be a member of the cluster.**
- **ACK: acknowledgment.**
- **ID_Node: the identifier of the node.**
- **Cluster_Vector: local database to store the IDs of the member nodes**

4.3. The accuracy of the algorithm

The accuracy of tracking algorithms implies low probability of missing the moving object. It is determined by square root of the difference between the predicted and actual location of the object in two dimensions.

The prediction error is given by:

$$\mathrm{Epr} = \sqrt{(X_{pre} - X_{real})^2 + (Y_{pre} - Y_{real})^2}$$

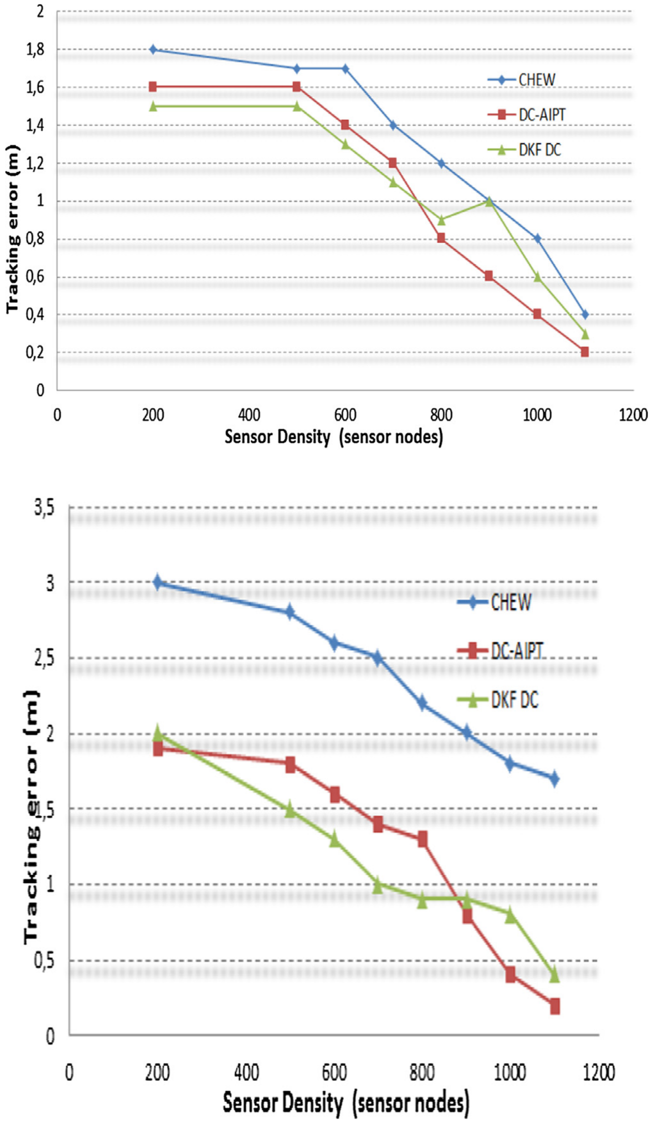

Fig. 12. Tracking error versus density.

where the Epr is the prediction error of predictor at time t and (Xpre, Ypre) are the calculated coordinates and (Xr, Yr) are the real ones.

Simulations are conducted to study the influence of the sensor density on tracking error. It was carried out with variable and random speeds of the target.

Fig. 12 (12.a target speed v = 500 m/s and 12.b with v = 1000 m/s) depicts the accuracy of the algorithms cited in the related work part (DC-AIPT, CHEW and DKF DC) with varying the sensor density. The tracking error of all algorithms is relatively small and acts almost similarly, since as the sensor density increases more and more, the sensors get closer to each other more and more. This makes the localization of the target more accurate which in turn makes the probability that the dynamically formed clusters miss the target is lower and lower.

The DKF DC and our proposed model have substantially low prediction error than the CHEW algorithm. The EKF predictor uses a first-order linearization that sometimes leads to instabilities during simulation time, especially for faster speeds of the target. Our algorithm provides better precision thanks to the wake-up waves described in Section 3.C. This can be proved again by looking at Fig. 13which represents an example of a real trajectory against those obtained by the proposed algorithms.

For normal trajectory, the three algorithms record identical results; on the other hand our algorithm presents the best results in case of a trajectory presenting sudden changes of directions.

5. The energy consumption

We rely on the energy model whose parameters are summarized in Table 1. To evaluate the energy consumed during the monitoring process, we record the energy cost of different tasks such as the sensors startup, the time of the activity, the location and the messages exchanged with the CH. The simulation time will be

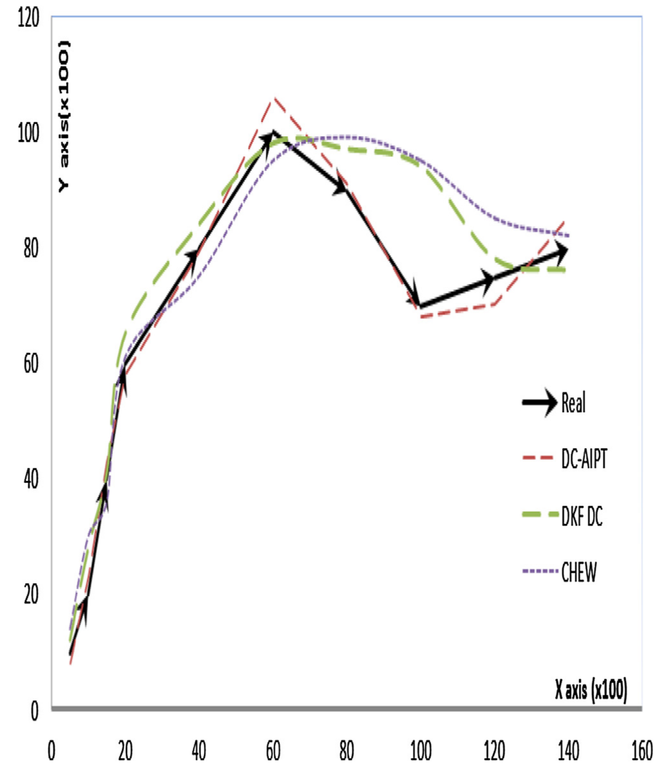

Fig. 13. An example of trajectories obtained by the three algorithms compared with the real trajectory of the target.

set to 200 s. The results presented in this section are corresponding to the mean value of the results obtained by simulation of several scenarios selected in a random manner. The proposed clustering algorithm has been designed with the aim of providing a solution which makes it possible to obtain a better rate of percolation independently of the distribution of the sensor nodes while having a better energy efficiency.

5.1. Impact of the critical density λ_c on energy efficiency

We theoretically demonstrated in section D-1 that the critical intensity is $\lambda_{c'}$ = 1.7 to have a compromise between percolation rate and energy consumption. To analyze the impact of the parameter λ_c, we consider the networks with sizes from 100 up to 1000 nodes. For each network size, we vary the value of λ_c from 0.2 to 5, then we measure the total energy consumed. The 3D graph of Fig. 14 shows the result of these tests. The increase in the number of nodes naturally increases the energy consumed but with different proportions according to the parameter $\lambda_{c'}$ in fact the graph has an apparent convexity when the critical intensity is close to the value found theoretically: λ_c= 1.7. The energetic gain is obvious and can reach 30% according to the distribution of the nodes. This proves once again the rigor of our approach. In the next section, we want to compare the energy efficiency of the proposed solution with the clustering schemes mentioned in the related work section. We can further observe that DC-AIPT algorithm, with $\lambda_c = 1.7$, outperforms all other algorithms in term of energy consumption as it is reduced by 15 and 30% compared to that in CHEW and DKF DC algorithms, respectively. Adopting a new prediction method that guarantees the percolation and the precision in the formed clusters, that is extremely helpful in mitigating the energy consumption. This can be explained by the fact that our algorithm while having for its main objective the improvement of the percolation of the target, had at the same time reduced the percentage of the disappeared target. This logically increases the number of nodes awakened in the network to redetect the target before monitoring. This percentage is obviously important in the other two algorithms Fig. 15a,b.

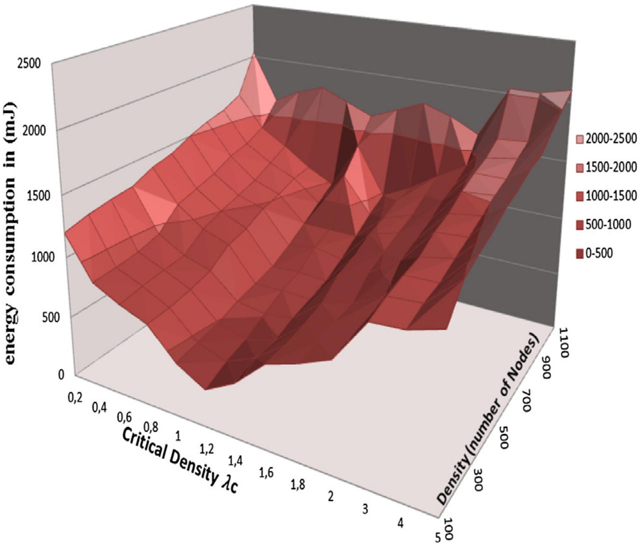

Fig. 14. Energy consumption versus density and critical density.

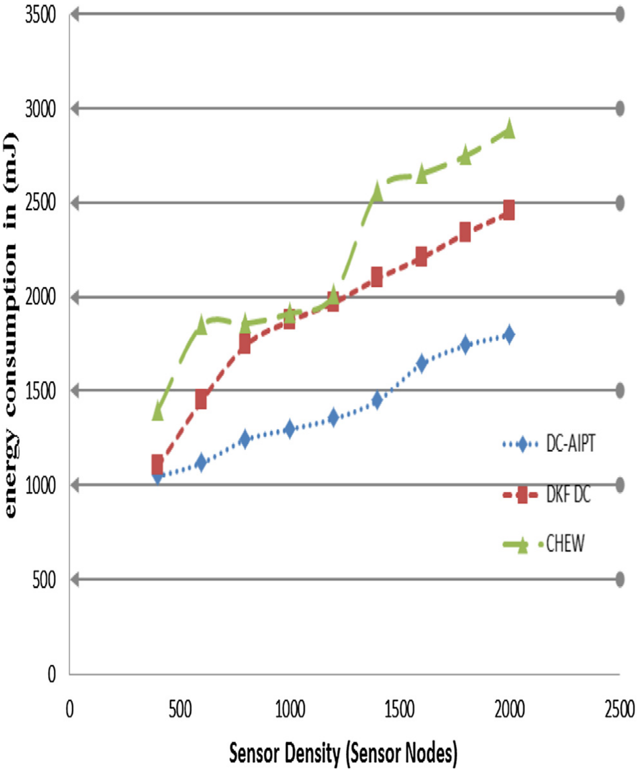

Fig. 15a. Energy consumption versus density (target speed 500 m/s).

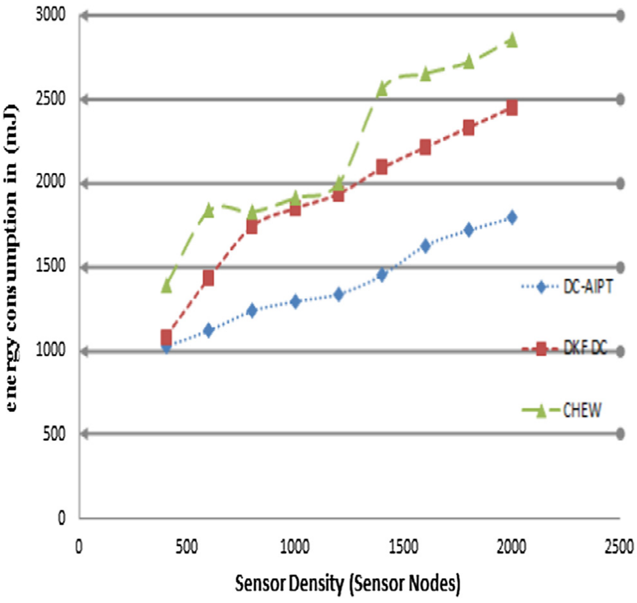

Fig. 15b. Energy consumption versus density (target speed 1000 m/s).

6. Conclusion

In this article, we have proposed a clustering algorithm for target tracking. Our algorithm has built dynamic clusters on demand and for the first time it has included the concept of target percolation as a major criterion of performance. The main objective of our solution is to handle the trade-off between the percolation of the target, tracking precision and energy consumption.

Thus, a counter for the selection of CHs has been proposed. This term allows creating a propulsive effect between clusters in case of high density around the target, which makes it possible reducing the overlaps and at the same time the number of generated cluster, and thus a better energetic efficiency. In case of low local density, the algorithm tries to create clusters with a large overlap ratio to have a better percolation. The target evolves within the network, which behaves like a spring that relaxes and compresses itself according to the density encountered.

We have justified the correctness of our analysis through proof, and validated the effectiveness of the algorithm with extensive simulation experiments.

The simulation results have demonstrated the effectiveness of the proposed models and the prevalence of our algorithm compared with other algorithms considering different network densities, target speeds, and network dimensions.

As future work is concerned, we are planning to design a variant of the ATHVC algorithm to adapt to an environment with multiple targets. This type of monitoring is complex and involves additional steps, including identification to help differentiate and count targets. In addition, our new evaluation metric for target tracking algorithms called percolation should be, through further study, evaluate the recently proposed monitoring algorithms in consultation with the authors of these articles.

Acknowledgements

The authors thank the anonymous reviewers for helpful comments that increased the quality of the article in particular Mr. Lhoussain SIMOUR.

Funding

This work has received funding from the laboratory RITM-ESTC/CED-ENSEM, University Hassan II Casablanca, Morroco.

Authors' contributions

This work is done in the context of the Ph.D. studies of MT under the supervision of AM and MO. All authors read and approved the final manuscript.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.eij.2019.04.002.

References

[1] Wang X, Wang S. Hierarchical deployment optimization for wireless sensor networks. IEEE Trans Mobile Comput 2011;10:1028–41.

[2] Chen Jiming, Cao Kejie, Li Keyong, et al. Distributed sensor activation algorithm for target tracking with binary sensor networks. Cluster Comput 2011;14 (1):55–64.

[3] Bartolini N, Calamoneri T, La Porta T, Silvestri S. Autonomous deployment of heterogeneous mobile sensors. IEEE Trans Mobile Comput 2011;10:753–66.

[4] Wu Bo, Feng Yan-Peng, Zheng Hong-Yan, et al. Dynamic cluster members scheduling for target tracking in sensor networks. IEEE Sens J 2016;16 (19):7242–9.

[5] Bernard Thibault, Bui Alain, Pilard Laurence, et al. A distributed clustering algorithm for large-scale dynamic networks. Cluster Comput 2012;15 (4):335–50.

[6] Vasuhi S, Vaidehi V. Target tracking using Interactive Multiple Model for Wireless Sensor Network. Information Fusion 2016;27:41–53.

[7] Sabet MT, Fathi AR, Mohammadi Daniali HR. Optimal design of the Own Ship maneuver in the bearing-only target motion analysis problem using a heuristically supervised Extended Kalman Filter. Ocean Eng 2016;123:146–53.

[8] Hirpara Kirti, Rana Keyur. Energy-efficient constant gain kalman filter based tracking in wireless sensor network. Wireless Commun Mobile Comput 2017;2017.

[9] Zhang Shouzhen, Xie Changjun, Zeng Chunnian, et al. SOC estimation optimization method based on parameter modified particle Kalman Filter algorithm. Cluster Comput 2018:1–10.

[10] Avci Besim, Trajcevski Goce, Tamassia Roberto, et al. Efficient detection of motion-trend predicates in wireless sensor networks. Computer Commun 2017;101:26–43.

[11] Christy Jeba Malar A, Kousalya G, Maode MA. . Markovian model based indoor location tracking for Internet of Things (IoT) applications. Cluster Comput 2017:1–8.

[12] Darman R, Ithnin N. Object tracking methods in wireless sensor network: Network structure classification. In: Proceedings of the International Conference on IT Convergence and Security. Beijing: IEEE Xplore Press; 2014. p. 1–3. doi: https://doi.org/10.1109/ICITCS.2014.7021793.

[13] Demigha O, Slimane HO, Bouziani A, Hidouci W-K. Energy efficient target tracking in wireless sensor networks with limited sensing range Barcelona, Spain. Proceedings of the 6th International Conference on Systems and Networks Communications, 2011.

[14] Wang Xingbo, Fu Minyue, Zhang Huanshui. Target tracking in wireless sensor networks based on the combination of KF and MLE using distance measurements. IEEE Trans Mobile Comput 2012;11(4):567–76.

[15] Wang Xue, Wang Sheng, Ma Jun-Jie. An improved co-evolutionary particle swarm optimization for wireless sensor networks with dynamic deployment. Sensors 2007;7(3):354–70.

[16] Demigha Oualid, Hidouci Walid-Khaled, Ahmed Toufik. On energy efficiency in collaborative target tracking in wireless sensor network: a review. IEEE Commun Surveys Tutorials 2013;15(3):1210–22.

[17] Alaybeyoglu Aysegul, Kantarci Aylin, Erciyes Kayhan. A dynamic lookahead tree based tracking algorithm for wireless sensor networks using particle filtering technique. Comput Electric Eng 2014;40(2):374–83.

[18] Liu B, Brass P, Dousse O, Nain P, Towsley D. Mobility improves coverage of sensor networks. In: Proceedings of the 6th ACM International Symposium on Mobile ad hoc Networking and Computing (ACM MobiHoc 2005). IL, USA: Urbana-Champaign; 2005. p. 300–8.

[19] Liu B, Towsley D. A study of the coverage of large-scale sensor networks Piscataway, NJ, USA. In: IEEE International Conference on Mobile Ad-hoc and Sensor Systems. p. 475–83.

[20] Banimelhem Omar, Mowafi Moad, Aljoby Walid. Genetic algorithm based node deployment in hybrid wireless sensor networks. Commun Network 2013;2013.

[21] Adlakha S, Srivastava M. Critical density thresholds for coverage in wireless sensor networks New Orleans, LA, USA. In: 2003 IEEE Wireless Communications and Networking (WCNC). p. 1615–20.

[22] Balister P, Kumar S. Random *vs.* deterministic deployment of sensors in the presence of failures and placement errors Rio de Janeiro, Brazil. In: Proceedings of IEEE INFOCOM 2009. p. 2896–900.

[23] Pavone Marco, Arsie Alessandro, Frazzoli Emilio, et al. Distributed algorithms for environment partitioning in mobile robotic networks. IEEE Trans Automatic Control 2011;56(8):1834–48.

[24] Broadbent Simon R, Hammersley John M. Percolation processes: I. Crystals and mazes. In: Mathematical Proceedings of the Cambridge Philosophical Society. Cambridge University Press; 1957. p. 629–41.

[25] Dunham William. Journey Through Genius: The Great Theorems of Mathematics. Wiley; 1990.

Mohamed Toumi is a graduate engineer in 2002 from school MOHAMMADIA (EMI) RABAT MORROCO, currently he is working as a professor at OFPPT (Morocco) . He joined the laboratory RITM - ESTC/CED- ENSEM, University Hassan II in 2013. In the years past, he supervised a number of students PFE, participated in several research projects, acted as program committee member, organizer of numerous scientific conferences. His research interests cover the field of communication networks, especially autonomic, self-organized and wireless mobile ad hoc networks. His core research area deals with service provisioning in mobile ad hoc networks.

Abderrahim Maizate received his Engineering Diploma in Computer Science from the Hassania School of Public Works since 2004 and DESA degree from ENSIAS in 2007. He received with honors the Ph.D. degree in Computer Networks and Telecommunications from Chouaib Doukalli university in El jadida in 2014. Since October 2014 he has been working as a Professor at the Computer Engineering Department at the Higher School of Technology of Casablanca (ESTC-UH2C). He is currently the Deputy Director of the Network, Computer, Telecommunications and Multimedia Laboratory at the same School and member of the research center TIC-DEV. He is also an IEEE member. His research interests include fields such as Wireless communications, WSN , smart cities, NDN, Big-data, cloud computing and security. He is also a TPC member and a reviewer for many international conferences. He is a member of the publishing committee of the Mediterranean Telecommunication Journal and member of the organizing committee of the international conferences SysCo 16 and CMT'2018.

Mohammed Ouzzif joined the Higher School of Technology in 1996 as assistant professor. He supported his Ph.D. degree in 1999 at the Faculty of Science of Rabat. The research presented during this defense concerns the specification and the formal description of videoconferencing systems using the SDL and Mondel language. In 2005, he presented his research for the National Doctorate at the National School of Computer Science and Systems Analysis. He has made contributions in the dynamic management of resource sharing in a collaborative system. This has been elaborated by distributed algorithms allowing the extension of static distributed mutual exclusion algorithms for the support of new joins and departures dynamically. These algorithms are formally specified using finite state machines and their verification using the model checking technique. Currently, Mr. Mohammed OUZZIF's research interests concern distributed systems and networks. He has published a considerable number of scientific articles in various journals, conferences and workshops and has given many regular and invited lectures. He is a Steering Committee member of the NYTES on Networked System, and he served on the Technical Program Committees of around 40 international conferences and workshops, most of which were related to wireless network and distributed systems. He is a professional member of the IEEE Computer Society.