WSN dynamic clustering for oil slicks monitoring

Said Harchi, Jean-Philippe Georges, Thierry Divoux

WSN dynamic clustering for oil slicks monitoring

Said Harchi, Jean-Philippe Georges, Thierry Divoux

Université de Lorraine, CRAN, UMR 7039, Vandœuvre lès Nancy, F-54506, France

CNRS, Centre de Recherche en Automatique de Nancy, UMR 7039, F-54506, France

firstname.name@univ-lorraine.fr

Abstract—**Clustering algorithms for wireless sensor networks are known as application-dependent. In the framework of the oil slicks monitoring in marine environment, this paper describes the modeling of a node implementing a dedicated dynamic partitioning algorithm, using a specific metric for which simulation provides optimal weights for combining various parameters (energy, density, mobility).**

Index Terms—**wireless sensors network; clustering; mobility**

I. INTRODUCTION

Initiated following a recommendation adopted at the United Nations Conference on the Human Environment (Stockholm, 1972), the CDI (Center for Documentation and Information) and the WMO (World Metrological Organization) launched a monitoring program of marine pollution by hydrocarbons. This program is part of the Integrated Global Ocean Station System (IGOSS). It can advantageously exploit the technology of wireless sensor networks (WSN). Indeed, the drift of oil slicks on the surface of seawater, their consistency and their size could be monitored continuously and in real time by several wireless sensors deployed on these slicks.

These sensors have a storage unit, a data processing unit, various units of measurement, a battery supply and a radio communication interface. Once deployed on the oil slicks, these wireless sensors self-organize into distributed subnetworks to collect data and transmit them to a data processing station called Base Station. Such a system can be beneficial (continuous monitoring and in real-time) compared for example to the classical plane flyover. However, although this type of network is increasingly used in various applications [1], [2], there are still challenges that could be to build a consistent information system based on this type of network.

The major challenge to use WSN in our application is to cope with the dynamics of the system due to:

- The mobility of oil slicks which induces a topological evolution of the network. Indeed, oil tends to spread immediately to the water surface and form slicks which, in theory, are circles whose diameter increases eight times in the hour following the spill, and then five times before the end of the week. Then, the wind, waves and the Coriolis force (due to Earth rotation) contribute to the stretch, displacement and fragmentation, and finally to their decomposition into droplets. Therefore, it is obvious that the future of slicks in the medium and long term is very difficult to predict.

- The mobility of nodes. Indeed, under the influence of factors such as wind or wave, nodes can move inside the oil slicks. Moreover, depletion of batteries or failure of nodes also change the network topology. Even if a node might slightly travel with a different speed than the oil slicks, it is assumed that a set of nodes will be representative of an oil slick location.

Thus, the network dynamics caused by these two types of mobility (slicks and nodes) can lead to a total or partial loss of connectivity for one or more nodes leading to an inconsistency of the global information system.

The architecture of this information system for the oil slicks surveillance using a WSN comprises three levels: level 3 consists of an infrastructure for processing data collected at level 2. This second level is composed of mobile collectors which periodically poll cluster heads defined at level 1. This last level represents the partitioned network on which we focus in this paper by describing a model of a sensor node, using the Opnet environment. This model implements a partitioning (clustering) algorithm adapted to a dynamic network. The choice of the partitioning strategy is dictated by the nature of our system, since the nodes will be grouped naturally on the oil slick. Finally, we evaluate the influence of some parameters of the metric on the stability of our network clustering.

This paper is organized as follows: in the next section, we describe the macroscopic model of the sensor node that we developed in the Opnet environment. In section III, we present the clustering algorithm which meets the challenge "dynamic network". In section IV, we discuss about the metric we have to implement in the previous clustering algorithm. In section V, we illustrate through simulation, the usefulness of our model and discuss the adequate choice of some parameters of the chosen metric. We conclude this work and present future research directions in the last section.

II. NODE MODELING

In order to evaluate the impacts of the clustering parameters for such oil slicks monitoring application, we decided to use the Opnet Modeler network simulation tool. However, only models of basic protocols like the IEEE 802.15.4 are available for this tool. Based on these models, we are proposing a complete sensor node model. The main development consists in adding a network layer implementing the clustering. Thus the model implements four layers (Fig. 1).
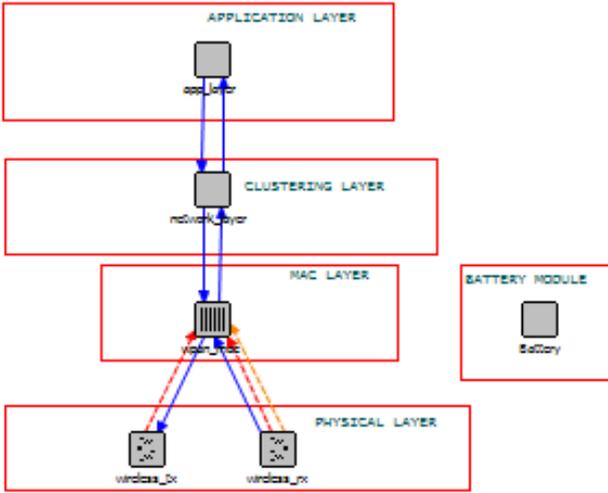
Fig. 1.   Sensor node model

Fig. 2.   Finite state machine of the clustering process

a) Physical and data link layers: are based on the IEEE 802.15.4 protocol (modeled by standard Opnet library). Medium access relies on an unslotted mechanism without beacon. When a node wants to send a message, it starts by testing if the medium is busy. If it is not the case, it sends directly otherwise it waits for a variable period defined in standard (IEEE 802.15.4). In this mode, it is expected that the energy consumption will be optimized and that the node will use the channel only when a message has to be sent.

b) Clustering layer: implements a distributed and hierarchical clustering algorithm. This layer aims at building the clusters, electing cluster heads (CH) and defining the routes that will follow the messages from a member to a CH.

The CH election is based on weights defined for each node. The weight of a node comes from a metric that should be adapted to the application features. It associates the remaining residual energy, density and mobility (defined as the evolution of the average distance between a node and its neighbors). The node with the larger weight is elected as CH. If two nodes have the same weight, the node with smaller mobility is elected.

Fig. 2 shows the finite state machine of the process implemented at the clustering layer in our sensor node model. Its is based on the different states/roles that a node might have.

- init : initialization of state variables
- ordinary: when a node is not yet associated to a cluster
- member: the node is associated to a cluster (and will send to the CH its measurements)
- gateway: the node is at the border of a cluster
- cluster head: the node is the cluster head (it receives data from cluster members and transfer them to the collector)

c) Application layer: implements a specific process that depends on the node's role in the clustering (cluster head or member). A member node deals only with the level 1 of the infrastructure while a cluster head is related to both levels 1 and 2. For members, this process sends periodically measurements to the CH (the event-triggered is not considered here as a continuous and in real-time monitoring of oil slicks
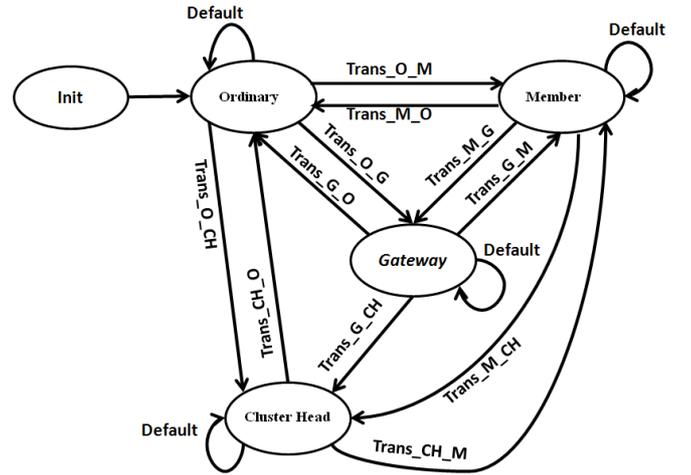
is expected). For CH, this process listens the data coming from the members plus a special event "visit of a mobile collector" (level 2). When this event occurs, this process will transfer to the collector the data collected.

III. CLUSTERING ALGORITHM

The clustering algorithm uses four broadcast messages:

- INVITE: sent by a cluster head to invite its neighbors to join its cluster
- ADHESION: sent by a node to a cluster head (to notify its membership to the cluster and also invite its neighbors to join the cluster)
- CLOSE: sent by a gateway to notify that it is joining a cluster and enclose a branch of the cluster tree
- HELLO: periodically sent by each node in order to enable the discovery of neighbors

Each message contains different fields as the node localization, the metric, the state or the parent of the source. The algorithm is based on two steps: neighbors discovery and cluster head election, and cluster building and maintenance.

A. Neighbors discovery and cluster head election

Initially, nodes start at the ordinary state as shown in Fig. 3. Here, each node sends to its neighbors an HELLO message. When a node receives an HELLO message, it updates its neighbors table by adding or updating the information: the address of the node, its coordinates, the number of neighbors and the value of its metric. These information will be next used to compute the node mobility and density.

This ordinary state is maintained for a time sufficient for the neighbors tables to be consistent. This time is fixed at 2 s in avoid the clustering instability. Then, the node compares its metric to others. If its value is the larger, transition trans_O_CH is activated and the node moves to the cluster head state (Fig. 2). Otherwise, it waits an INVITE message.

Fig. 3. Neighbors discovery and cluster head election for a node (u) that will receive HELLO message from node (v)

```
 1: state ← ordinary
 2: SEND(HELLO)
 3: if a HELLO message is received then
 4:    update the neighbors table
 5:    if metric(u)>metric(v) then
 6:       state ← cluster head
 7:    else
 8:       wait for an SEND(INVITE) message
 9:    end if
10: end if
```

B. Cluster building and maintenance

Once a node enters in the cluster head state, it launches the building of the cluster shown in Fig. 4 by sending an INVITE message to its neighbors. When a node receives such message,

- at the ordinary state, it stores the information related to the cluster head in its neighbors table, marks the cluster head node as its parent (for routing). The transition trans_O_M is activated and it moves to the member state. It sends then an ADHESION message to its neighbors (which are invited to join the cluster).
- at the member sate, the message is ignored.
- at the cluster head state, the node compares its metric with the one included in the INVITE message. If it is better, the sender is now considered as its parent, the transition trans_CH_M is activated, the node shifts to the member state and sends an ADHESION message. Otherwise, the INVITE message is ignored and the node responses by a new INVITE message in order to notify that one of its neighbors has a better metric.

When a node receives an ADHESION message, it updates its neighbors table and:

- at the member or the gateway states, the neighbors with the best metric is chosen as its parent. Depending on wether the parent has changed, an ADHESION or a CLOSE message is sent.
- at the ordinary state, the message is stored and the node waits during a given time. After this time, the neighbor which is member of the cluster and has the best metric is selected as parent. Moreover,
    - if all ADHESION messages received are coming from the same cluster, trans_O_M is activated and the node shifts to the member state. A new ADHE-SION message is sent by the node to its neighbors.
    - otherwise, trans_O_G is activated: it shifts to the gateway state and sends a CLOSE message.
- at the cluster head state, the message is ignored.

When a node receives a CLOSE message, it updates its neighbors table and stops current transmission ADHESION, INVITE and CLOSE messages.

Fig. 4. Determination of the parent of a node (u) when it receives a message from node (v)

```
 1: update the neighbors table
 2: if an INVITE message is received then
 3:    if (state = cluster head and metric(u)<metric(v)) or
       state = ordinary then
 4:       parent ← v
 5:       state ← member
 6:       SEND(ADHESION)
 7:    else if state = cluster head then
 8:       SEND(INVITE)
 9:    end if
10: else if an ADHESION message is received then
11:    if state = member or gateway then
12:       parent ← neighbor with the best metric
13:       if parent changed then
14:          SEND(ADHESION)
15:       else
16:          SEND(CLOSE)
17:       end if
18:    else if state = ordinary then
19:       update the neighbors table
20:       parent ← neighbor with the best metric
21:       if all ADHESION messages received are coming
          from the same cluster then
22:          state ← member
23:          SEND(ADHESION)
24:       else
25:          state ← gateway
26:          SEND(CLOSE)
27:       end if
28:    end if
29: end if
```

IV. CLUSTERING METRIC

A. State of the art

Many clustering techniques, centralized or distributed, have been proposed in the literature. They vary according to node deployment, the starting strategy, the election process, the features of the cluster head (CH), the size or diameter of the cluster, the used architecture, the network modeling, etc.. For example, a CH could be elected by the ordinary nodes of a cluster or pre-assigned by the network designer. It could also be an ordinary node or a super-node with fewer resource constraints (energy reserve, radio power, throughput, etc.). The cluster size (number of nodes) may be fixed or variable. In a cluster, any member could be at most 1-hop or k-hops from its CH [3], [4].

Given that our network is homogeneous and dynamic, without pre-assigned CH, we will focus on election-based partitioning algorithms. Thus the choice of CHs involves a decision criterion. It is usually a metric or combination of metrics such as: the nodes identifier [5], the nodes degree (i.e. the number of neighbors at k-hops) [6], the nodes density, the nodes residual energy, the nodes mobility [7], a random function [8], a probabilistic function [9], or a weighted com-

bination of all these elements [10].

Thus, from the literature, it can be highlighted that the DCA algorithm (Distributed Clustering Algorithm) proposed by Basagni [11] where the weight of the nodes is defined by the inverse of their velocity. That increases the probability of the least mobile nodes to become a stable CH. A node is elected as CH if it has the maximum weight among the nodes in its neighborhood. The DCA algorithm was designed for networks where nodes are static or with low mobility.

The DMAC algorithm (Distributed and Mobility-Adaptive Clustering) [12] represents an evolution of the DCA algorithm that allows the mobility of the nodes during or after the computation of the cluster. It also allows each CH to have up to (k) neighbors, and reduces the number of reassignments by adding a restructuring threshold (h).

In the MOBIC algorithm presented in [7] the CH election reflects the relative mobility of nodes. The relative mobility of a node is determined according to its neighbors transmission power. The node with the lowest mobility becomes CH. This heuristics is complex because it requires that each node is able to evaluate the power level of its neighbors to estimate its relative mobility. Moreover, the calculation of this metric does not consider physical phenomena that degrade the signal quality (obstacles, reflection, etc.).

The WCA algorithm (Weight Clustering Algorithm) was proposed in [10]. It uses a weighted sum of several metrics such as the degree, the Euclidean distance, the relative mobility, and the life time of a node as a CH. The node with the lowest weight among its neighbors becomes CH. The weight of a node (u) is defined as follows:

$$weight\left(u\right) = \alpha D_u + \beta P_u + \lambda M_u + \sigma T_u$$

where:

- $\alpha + \beta + \lambda + \sigma = 1$
- D_u is the difference between the degree of the node (u) and a constant M representing the maximum size of a cluster
- P_u is the sum of the distances between the node (u) and its neighbors. These distances are obtained by means of a locating system such as GPS
- M_u is the average relative mobility of the node (u) obtained as in the MOBIC algorithm
- T_u is the life time of a node as a CH

WCA does not minimize the cost of maintenance because the partitioning process is restarted when a node is without any CH. The weight calculation requires significant traffic and it uses a GPS system that is expensive and energy consumer.

[13] uses as a metric, the density which is defined by the ratio of the number of links by the number of nodes in the k-neighborhood of the considered node. Periodically, each node calculates its density and broadcasts it to its 1-neighborhood. Thus each node is able to compare its own density with that of its neighbors. Then it decides to promote itself as CH if it has the highest density, or to choose as CH its neighbor with the highest density. In case of equality the stability is favored.

[14] proposes the CSOS algorithm (Cluster-based Self-Organisation Scheme) which builds 2-hop clusters, of similar size. The composite metric is determined as follows:

$$weight_u\left(t\right) = \alpha d_{u,2}\left(t\right) + \beta E_u\left(t\right) + \gamma M_u\left(t\right)$$

where:

- $\alpha + \beta + \gamma = 1$
- $d_{u,2}\left(t\right)$ is the the 2-hops density
- $E_u\left(t\right)$ is the residual energy
- $M_u\left(t\right)$ is the node mobility

[15] constructed a backbone on which are built clusters using a metric defined as:

$$P_{stability} = \epsilon\left(\alpha\left(1+\Delta\right)^{-1} + \beta\left(1+M\right)^{-1}\right)$$

with:

- ϵ is the residual energy of the node
- $\Delta = |\Delta_{real} - \Delta_{opt}|$: distance to an optimal degree (Δ_{real}: real degree, Δ_{opt}: degree depending on application)
- M is the relative mobility of the node
- The parameters α and β depend on the application.

This overview shows that we can choose various parameters to partition our network. These parameters can be considered separately or in combination. In addition, each of these parameters may be subject to different definitions including the density that can be seen at 1, 2, 3 or k-hops. Finally, different types of weighting are proposed. Furthermore, these different studies have highlighted the influence of the application.

In this paper we deal with a particular application where the network dynamics will be strong. Also we will propose the definition of a clustering algorithm that is based on a metric combining energy, density and mobility. By modeling this algorithm in the Opnet environment we will demonstrate that it is possible to define the best weights applied to these three parameters in the context of the continuous and real-time oil slicks monitoring.

B. Statement

The metric used is a weighted ponderation of energy, density and node mobility. We choose the one-density, that is to say, calculated to 1-hop, instead of k-density ($k > 1$) based on the work of Mitton [16]. It showed that the 1-density provides a more robust structure compared to the k-density. In addition to a 1-hop density minimizes information over the network and therefore the bandwidth and the energy consumption. The metric is hence expressed as:

$$weight_u\left(t\right) = \epsilon_u\left(t\right)\left(\alpha d_{u,1}\left(t\right) + \left(1-\alpha\right)M_u\left(t\right)\right) \quad (1)$$

where:

- $d_{u,1}\left(t\right) = N_{u,1}\left(t-\Delta t\right)/N_{u,2}\left(t-\Delta t\right)$: the density of node (u) with $N_{u,k}\left(t\right)$ the number of k-hop neighbors of node (u)
- $M_u\left(t\right)$: the mobility of node (u) at time t which might be relative or absolute
- The parameters α depend on the application
- $\epsilon_u\left(t\right)$: remaining energy of node (u) at time t. It is computed according to the consumption pattern energy proposed by [17] for IEEE 802.15.4 MICAZ motes.

In this paper, it is assumed that nodes are able to determine their position, such that we will use the relative mobility defined as:

$$M_u\left(t\right) = \left| \frac{d_u\left(t\right) - d_u\left(t - \Delta t\right)}{\Delta t} \right|$$

with:

$$d_u\left(t\right) = \frac{1}{N_{u,1}\left(t\right)} \sum_{v \in N_{u,1}(t)} \text{dist}_{u,v}\left(t\right)$$

where:

- $x_u\left(t\right), y_u\left(t\right)$: the coordinates of node (u) at time t
- $\text{dist}_{u,v}\left(t\right)$: the euclidian distance between (u) and (v)
- $d_u\left(t\right)$: the average euclidian distance between node (u) and its 1-hop neighbors

Equation (1) is a generic metric used to compute which node will be elected as the cluster head and which cluster a (member) node will belong to. According to the parameters, it means that the logical topology might change. In this paper, we are not focusing on the residual energy level but we are concerned to see how the parameter α will act for such mobile WSN applications. In particular, the parameter α will be studied in terms of quality of clustering (mainly its stability).

V. ILLUSTRATION

A. Simulations

This section presents simulation results obtained with the Opnet tool and the sensor node model detailed in the previous section. Objective of the simulation is to emphasize the impact of the parameter α of the metric on the clustering quality. This quality is evaluated in terms of number of clusters built, number of cluster head updates during a simulation and life time of the cluster head status on nodes. The varying parameters considered during the simulation are α and the nodes speed from $0,1$ to $6\ m/s$ which corresponds to realistic climate conditions. For example, $0,2\ m/s$ corresponds to a wind of $40\ km/h$. The simulated scenario is related to the hazardous deployment of 100 sensors on a square of $1\ km^2$. Sensors were placed at the beginning of the simulation on a square of $50\ m^2$. Nodes are configured with a radio range of $50\ m$. The shifting of the nodes is simulated according to the Random Way Point [18], [19] profile. It means that a node remains fixed during a finish time after moving. For moving, trajectory is randomly selected. These two steps are repeated until the end of the simulation. This model might be not very representative of our application, and further works will provide a realistic model. On a node, the neighbors discovery process is periodically run each second, such that the density and the mobility used in the metric will be computed with $\Delta t = 7\ s$. In order to face with intermittent communication capacities, the proposed metric consider the whole list of messages received during a window time Δt. Hence the impact of messages losses will remained limited. Furthermore a node is not anymore considered as a neighbor if no HELLO message is received after $7\ s$.

Fig. 5 shows for a given value of $\alpha = 0.5$ and a constant nodes speed of $2\ m/s$, the evolution of the number of clusters during a simulation of $2\ mn$. Longer simulations have

been launched to visualize the node energy consumption. For instance, after $10\ mn$, only $0.0016\ \%$ of energy was consumed. Number of clusters is evaluated here each second.
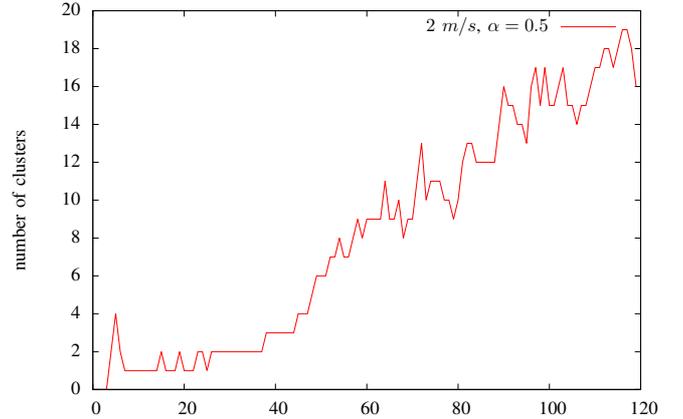
Fig. 5. Number of clusters during simulation (time in s)

The dynamic illustrated on Fig. 5 reflects this. Initially, the sensors are close, the graph is connex and hence one or two cluster might be sufficient. Next, sensors will start to move from each other, such that the density of the graph decreases, subnetworks appear and finally, more clusters are defined. Other simulations are next launched in order to illustrate the impact of the weight α.

B. Results

For each configurations (mobility speed and α values), 900 simulations (based on 30 seeds) are launched. Average values are computed with a confidence interval of $95\ \%$. The quality of the metric defined previously is evaluated here in terms of stability of the clustering. When mobile collectors will have to periodically go through the different cluster heads in order to collect their data, it is important that the trajectory of a collector remains as possible (sensors are still moving) like the previous one. It means that inside a cluster, such application are interested in the fact that the same sensor remains the cluster head. Hence, the election rate of cluster heads and the life time of the cluster head status was evaluated trough simulations. Different speeds and different values of the metric weight α were tested. Fig. 6 shows for different speeds, the average number of times a node became head of a cluster for an α value and Fig. 7 shows the average time a node remains cluster time.

Fig. 6 shows that $\alpha = 1$ reduces the nodes rate election as CH. Similarly, Fig. 7 shows it is better to have $\alpha = 1$ for a node to maintain its CH status. It means that the higher the speed, the more it damages the stability of the clustering.

However, Fig. 8 shows the interest to adapt α to the speed in order to minimize the number of clusters.

It can be highlighted one more time that for high speed, mobility must be neglected regarding density. But for low and medium speeds, the compromise is interesting because it reduces the number of clusters (Fig. 8) without affecting the stability (Fig. 6 and 7). Indeed, for speeds lower than $2\ m/s$,
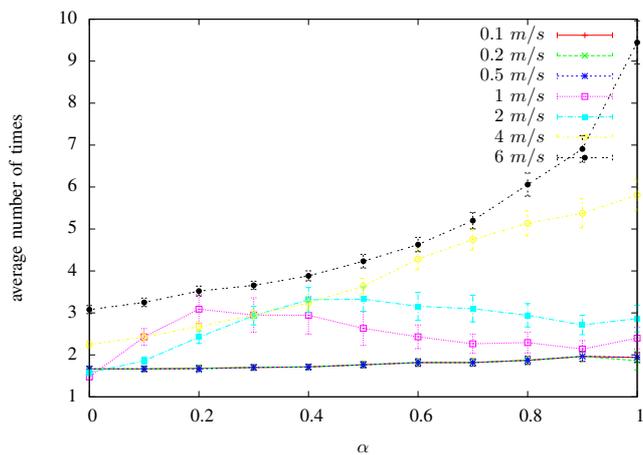
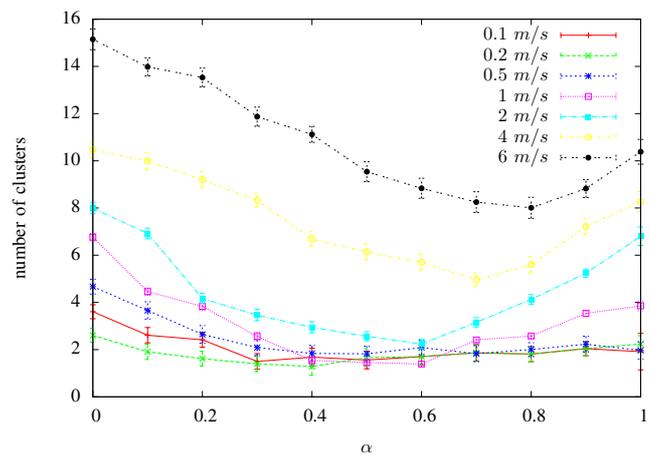Fig. 6. Number of times a node became head of a cluster according to α and the speed

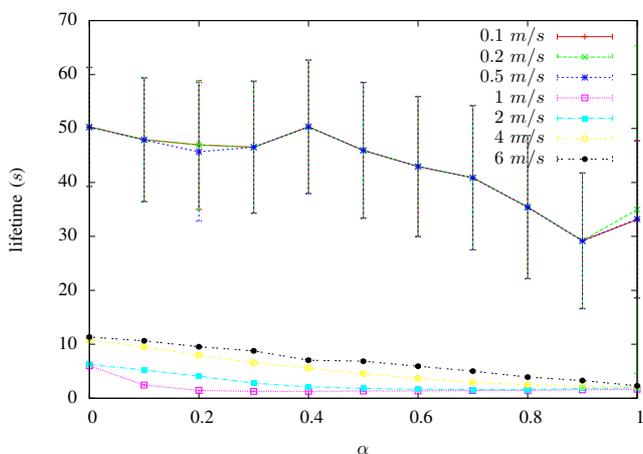Fig. 8. Number of clusters according to α and the speed

Fig. 7. Life time of cluster head status according to α and the speed

the election rate and the lifetime of a CH is quiet the same from $\alpha = 0.3$ to $\alpha = 1$.

VI. CONCLUSION

In case of disaster, the node model proposed in this paper enables to design the monitoring system to be deployed in terms of number of nodes, the measurement period, the clustering dynamics and the metric weights regarding area surface to supervise and climate conditions.

Future works will focus now on the level 2 of the monitoring system dedicated to the data mobile collectors management for which a session protocol has to be developed in order to retrieve the context of theirs previous visits. The goal is to recover a discrete connectivity in order to periodically rebuild a consistent information system.

REFERENCES

[1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, Aug. 2008.

[2] F. Regan, A. Lawlor, B. Flynn, J. Torres, R. Martinez-Catala, C. O'Mathuna, and J. Wallace, "A demonstration of wireless sensing for long term monitoring of water quality," in *IEEE 34th Conference on Local Computer Networks*, Oct. 2009, pp. 819–825.

[3] A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Communications*, vol. 30, no. 14-15, pp. 2826–2841, Oct. 2007.

[4] W. Naruephiphat and C. Charnsripinyo, "Clustering techniques in wireless sensor networks," in *International Conference on New Trends in Information and Service Science*, Jul. 2009, pp. 1273–1278.

[5] A. Ephremides, J. Wieselthier, and D. Baker, "A design concept for reliable mobile radio networks with frequency hopping signaling," in *Proceedings of the IEEE*, vol. 75, no. 1, Jan., 1987, pp. 56–73.

[6] M. Gerla and J. T. Tsai, "Multicluster, mobile, multimedia radio network," *Wireless Networks*, vol. 1, no. 3, pp. 255–265, Aug. 1995.

[7] P. Basu, N. Khan, and T. Little, "A mobility based metric for clustering in mobile ad hoc networks," in *International Conference on Distributed Computing Systems Workshop*, Apr. 2001, pp. 413–418.

[8] R. Nagpal and D. Coore, "An algorithm for group formation and maximal independent set in an amorphous computer," Massachusetts Institute of Technology, Tech. Rep., 1998.

[9] S. Bandyopadhyay and E. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor networks," in *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications (INFOCOM)*, vol. 3, Apr. 2003, pp. 1713–1723.

[10] D. S. K. Chatterjee. M and T. D, "Wca: A weighted clustering algorithm for mobile ad hoc networks. cluster computing," *Cluster Computing*, vol. 5, pp. 193–204, 2002.

[11] S. Basagni, "Distributed clustering for ad hoc networks," in *Fourth International Symposium on Parallel Architectures, Algorithms, and Networks*, Jun. 1999, pp. 310–315.

[12] S. Basagni and E. Jonsson, "Distributed and mobility-adaptive clustering for ad hoc networks," Tech. Rep., 1998.

[13] N. Mitton, B. Sericola, S. Tixeuil, E. Fleury, and I. Lassous, "Self-stabilization in self-organized wireless multihop networks," *Ad Hoc and Sensor Wireless Networks*, 2009.

[14] M. Lehsaini, H. Guyennet, and M. Feham, "A novel cluster-based self-organization algorithm for wireless sensor networks," in *International Symposium on Collaborative Technologies and Systems*, May 2008, pp. 19–26.

[15] F. Theoleyre and F. Valois, "A self-organization structure for hybrid networks," *Ad Hoc Networks*, vol. 6, no. 3, pp. 393–407, May 2008.

[16] N. Mitton, "Auto-organisation des reseaux sans fil multi-sauts a grande echelle," Ph.D. dissertation, INSA de Lyon, 2006.

[17] P. Jurčík and A. Koubâa, "The ieee 802.15.4 opnet simulation model: Reference guide v2.0," IPP-HURRAY!, Polytechnic Institute of Porto (ISEP-IPP), Tech. Rep. HURRAY-TR-070509, May 2007.

[18] C. Bettstetter, G. Resta, and P. Santi, "The node distribution of the random waypoint mobility model for wireless ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 2, pp. 257–269, Sep. 2003.

[19] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communication & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, vol. 2, pp. 483–502, Sep. 2002.