Correction, generalisation and validation of the MaxMin d-cluster formation heuristic ⋆

Alexandre Delye de Clauzade de Mazieux, Michel Marot, Monique Becker

GET/INT ; UMR5157 SAMOVAR ; Institut National des Télécommunications ; 91011 CEDEX Evry, France;

Abstract d-dominating sets in graphs are very important in wireless hierarchical networks. In the first part, the paper simplifies and extends the MaxMin d-cluster formation heuristic proposed by Amis et Al. and then proves the correctness of the extended heuristic which forms d-dominating sets. This clusterhead selection process has the advantage of being distributed and scalable.
In the second part, the cluster formation process proposed by Amis et al. is extensively studied, by using the mathematical analysis framework developed in the first section. We give an example which shows that the cluster formation heuristic presented by Amis et al. is not always valid. This is an important result since the Max-Min d-cluster formation is a well known heuristic.

Introduction

Let $\mathcal{G} = \{V, E\}$ be a graph with sets of vertices V and edges E . Clusterheads form a subset, S of V which is a $d-dominating$ set over $\mathcal{G}$. Indeed, every vertex not in S is joined to one member of S through a path of d edges in E at least.
Amis et al proved that given a graph $\mathcal{G}$, given an integer d and a given integer k, it is hard to know if there exists a $d-dominating$ set with a size lesser than k. More precisely, the authors proved that it is a NP-complete problem. The "Max-Min d cluster formation" algorithm is proposed. The $d-dominating$ set is first selected by using nodes identifiers and then a cluster is formed, by using given rules. This algorithm is well known and cited because it is the only one which is distributed, scalable and forms multi-hop cluster in a short time.

The above algorithm is interesting because of the few numbers of exchanges between nodes. Clusterheads are selected among the nodes which have a maximal local address. This algorithm is generalised in order to select nodes depending of a given criterion. The given criterion might be the degree of nodes, density or energy. The first section of this paper simplifies and proves the correctness of our generalised algorithm to select clusterheads. In the second section, the cluster formation process proposed in [1] is extensively studied, by using the analysis mathematical framework developed earlier.

⋆ This work is funded by the Programme Initiative Réseaux Spontanés of the Groupe des Écoles des Télécommunications

1 Related Work

Aggregation of nodes in clusters allows to reduce the complexity of routing algorithms, to optimize the resource by allowing the medium to be managed locally by a clusterhead, to facilitate the data aggregation, to simplify the management of the network and in particular the addresses assignment, to optimize the energy consumption, and finally to make the network more scalable.

The authors of LEACH (cf. [2]) propose an autoconfigurable architecture based on clusters which minimise the energy of nodes. As this algorithm gives only guarantees on average on the number of clusters and their placement, a centralized version of the algorithm (LEACH-C) is also proposed. It allows to determine the optimal configuration to minimize the spent energy. To solve the problem of the absence of guarantee on the good formation of the clusters due to the LEACH algorithm, O. Younis and S. Fahmy (cf. [3]) propose the HEED algorithm which allows to select a clusterhead according to its energy and a given cost function, depending on the target performance goals, from the number of its neighbours, their proximity or the average of the minimum power necessary to its neighbours to reach it. Either very dense clusters, or conversely clusters distributing the load can then be obtained.

Several, but few, multihop clustering algorithms (e.g. [1,3,4]), have been proposed in literature (the justification for using mutihop clusters may be found in [5]). Among them, the authors of [1] propose a heuristic to form a d-dominating set. This paper, highly cited in the literature, presents interesting results showing that setting up d-hop clusters is an NP-

difficult problem and they propose a heuristic which is studied and improved in our paper.

2 Formation of d-dominating sets based on a given criterion

This part is extending the results published in CRAS [6] (Compte Rendu à l'Académie des Sciences).

2.1 Notations and introduction to the algorithm

Let us consider $x \in V$, $\mathcal{N}_i(x)$ is the set of neighbours which are less than i hops from x ; $(\mathcal{N}_i(x))_i$ is an increasing sequence for set inclusion. Let Y be a set on which a total order relation is defined. Let v be an injective function of V in Y. Let X be the image set of V by v ; v is a bijection of V over X. The reverse function is noted v^{-1}: $\forall x \in V \quad v^{-1}(v(x)) = x$.
The algorithm includes $2d$ runs. The d first runs constitute the *Max phase*. The d last runs constitute the *Min phase*. Each node updates two lists *Winner* and *Sender*, of $2d+1$ records. Winner is a list of elements of X. Sender is a list of elements of V. Let us note $W_k(x)$ and $S_k(x)$ the images in x of the functions W_k and S_k, defined by induction.

The basic idea of the $d-dominating$ setting is the following: during the first phase, the Max phase, a node determines its dominating node (for the given criterion) among its d hop neighbours ; a second phase, the Min phase, lets a node knows whether it is a dominating node for one of its neighbour nodes. For a given criterion, the only dominating set is built from this simple process.

Initial Phase: $k = 0$

$$\forall\, x \in V \quad W_0(x) = v(x) \quad S_0(x) = x$$

Max Phase: $k \in [\![1; d]\!]$
Let us assume that the W_{k-1} and S_{k-1} functions have been built.
For $x \in V$, let $y_k(x)$ be the only node of $\mathcal{N}_1(x)$ which is such that:

$$\forall\, y \in \mathcal{N}_1(x) \setminus \{y_k(x)\} \quad W_{k-1}(y_k(x)) > W_{k-1}(y)$$

W_k and S_k are derived from:

$$\forall\, x \in V \quad W_k(x) = W_{k-1}(y_k(x)) \quad S_k(x) = y_k(x)$$

Min phase: $k \in [\![d+1; 2d]\!]$
Let us assume that the W_{k-1} and S_{k-1} functions have been built.
For $x \in V$, let $y_k(x)$ be the only node of $\mathcal{N}_1(x)$ which is such that:

$$\forall\, y \in \mathcal{N}_1(x) \setminus \{y_k(x)\} \quad W_{k-1}(y_k(x)) < W_{k-1}(y)$$

W_k and S_k are derived from:

$$\forall\, x \in V \quad W_k(x) = W_{k-1}(y_k(x)) \quad S_k(x) = y_k(x)$$

Definition 1 *Let S be the set defined by:*
$S = \{x \in V,\, W_{2d}(x) = v(x)\}$ [1]

Theorem 1 *Each node $x \in V \setminus S$ may determine one node of S at least which is in $\mathcal{N}_d(x)$. It needs only to derive it from its* Winner *list:*

- *if x finds a pair $(v(y))$ in its* Winner *list (that is to say that $v(y)$ appears once in each of the two phases at least), then $y \in S \cap \mathcal{N}_d(x)$. If the node x find several pairs, it chooses the node y with the smallest value $v(y)$ among the pair values that it found.*
- *if not, let y be the node such that $v(y) = W_d(x)$. Then $y \in S \cap \mathcal{N}_d(x)$.*

The preceding theorem, whose proof will be given in the next part, lets us immediately derive the following corollary.

Corollary 1 *S is a d-dominating set for the G graph.*

2.2 Algorithm proof

We shall not prove the three first lemmas which derive directly from the definitions.

Lemma 1 $\forall (x,k) \in V \times [\![1; d]\!]$

- $W_k(x) = Max\,\{W_{k-1}(y), y \in \mathcal{N}_1(x)\}$
- *$S_k(x)$ is the only element y in $\mathcal{N}_1(x)$ such that* $W_{k-1}(y) = W_k(x)$

Lemma 2 $\forall (x,k) \in V \times [\![d+1; 2d+1]\!]$

- $W_k(x) = Min\,\{W_{k-1}(y), y \in \mathcal{N}_1(x)\}$
- *$S_k(x)$ is the only element y in $\mathcal{N}_1(x)$ such that* $W_{k-1}(y) = W_k(x)$

Lemma 3 $\forall\, (x,k) \in V \times [\![0; d]\!]$
$W_k(x) = Max\,\{v(y),\, y \in \mathcal{N}_k(x)\}$

Definition 2 *Let us note $M(x)$ the value $W_d(x)$.*

Theorem 2 $\forall\, x \in V \quad \forall\, y \in \mathcal{N}_d(x) \quad M(y) \geqslant v(x)$

Proof Let us assume $x \in V$ and $y \in \mathcal{N}_d(x)$. From Lem. 3, it follows:
$M(y) = W_d(y) = Max\,\{v(z),\, z \in \mathcal{N}_d(y)\}$. And from $x \in \mathcal{N}_d(y)$, it may be deduced that
$Max\,\{v(z),\, z \in \mathcal{N}_d(y)\} \geqslant v(x)$.

[1] This definition is not the same as the one which is given in [1] but both definitions are equivalent(see Th. 5 page 3).

Lemma 4 $\forall (x,k) \in V \times [\![d+1;2d]\!]$
$W_k(x) = Min\{M(y),\ y \in \mathcal{N}_{k-d}(x)\}$

Proof The proof is an induction on k, after having chosen x.

Lemma 5 $\forall (y,k) \in V \times [\![d+1;2d]\!]$
$\exists\ !x \in \mathcal{N}_{k-d}(y) \quad M(x) = W_k(y)$

Proof $W_k(y) = Min\{M(z),\ z \in \mathcal{N}_{k-d}(y)\}$. So it exists x in $\mathcal{N}_{k-d}(y)$ such that $M(x) = W_k(y)$. x is unique since the v application is injective.

Theorem 3 *Let us consider $x \in V$. Let y be the only node such that $M(x) = W_d(x) = v(y)$. Then $y \in S$.*

Proof From Def. 1 it follows that it has to be proven that $W_{2d}(y) = v(y)$. The node y is among the d hop neighbours of x since $W_d(x)$=$v(y)$, so in the other way round, x is among the d hop neighbours of y. Firstly, $Min\{M(z),\ z \in \mathcal{N}_d(y)\} \leqslant v(y)$ since $x \in \mathcal{N}_d(y)$ and $M(x) = v(y)$. Secondly it follows from Th. 2 that: $\forall\ z \in \mathcal{N}_d(y) \quad M(z) \geqslant v(y)$. So $Min\{M(z),\ z \in \mathcal{N}_d(y)\} \geqslant v(y)$.
A conclusion is $Min\{M(z),\ z \in \mathcal{N}_d(y)\} = v(y)$ and $y \in S$.

Corollary 2 *Let us consider $x \in V$. Let y be the only node such that $M(x) = W_d(x) = v(y)$. Then $y \in S \cap \mathcal{N}_d(x)$.*

Proof Theorem 3 proves that $y \in S$ and from the proof it appears that $y \in \mathcal{N}_d(x)$.

Theorem 4 *Let us consider $y \in V$ and $k \in [\![d+1;2d]\!]$. Let $x \in V$ be the only node such that $v(x)$=$W_k(y)$. Then $x \in S$.*

Proof From Lem. 5 it may be derived that
$\exists! z \in \mathcal{N}_{k-d}(y) \quad M(z) = W_k(y)$. It follows $M(z) = v(x)$.
When applying Th.3 to z and x, it follows: $x \in S$.

Corollary 3 *Let us consider $x \in V$. Let us assume that there is an $y \in V$ such that the $v(y)$ value appears again at least once in the Max phase and at least once in the Min phase for the node x. Then $y \in S \cap \mathcal{N}_d(x)$.*

Proof Theorem 4 proves that $y \in S$ because $v(y)$ appears in the Min phase. And since $v(y)$ appears once in the Max phase at least, then $y \in \mathcal{N}_d(x)$. So $y \in S \cap \mathcal{N}_d(x)$.

Remark 1 From the first point of Th. 1, it seems reasonable to choose the k-dominating node corresponding to the smallest pair, when there are several ones. This choice leads to sets which are dominated by a smaller criterion value node.

This definition of S (see Def. 1) is different from the definition given in [1]. For them, S' is defined as: $S' = \{x \in V,\ \exists k \in [\![d+1;2d]\!]\ W_k(x) = v(x)\}$.

Clearly, $S \subset S'$. The next theorem proves that the reverse inclusion is also true.

Theorem 5 $S = S'$.

Proof Let us consider $x \in S'$. $W_{2d}(x) \leqslant W_k(x)$ is a consequence of Lem. 2. So $W_{2d}(x) \leqslant v(x)$. Let us assume that $W_{2d}(x)$<$v(x)$. Lemma 5 implies:
$\exists\, y \in \mathcal{N}_d(x) \quad M(y) = W_{2d}(x)$. So $y \in \mathcal{N}_d(x)$ and $M(y) < v(x)$. But Th. 2 says that it is not true since $\forall y \in \mathcal{N}_d(x)\ M(y) \geqslant v(x)$. So $W_{2d}(x) = v(x)$ and $x \in S$.

Corollaries 2 and 3 prove Th. 2.1. Our definition is equivalent to the definition in [1]. Our definition is more performing since the whole Min phase does not need to be run.

2.3 Algorithm characterisation

The building of the d-dominating set is distributed, because it is neither necessary to know the whole topology, nor the criterion value on each node. The number of computations which have to be completed for each node, is scalable: if the node distribution is Poisson of parameter λ on a plane, if R is the transmission rate and if an edge between two nodes exists only when their distance is less than R, then the number of communications from one node is equal to $2d(1+\lambda\pi R^2)$. The time necessary to build a d-dominating set is $2d$ steps. For this d-dominating set:

$$x \in S \Leftrightarrow \begin{cases} \mathcal{N}_x(d) = \varnothing \\ or \\ \exists y \in \mathcal{N}_x(d)\ v(x) = Max\{M(z), z \in \mathcal{N}_y(d)\} \end{cases}$$

Theorem 6 *Let us consider a graph (it may be finite or infinite) and d the maximal depth chosen, let us note S_d the d-dominating set derived from the algorithm. For the same graph and for $d+1$, let S_{d+1} be the dominating set derived from the algorithm. Then $S_{d+1} \subset S_d$.*

Proof Let us consider $x \in V \setminus S_d$. $\mathcal{N}_d(x) \neq \varnothing$ so $\mathcal{N}_{d+1}(x) \neq \varnothing$. Let us consider $y \in \mathcal{N}_{d+1}(x)$ and $w \in \mathcal{N}_d(x) \cap \mathcal{N}_1(y)$. $w \in \mathcal{N}_d(x)$ so $\exists z \in \mathcal{N}_d(w)\ v(z) > v(x)$. $z \in \mathcal{N}_d(w)$ and $w \in \mathcal{N}_1(y)$ so $z \in \mathcal{N}_{d+1}(y)$ and $v(z) > v(x)$.
It follows: $\mathcal{N}_{d+1}(x) \neq \varnothing$ and
$\forall y \in \mathcal{N}_{d+1}(x) \quad \exists z \in \mathcal{N}_{d+1}(y) \quad v(z) > v(x)$ so
$x \in V \setminus S_{d+1}$. It may be derived that: $S_{d+1} \subset S_d$.

2.4 A few criteria that might be useful

This algorithm may be used to build dominating sets when a criterion is the node identification, so the [1] algorithm would be a particular case.
The node degree $d(i)$ (i.e. number of neighbours) may also be used: the criterion may be the couple (node degree, node id) and a total order relation may be defined by:

$$(d(x), x) > (d(y), y) \quad \Leftrightarrow \begin{cases} d(x) > d(y) \\ or \\ d(x) = d(y) \; et \; x > y \end{cases}$$

The residual energy of a sensor in a sensor network may also be a good criterion when building the d-dominating set which is the set of the clusterheads.

3 Cluster formation

In the second section, we proved that the nodes can determine a d-dominating set over the graph, by using a given criterion. To join a cluster x, with a given $c(x)$ clusterhead, nodes must establish a path to reach $c(x)$ provided all nodes in the path belong to the same cluster x.

Therefore, it is necessary to find an algorithm to partition the topology in the connected components, called clusters. In this section we shall study the formation of these clusters.

3.1 On a proposed solution in the literature

In paper [1], the authors proposed a formation of above path, at the end of the formation of the d-dominating set. We have proved that there exist some cases for which the formation of the path is not valid.

Max-Min d cluster formation proposal. The authors of paper [1] proposed the following algorithm to determine the father of each node. The rules are examined in sequence and the algorithm stops for the node x where x be a node of E, as soon as one of the rules is verified.

- *Rule 1:* if $x \in S$, then x is a cluster of which it is the clusterhead and selects itself as father;
- *Rule 2:* Else, if x finds a pair $(v(y))$ in its *Winner* list (i.e. if $v(y)$ appears once in each of the two phases at least), then x selects y as clusterhead [2]. If the node x finds several pairs, it selects the node y whose value $v(y)$ is the smallest, among the found pairs, as a clusterhead. Let $k \in [\![1; d]\!]$ be such as $W_k(x) = v(y)$. x chooses then $S_k(x)$ as father [3].
- *Rule 3:* Else, let the node y be such as $v(y)$=$W_d(x)$. Then x selects y as clusterhead [2]. x selects $S_k(x)$ as father[3].

Therefore, in some cases it is necessary to use an additional rule to make sure that node $p(x)$ the father of the node x and x are in the same cluster $c(x)$. It may be that following the application of the three preceding rules: $c(p(x)) \neq c(x)$. This rule is named *convergecast* in paper [1] and we quote it below:

> "Once a node has identified itself as a gateway node, it then begins to inform (convergecast) its clusterhead by sending a list formed with its node id, all neighbouring gateway nodes and their associated clusterhead to its father. A node uses its SENDER table to determine its father. The process continues with the father which adds its own id to the previous list and sends it to its own father. When the clusterhead has heard each of its neighbours, it knows all the links between it and nodes in its cluster. Moreover it knows all the links between its cluster and the other neighbouring clusters thanks to the data provided by the gateway nodes."

Consequently, the above rule introduces a new condition. It is necessary that: $\forall x \in E \; p(p(x)) \neq x$. In the contrary case, the rule would lead to an infinite loop. However, this condition cannot be observed, as shown in the following example.

On an example where the algorithm leads to a bug. We choose the parameter $d = 5$. The 11 nodes are numbered from 1 to 11. An edge is set between the nodes 11 and 1, 1 and 2, 2 and 3, 3 and 5, 5 and 10, 10 and 4, 4 and 6, 6 and 7, 7 and 8, 8 and 9, 6 and 2. Based on number of the node as the criterion and after application of rules 1, 2 and 3; Table 1 depicts the result of the father and clusterhead selection algorithm. In this example, at the end of rules 1, 2 and 3, the node 3 has node 5 as father and node 10 as clusterhead. However, the node 5 has node 3 as father and node 11 as clusterhead.
Hence, the use of the *convergecast* rule is not possible, as the loop is introduced by the nodes 3 and 5, both of which are gateway nodes also. The next paragraph proves that this phenomenon is due to the use of the *Rule 2*.

[2] We proved in the first part (cf. Th. 1 page 2.), that in this case, the node y is well in $S \cap \mathcal{V}_d(x)$. The application of the *Rule 1* thus makes it a cluster.

[3] By definition, $S_k(x) \in \mathcal{N}_1(x)$, cf. page 2.

Table 1 Max-Min d-cluster formation heuristic applied to the example

	1	2	3	4	5	6	7	8	9	10	11
$Max1$	11	6	5	10	10	7	8	9	9	10	11
$Max2$	11	11	10	10	10	10	9	9	9	10	11
$Max3$	11	11	11	10	10	11	10	9	9	10	11
$Max4$	11	11	11	11	11	11	11	10	9	10	11
$Max5$	11	11	11	11	11	11	11	11	10	11	11
$Min1$	11	11	11	11	11	11	11	10	10	11	11
$Min2$	11	11	11	11	11	11	10	10	10	11	11
$Min3$	11	11	11	11	11	10	10	10	10	11	11
$Min4$	11	10	11	10	11	10	10	10	10	11	11
$Min5$	10	10	10	10	11	10	10	10	10	10	11
$Clusterhead$	11	11	10	10	11	10	10	10	10	10	11
$Father$	11	1	5	10	3	4	6	7	8	10	11

Necessary and sufficient condition for loops to appear. Notice that if a node i is such that $v(c(i)) < M(i)$ then the *Rule 2* was used. Now, the necessary conditions for the phenomenon of loops to appear are investigated. Let us assume that there is a loop and prove that *Rule 2* was used.
Let us consider node i, $c(i)$ its clusterhead and $p(i)$ its father, selected according to the paper [1]. If i and j are two nodes, let us note $d(i,j)$ the distance, i.e. the smallest number of hops between i and j. Now, let x, y and z be the three nodes. If the shortest path between x and y is in k_1 hops and between y and z is in k_2 hops, then the shortest path between x and z is in less than $k_1 + k_2$ hops: $d(x,z) \leq d(x,y) + d(y,z)$.
Then, for any node such that $c(i) \neq p(i)$:

$$d(i, c(i)) = d(p(i), c(i)) + 1$$

since $p(i)$ is the node allowing i to know $c(i)$.
Let i and j the be two nodes such as $p(i) = j$ and $p(j) = i$. i and j are thus not clusterhead since they each one have a different father. The preceding equality applies to i and j:
$d(i, c(i)) = d(j, c(i)) + 1$ and $d(j, c(j)) = d(i, c(j)) + 1$

The following reduction ab absurdo proves that $c(i) \neq c(j)$. Assume that $c(i) = c(j) = l$, then $d(i,l) = d(j,l) + 1$ and $d(j,l) = d(i,l) + 1$ which is absurd, so $c(i) \neq c(j)$.

Let us suppose, without any generality restriction, that $v(c(i)) > v(c(j))$. Node i belongs obviously in the d hops neighbourhood of $c(i)$. Therefore, according to the equality true for all the nodes, $p(i)$ also is in the d hops neighbourhood of $c(i)$, that is to say, here: $j \in \mathcal{V}_d(c(i))$. Thus $c(i) \in \mathcal{V}_d(j)$. So, $M(j) \geq v(c(i))$ and then $M(j) > v(c(j))$. Hence, the *Rule 2* was used according to what precedes.
In other words, the application of the Rule 2, as proposed by the paper can lead to insolvable problems. Then, we continue to search whether the suppression of *Rule 2* could be appropriate and indeed as follows we proved, that by removing *Rule 2* there is no further larger loop problem.
Notice first that the suppression of the *Rule 2* leads to a new property: if the node i is not a clusterhead, then $v(c(i)) = M(i)$ (*Rule 3*).
Let i be a node which belongs to a loop. Without any generality restriction, let us show that a loop with a length 5 cannot occur.
Let j, k, l, m and i be the father of i, j, k, l and m respectively. Since, j is father of i, j belongs to the d hop neighbourhood of $c(i)$. So, $M(j) \geq v(c(i))$. But $v(c(i)) = M(i)$ thus $M(j) \geq M(i)$.
So, $M(j) \geq M(i)$, $M(k) \geq M(j)$, $M(l) \geq M(k)$, $M(m) \geq M(l)$ and $M(i) \geq M(m)$.
It may then be deduced that
$M(i) = M(j) = M(k) = M(l) = M(m)$ then
$c(i) = c(j) = c(k) = c(l) = c(m) = c$. Therefore, it can be written (by applying to each node the general equality $d(i, c(i)) = d(p(i), c(i)) + 1$ since no node among i, j, k, l is clusterhead):

$$\begin{aligned} d(i,c) &= d(j,c) + 1 \\ d(j,c) &= d(k,c) + 1 \\ d(k,c) &= d(l,c) + 1 \\ d(l,c) &= d(m,c) + 1 \\ d(m,c) &= d(i,c) + 1 \end{aligned}$$

which is absurd.
The same kind of demonstration can be applied for any other loop of any given length.
Hence, if the *Rule 2* is removed, which is necessary, there will not be any more loop problems.

The "convergecast" rule is not sufficient to solve the problems.

The following example shows that if the suppression of the *Rule 2* implies that there are no more loops in the algorithm, this suppression does not remove all the problems. Indeed, the following example shows that we can have $j = p(i)$ and $c(j) \neq c(i)$.

Let us use the parameter $d = 2$. Let us consider 5 nodes, numbered from 1 to 5. The edges are between nodes 1 and 2, 2 and 3, 3 and 5, 2 and 4.
The used criterion is the number of the node. The result of the father and clusterhead selection algorithm, after application of rules 1, 2 and 3 is given in Table 2.

It can be noticed that the node 1 has node 2 as a father and is in the cluster 4 whereas the node 2 is

Table 2 Max-Min d-cluster formation heuristic applied to the example

	1	2	3	4	5
Max1	2	4	5	4	5
Max2	4	5	5	4	5
Min1	4	4	5	4	5
Min2	4	4	4	4	5
Clusterhead	4	5	5	4	5
Father	2	3	5	4	5

in the cluster 5. It is not possible to go from sons to fathers and to be sure to go through son's clusterhead before the father be attached to another clusterhead. This appears clearly on the above example when going from the node 1.

Thus this type of problem still exists. The convergecast is thus not a solution to the fact that a node i, such as $c(i) \neq c(p(i))$ can exist.

3.2 Another proposal for the formation of the cluster

Let us start with the clusterheads: if the node i is a clusterhead, after application of the *Rule 1*, then node i informs its neighbours that it is a clusterhead. The neighbours who have not already chosen a clusterhead choose i as clusterhead. Then, they also transmit a message to their neighbours saying that they are at one hop from the clusterhead i. The neighbours of these nodes which did not already choose a clusterhead then will choose i as clusterhead by attaching themselves to one of the neighbours of i and proceed in the same way by informing their neighbours that they are at 2 hops from i. This process is repeated d times so as not to exceed d hops. This mechanism guarantees that there cannot be a loop and that all the connected components, which are the clusters, are trees and that the roots of these trees are the clusterheads.

Conclusion

In this paper, we simplified (cf. Th. 5) the heuristic presented in the paper [1]. Indeed, we proved that a node does not need to check all the list of the second phase of the algorithm to be called clusterhead. It is sufficient for the node to observe the last element of this phase. We generalize this heuristic to any given criterion and not only to the identifier of the nodes. This allows to take into consideration other factors influencing the performance of the network. For example, the energy of a wireless sensor network; a rare resource which needs to be optimized in this type of network, benefits from a hierarchical routing introduced by the determination of clusters with a maximum depth d (cf. paper [5]). We proved the correctness of our generalized heuristics.
In the second part, we gave an example which shows that the cluster formation process proposed in [1] is not always valid. The example leads to a loop, which forbids the cluster formation provided by Amis et al. We proved that such examples are a consequence of the use of a specific rule. We proved that the suppression of this rule is not a solution because it leads to other type of problems. This is an important result since Amis et al. algorithm is well known. We then suggested a correct cluster formation process.
We are now studying several criteria which can be used with the generalised d-dominating set formation. For example, by using the node degrees criterion, we found that the number of clusters is equivalent to the one obtained by using the identifier criterion. The use of the degree criterion is more interesting than the identifier one because the formed cluster only depends on the network topology. Others specific criteria might be interesting too, for specific applications.

References

1. Amis, A., Prakash, R., Vuong, T., Huynh, D.: Max-min d-cluster formation in wireless ad hoc networks. In: Proc. of Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. (2000)
2. Heinzelman, W.B., Chandrakasan, A., Balakrishnan, H.: An application-specific protocol architecture for wireless microsensor networks. IEEE Transactions on Wireless Communications **1**(4) (2002) 660–6670
3. Younis, O., Fahmy, S.: Distributed clustering in ad-hoc sensor networks: a hybrid, energy-efficient approach. In: Proc. of IEEE Infocom 2004, Hong Kong (2004)
4. Ohta, T., Inoue, S., Kakuda, Y.: An adaptive multihop clustering scheme for highly mobile ad hoc networks. In: Proc. of the Sixth International Symposium on Autonomous Decentralized Systems (ISADS 03). (2003) 293– 300
5. Mhatre, V., Rosenberg, C.: Design guidelines for wireless sensor networks: Communication, clustering and aggregation. Ad Hoc Networks Journal, Elsevier Science **2** (2004) 45–63
6. Delye de Clauzade de Mazieux, A., Marot, M., Becker, M.: Construction d'un ensemble d-dominant sur un graphe en utilisant un critère donné. Comptes rendus - Mécanique, Académie des Sciences, Elsevier Science **334**(11) (2006) 669–673