6

On Clustering in Sensor Networks

Michel Marot, Alexandre Delye and Monique Becker
Télécom SudParis
France

Why to build clusters in sensor networks ? Agregating nodes in clusters allows to reduce the complexity of the routing algorithms, to optimize the medium resource by letting it to be locally managed by a cluster head, to make easy the data fusion, to simplify the network management and particularly the address allocation, to optimize the energy consumption, and at last to make the network more scalable. Using clusters allows also to stabilize the topology if the cluster size is large in comparison to the speed of the nodes. This chapter is dedicated to clustering in sensor networks. First, the state of the art is presented, followed by the detailed presentation of one of the best and most cited cluster formation method with its validation and correction. Then, the next parts of the chapter are dedicated to some considerations on cluster modelling. In the last part, a method to assign addresses to the nodes within a cluster is presented.

1. Overview of the state of the art on cluster construction and cluster head election

The following state of the art has been partially established from the work of Deosarkar Deosarkar et al. (2008), Kumarawadu Kumarawadu et al. (2008) and Abbasi Abbasi & Younis (2007) and their colleagues. This subject has been the matter of a huge number of publications, and we do not pretend to be exhaustive. Nevertheless, it gives a good overview of the main problematics.
Cluster formation algorithms can be classified into:

- implicit (the nodes congregate in groups) or explicit (the nodes congregate around a cluster head);
- active (the clusters are the results of the execution of dedicated protocols) or passive (the clusters are formed spontaneously by deducting the information about the network topology by hearing the MAC messages used to transmit the data traffic);
- hierarchical (clusters of clusters) or non hierarchical;
- centralized or distributed, the distributed algorithms possibly being emerging if they allow to obtain from local behaviors a global result which is predictible in a deterministic or a stochastic fashion.

The role of the cluster head may vary from an architecture to another. It is generally the cluster head which manages the cluster (address assignment, possible assignment of the time slots and resources to the nodes, etc.). It is also the cluster head which restransmits the sensed data to the base station, either by sending directly the data to the base station or in a multi-hop fashion by sending the data to other cluster heads which then relay them in turn, or by retransmitting the data to nodes which may be simple nodes and not necessary cluster heads.

At last, the cluster heads generally fusion the data before to transmit them towards the base station. This shows how the cluster head must bear a heavy processing and retransmission load which make it subject to an energy consumption larger than the other simple nodes. During the cluster formation phase, the choice of the cluster heads impacts then a lot the performance of the network.

From the beginning of the first studies on clustering, it has been quickly understood that the burden of this energy consumption had to be spread on the nodes by rotating the role of cluster heads between the nodes: it is the main contribution of the LEACH algorithm (cf. Heinzelman et al. (2002)). The authors of LEACH propose a self-configurable architecture based on clusters minimizing the energy consumption of the nodes. The cluster heads transmit directly the data to the base station, and thus they spend more energy than the other nodes which send the data to their cluster heads closer than the base station. To spread the energy consumption, they propose an algorithm where any node becomes periodically cluster head with a probability which is increased in function of the duration spent for the last time at which they were cluster heads, and chosen so that the average number of clusters is a parameter of the algorithm. As it gives only guarantees in average of the number of clusters and their locations, a centralized version of the algorithm (LEACH-C) is also proposed. It allows to determine the optimal configuration to minimize the spent energy from the exact location of the nodes by using a simulating annealing (the problem is NP-hard). The choice of the cluster heads is done by LEACH in a random way, which may result in a bad spatial distribution of the cluster heads, clusters with inequal sizes, and a non optimal distribution of the energy consumption since this criterion is not taken into account in the choice of the cluster head.

It is the idea to rotate the burden of cluster head between the nodes which did the success of LEACH, but the network performance can still be improved first by letting a sensor to send its message to its nearest neighbor instead of letting it to send it to the cluster head which may be a little bit farer and second by letting a single node to send all the data to the base station instead of having several cluster heads to send parts of the network data to the base station: it is the goal of PEGASIS Lindsey & Raghavendra (2002) which organizes the sensors in a chain, where the nodes, when they have a packet to send, send it to the next node in the chain, which agregates the received data with the ones it must send, and does the same until the data arrive to a sensor which plays in the chain the role of cluster head. This role is rotated in a deterministic fashion between the sensors of the chain: they must play this role in turns. In this case, it is necessarily a multi-hop algorithm.

PEGASIS and LEACH inspired several variants (cf. 1.2), taking into account the residual energy, the possibility of agregation in trees, or combining both LEACH and PEGASIS: Ye et al. (2005), Jung et al. (2007), Satapathy & Sarma (2006), Huang et al. (2007), Tian et al. (2007) (algorithm ECR), Hao et al. (2008), TCCA in Selvakennedy & Sinnappan (2007) Liang & Yu (2005), Handy et al. (2002), Depedri et al. (2003) (LEACH-B), Gupta & Dave (2008), energy-LEACH in Xiangning & Yulin (2007). Yiming & Jianjun (2007), Jang et al. (2007), Qing et al. (2006), LEACH-ET in Lijun et al. (2006), LEACH-F in Heinzelman (2000), Wang et al. (2007), ERA in Chen et al. (2007), Loscri et al. (2005).

The choice of the cluster heads (cf. 1.1) is crucial for the performance of the network. Research about the choice itself or methods to reduce the overhead due to the signalling necessary to renew the cluster heads have been carried, either by letting the cluster heads directly transmit to their successors the role of cluster head based on the information exchanged the first time to choose the first cluster heads, (e. g. Rajiullah & Shimamoto (2007), Nam & Min (2007)), or by using an information transmitted in the data frames (cf. Gerla et al. (2000)). Other methods have been proposed like the Türing morphogenesis (cf. Henderson et al. (2004) et Henderson et al. (1998)), the aim of which being to constraint the shape of the clusters (cf. 1.3). At last, building multi-hop clusters (cf. 1.4) combines both the difficulty of the cluster head election and the one of the cluster construction (routing trees inside the clusters) once the cluster heads chosen.

1.1 On the criterion for the choice of the cluster head

Clusters can be built without any constraint on the node which becomes cluster head. Nagpal and Coore propose CLUBS in Nagpal & Coore (1998) which allows to build single hop clusters. After a random time, nodes broadcast a message advertising they are luster heads. The clusters can overlap.

Considering that the cluster formation algorithm must be simple, Xu and Gerla propose in Xu & Gerla (2002) RCC, for multi-hop networks, which allows each node to elect itself as a cluster head and then to broadcast after a random time an advertisement message, the other nodes joining the first node having sent such an advertisement. This algorithm is more stable than another one using the node degree of the highest address because, in these cases, if a sensor having a larger degree or address comes in the network, the cluster head election mechanism must be executed again. RCC is also extended to the multi-hop case where the sensors relay at most K times a cluster head advertisement. The network is then seen as a set of local networks interconnected by the cluster heads and the authors evaluate the performance of proactive routing algorithms within the sensors and on demand routing algorithms between clusters.

The drawback of LEACH is that, since the nodes elect themselves as cluster heads with a certain probability, it is possible that there be not the same number of cluster heads in function of the time, and even that there be no cluster head at all at certain times. Thus, a criterion is used to determine the cluster head and it is generally the address of the node, the remaining energy or the number of neighbors. The easiest way to build clusters, besides the use of the geographical information, is to choose as cluster heads the nodes which have the smallest identifiers. It is exactly what C.R. Lin and M. Gerla propose in Kwon & Gerla (1999), where they present a distributed algorithm allowing to construct two-hop clusters (where each node can reach any other node in the cluster in at most one hop from the cluster head: the cluster is constituted of the cluster head and all its neighbors). The goal of the authors is to allow the spatial reuse of the bandwidth through clustering, to control the bandwidth within each cluster and to have a more stable topology. To build two-hop clusters allows to reuse the power control algorithms developed in the context of cellular networks (cf. Lin & Gerla (1997)).

In Rajiullah & Shimamoto (2007), Rajiullah and Shimamoto propose to decrease the traffic and processing load necessary to renew the choice of the cluster head by letting the first cluster heads to decide which nodes will be cluster heads later. A cluster head, as soon as it has reached a low energy threshold, pass the baton to the next one, which advertises its neighborhood that it becomes the new cluster head. The nodes under its coverage radius choose then the nearest cluster head, on the basis of the strength of the signal they receive. The criterion determining the choice of the cluster head is the smallest node identifier. The same idea is proposed by Nam and Min in Nam & Min (2007) where LEACH is used to build clusters at the beginning, but the cluster heads designate then themselves their successors which are chosen each one in its turn within theirs clusters. The criterion of the choice of the cluster head is the address identifier. Liu, Lee and Wang, in Liu et al. (2007) use also the node identifier. This algorithm is described below.

Tillapart and his colleagues present in Tillapart et al. (2005) LMSSC: a method to partition the nodes in clusters which consists in defining the clusters on the basis of the highest degree nodes, but the cluster heads are chosen in a centralized way by the base station by minimizing the criterion of the ratio of the remaining energy of a node over the sum of the squared distances of this node to all the others plus its squared distance to the base station.

Chang and Kuo propose in Chang & Kuo (2006) an algorithm for the choice of the cluster heads, MECH, where all the nodes send "hello" messages, which are not retransmitted, and the nodes which receive a number of such messages at least equal to a certain threshold elect themselves as cluster heads. It is thus the node degree which is used here as the selectino criterion. The same idea is used in Wen & Sethares (2005) where the nodes declare themselves cluster heads if they do not have one in their neighborhood and after the expiration of a time out randomly generated and decremented at each reception of a "hello" message, these messages being periodically sent by each sensor. Kim et al. (2008) or Chan & Perrig (2004) described below use also the node degree.

The criterion for the choice of the cluster head can also be based on a weigth function of several such criteria (cf. DWEHC in Ding et al. (2005), HEED in Younis & Fahmy (2004), Fan & Zhou (2006), WCA in Chatterjee et al. (2001), Li et al. (2006)). Using weights allows also to take into account the speed (like DMAC in Basagni (1999b), Basagni (1999a), Basagni et al. (2004), Bettstetter (2004), or Chinara & Rath (2008)).

In Ding et al. (2005), Ding, Holliday and Celik propose a cluster formation method, DWEHC, where the sensors elect themselves cluster heads if they have the highest weigth in their one hop neighborhood, which is the product of their remaining energy and the average of their distances to their neighbors. The cluster heads broadcast then an advertisement message and the sensors join the nearest candidate cluster head, in a multi hop way up to the limit of the size of the clusters. In Klaoudatou et al. (2008), the speed is used a criterion (cf. below).

Generally, cluster formation algorithms are made of a cluster construction phase and a maintenance phase, which is particularly important when the nodes are mobiles. Often the first phase assumes that the nodes are almost motionless. To relax this assumption, S. Basagni (cf. Basagni (1999b) and Basagni (1999a), cf. also Basagni et al. (2004) and Bettstetter (2004) for considerations on the performance) proposed an algorithm (DMAC) which associates a weigth to each node and so that the cluster heads are the highest weight nodes and never can be neighbors. The weight may be dependent on the speed or the power level.

These criteria are generally transmitted in a dedicated signaling, but to reduce the overhead due to the control information conveyed to build the clusters, M. Gerla, T.J.Kwon and G. Pei (cf. Gerla et al. (2000)) propose and algorithm using a minimal information transmitted in the data MAC frames. Listening to all the node traffic in their neighborhood allows each node to get this information. This algorithm is suitable in situations where the nodes are very mobiles.

1.2 Variations in LEACH or PEGASIS modes

After LEACH a multitude of variants have been published, like in Hao et al. (2008), where the base station receives the geographical positions of the nodes, and then it partitions the network into different geographical zones which become clusters. LEACH is used in this paper to choose the cluster heads within the clusters. Between the cluster heads, a multi hop mechanism is used to transmit the data. Generally speaking, all the variants are more or less motivated by the necessity to take into account the remaining energy of the nodes in the probability for a node to be elected as a cluster head. In Selvakennedy & Sinnappan (2007) TCCA is proposed, for which the clusters are multi hop ones. Liang & Yu (2005), Handy et al. (2002), Depedri et al. (2003) (LEACH-B), Gupta & Dave (2008), Xiangning & Yulin (2007), Yiming & Jianjun (2007), and Jang et al. (2007), Qing et al. (2006) (which adaptes LEACH to the case where the sensors have initially heterogeneous energy levels) may also be cited. Xiangning & Yulin (2007) propose energy-LEACH which selects as cluster heads the nodes which have a remaining energy level higher than a given threshold and multihop-LEACH whose behavior is the same as LEACH but with the possibility to communicate in multi hop from cluster heads to cluster heads towards the base station. Jang et al. (2007) weights, as the other algorithms, the pobability to become cluster head with the remaining energy, but first this weight intervenes only when the nodes have consummed 50% of their energy otherwise the operation is exactly the same as LEACH, and, second, it also defines a cost function used by any simple node to choose its cluster heads, which is a function of the received signal strength as in LEACH but also of the remaining energy of the cluster head.

In Lijun et al. (2006), the authors propose (LEACH-ET) to trigger the cluster head changes only when a node has reached a given energy threshold or if a cluster head has emptied its battery instead of doing it periodically. In Wang et al. (2007), the authors, noticing that the number of nodes varies in function of the time, propose to dynamically adapt the probability to become cluster head in function of the actual number of sensors in the network. Another LEACH variant aiming at prolonging the network lifetime is proposed in Chen et al. (2007): ERA. The cluster heads are chosen as in LEACH, but the nodes, instead of joining the nearest cluster heads (which has the highest received signal strength), join the cluster head for which the remaining energy on the path (remaining energy of the nodes on the path minus the energy necessary to join the base station) is the highest. At last, Loscri and his colleagues extend LEACH to a two level hierarchy in Loscri et al. (2005). This increases the efficiency of the data fusion and thus the economized energy, if the base station is far from the network.

PEGASIS presents several problems, and particularly to take into account neither the remaining energy of the nodes in the choice of the cluster heads nor the distance to the base station: it is the sensor $i \bmod N$ which is chosen at the i^{th} round as a cluster head. Moreover, the greedy algorithm used by PEGASIS, by nature, can lead to non optimal chains, the total agregation time may also be very long. That is why the authors of Jung et al. (2007) propose that the base station broadcasts a set of thresholds which corresponds to different signal levels and which define reception zones around the base station. The sensors, in function of these received thresholds, can then be distributed in concentric zones around the base station. Within each zone, the sensors apply PEGASIS: they constitute a chain internal in the zone where they play the role of cluster heads each one in its turn and agregate the data from neighbors to neighbors in the chain. However, the cluster head of the zone i agregates its data with its own data which are then transmitted in a single packet to the cluster head of the zone $i-1$.

An idea similar to PEGASIS is presented in Satapathy & Sarma (2006) where the use of trees replaces the chain of PEGASIS. As soon as the root of a tree dies because of an empty battery, it is changed. The use of a tree necessitates more than a fusion operation, and thus less packet transmissions, and thus an energy gain. At last, the authors of Huang et al. (2007) propose to combine the advantages of the use of clusters in LEACH with the advantages of the trees presented in Satapathy & Sarma (2006) by using several clusters constituting each one a tree. The cluster head is chosen the closest possible to the base station (all the sensors are assumed to know their geographical positions) under the constraint of a minimal remaining energy.

Tian, Wang and Zhang propose in Tian et al. (2007) an algorithm, ECR, which combines the advantages of LEACH and PEGASIS: it allows to have several clusters at the same time to decrease the latency due to the agregation, to use chains within the clusters and between the cluster heads to perform the agregation and at last it uses the remaining energy of the nodes to select the head of the cluster heads (i.e. the cluster head of the chain of the cluster heads, that we call protocaryomme in the following). The sensors are assumed to know their relative positions, a coordinate system of which the "Y" axis is so that the base station if far from the network in this direction and the number N of clusters. The clusters are defined in bands parallel to the X axis knowing N and the sensor broadcast their identifiers, their cluster identifiers and their positions. They can then constitute in a distributed fashion the chain within their clusters. The base station assigns the role of protocaryomme to a sensor which is by the way the cluster head of its cluster, and a greedy algorithm is used to constitute the chain of the cluster heads from the protocaryomme. Besides the data, each node inserts the maximum between its remaining energy level and the one it has received from its neighbor, with the identifier of the node corresponding to the retained energy level. One thing leading up to another, the packet arriving to the base station contains the identifier of the node having the highest remaining energy level which is then elected as the protocaryomme.

1.3 On the methods constraining the shape of the clusters in terms of numbers of nodes, laying out, etc

The drawback of LEACH is that, as the nodes elect themselves cluster heads with a certain probability, it is possible that there be not the same number in function of the time, and even that there be no cluster head at all. To solve this problem, O. Younis and S. Fahmy (cf. Younis & Fahmy (2004)) propose the HEED algorithm which allows to select a cluster head in function of its remaining energy and a cost function defined, depending of the target objectives, either on the number of neighbors or on the average of the minimal power necessary to be reached by the neighbors. Either very dense clusters or clusters with a well distributed load can thus be obtained. In Fan & Zhou (2006), partly inspired from WCA presented in Chatterjee et al. (2001) and which does not take into account the residual energy of the node, the cluster heads are chosen with weights functions of the inverse of the node residual energy, their degree, the sum of the distances to their neighbors and the distance to the base station. This function, when it is minimized, leads to choose sensors having the highest residual energy, having a degree as close as possible to a value which is a parameter of the algorithm and minimizing the distance between the nodes and the base station. A similar intuition leads the authors Li et al. (2006) to propose an algorithm where the cluster heads are chosen by maximizing a cost function of the residual energy, the number of neighbors and the time spent for the last time the node was cluster head. Initially, the base station defines the perimeter of the clusters and chooses the first cluster heads, but, later, the clusters pass the baton by choosing themselves the next cluster heads by taking the nodes which maximize this function in the clusters. Then the new cluster heads send an advertisement message and the nodes join their new cluster heads in function of the signal strength level. The authors of Guo et al. (2007) propose to extend HEED to the case where the routing between cluster heads is in a multi hop fashion to the base station (CMRP algorithm).

Gupta and Younis consider in Gupta & Younis (2003b) an heterogeneous network of which the cluster heads are the nodes having no energy constraint and which can all communicate together. To build the clusters, they discover their neighbors (i.e. the sensors for which they are in visibility), and then they distribute them between them in order to minimize the total transmission cost of the sensors to their cluster heads and to distribute almost evenly the number of sensors. It is an iterative process where a cluster head attributes itself the sensors which are in its coverage progressively increased by the minimum of the distances between the cluster heads and its neighbors to the median of the distances. All the nodes are equiped with a GPS. In the same context, in Gupta & Younis (2003a), the same authors address the issue of the cluster head failure. The network uses a TDMA like transmission mechanism for which some slots are dedicated to the cluster heads to communicate their status. When all the cluster heads have no more information about one of them, they distribute its sensors between them. For this purpose, any cluster head has two lists: a list of sensors of its own cluster and another list of other sensors for which it is the backup cluster head. The first list is obtained according to the method proposed in Gupta & Younis (2003b), the second one is obtained with a simple visibility condition between the cluster head and a sensor.

In Klaoudatou et al. (2008), Klaoudatou et al. consider medical surveillance sensor networks of which the nodes are mobiles. They select the closest node to the base station (in ad-hoc environment during the emergencies on the spot or using access points in the hospital) as a cluster head. Actually, they notice that the mobility allows then to turn this role of cluster head between the different sensors. Chinara & Rath (2008) considers also the case of mobile sensors. They estimate then their speed during the last time period and the least mobile ones are chosen as cluster heads.

Liu, Lee and Wang, in Liu et al. (2007), propose two algorithms. The first on, ACE-C (Algorithm of Cluster head Election by counting), aims at determining the cluster head on the basis of the node identifier: there are N nodes in the network, C cluster heads are required in the network, a node x is then a cluster head all the N/C periods. At the beginning of a new period, a cluster head broadcasts a message to all the nodes advertizing it becomes cluster head and containing its geographical position and its speed vector and the others choose the nearest cluster head. For this purpose, they estimate their relative distance from the position of the cluster heads between the time of the current election and the previous one. The use of the speed vector is not clear in the paper. If the battery of a node is empty when it must become cluster head, all the nodes are informed and the nodes integrate that information in their calulations. This algorithm having the drawback of a possible bad distribution of the cluster heads, a second one is proposed: ACE-L (for "Localization"). Fix anchors are distributed in the network. Any node evaluates its distance to the anchor, which is used to proportionately generate a time out after what the node emits a message advertizing it is a cluster head. The first emitting node is the closest one to the anchor and it becomes then a cluster head.

Another proposal is given by Kim and his colleagues in Kim et al. (2008) to distribute in the middle the cluster heads, that is to avoid that the cluster heads be grouped at the same place. A predifined cluster head number is chosen at the network initialization, possibly misplaced. Each cluster head broadcasts under its coverage an advertisement message. Any node receiving it counts the number of received messages. The cluster heads choose then the cluster head in their coverage which should replace them either by designating a node having received few advertisement messages if the cluster is sparse or, contrary, a node having received a large number of such messages if the cluster is dense. This causes a repulsion effect between cluster heads which tends to give a homogeneous coverage of the network by the clusters. The cluster head selection criterion is then the number of cluster heads in the coverage before the new election.

H. Chan and A. Perrig, in Chan & Perrig (2004), propose a similar algorithm which allows to obtain perfectly homogeneous clusters by minimizing the overlaps, of which the complexity depends only on the sensor density. It then counts the number of loyal followers, that is the number of nodes which would have only it as a cluster head if it became a cluster head. If this number is larger than a certain threshold, it becomes cluster head. By so counting the number

of loyal followers, and not only the number of sensors able to belong to several clusters, the chosen candidate cluster head is the one for which the cluster has a minimal cluster overlap. This causes a repulsion effect between clusters, and thus a better distribution of the clusters.
Another proposal aiming at avoiding a non even distribution of the clusters in LEACH is presented in Ye et al. (2005). The candidate cluster heads elect themselves with a fixed probability T, they broadcast an advertisement message, which contains a residual energy level. If such a candidate receives such a message for which the level is greater than its one, it effaces itself, otherwise it proclaims cluster head. An ordinary node joins then the cluster head which minimizes a cost function taking into account its distance to the cluster head and the distance between the cluster head and the base station.
The BCDCP algorithm presented by Muruganathan et al. in Muruganathan et al. (2005) consists in selecting, among the ones having a residual energy greater than the average two nodes which have a maximal distance between them, in distributing between them the nodes of the network in a manner as even as possible and in iterating this process until the desired number of cluster. This allows to ensure there is well the desired number of cluster heads with almost the same number of sensors in each cluster. The nodes have power levels which can vary and they transmit directly their data to their cluster heads which fusion them and send them to the base station from cluster heads to cluster heads. This partitioning and cluster head election algorithm is centralized at the base station.
In ya Zhang et al. (2007), the clusters are obtained by the base station with the algorithm of the k-means for the classification of the nodes in clusters. The choice of the cluster head is done by minimizing the distance between the nodes and the cluster head (this distance is also minimized in the classification) at the beginning, then the clusters remains the same all along the network lifetime, but, periodically, the node having the highest residual energy in the cluster replace the cluster head. The idea is thus to build "natural" clusters corresponding to the node agregates. There is a predifined number of clusters but also a limit threshold for the cluster size which allows to split them into several clusters if they have reached this limit. The idea to build the clusters at the beginning and to leave them after without changing them but to only turn the cluster head role between the nodes of a same cluster is also proposed in an evolution of LEACH-C: LEACH-F (cf. Heinzelman (2000)) which uses at the beginning of the network lifetime the same method as LEACH-C.
Demirbas and his colleagues present FLOC in Demirbas et al. (2004). The nodes can communicate according to two modes: in i-band, a reliable manner, in the limit of a certain unit radius and in o-band beyond this radius but in non reliable mode and still within the limit of another larger radius. The nodes elect themselves candidate cluster heads after a random time and broadcast then an advertisement. If a sensor receives this message and if it is already in the i-band of another cluster head C, the candidate renounce its pretension to become cluster head and it joins C possibly in o-band mode. If a sensor receives this message and if it is in the i-band of the candidate but also in the o-band of a cluster head C, it leaves C to join the candidate. This proposal aims to guarantee clusters having the "solid disk" property: all the nodes at a unit distance of a cluster head are in its cluster or, in other words, there is no overlap of unit radius clusters. This allows to bound the number of clusters, to decrease the signaling (a cluster head has not to listen to all the sensors which are in its coverage but which belongs to other clusters), to obtain a better spatial coverage for the data agregation, etc.
In Zhang & Arora (2003), Zhang and Arora assume the sensor to have a perfect knowledge of the geography and they constitute hexagonal cells. A root node finds its ideal position from the center of its neighbors cells and selects as cluster heads of these cells the closest

node to its ideal position. If there is no such a node (if the coverage radius is to small), the sensors of the cell are distributed among the neighboring cells. The underlying motivation for this perfectly geographical hexagonal partitioning is multiple: numerous sensor network applications give identical results per geographical zones, easy compression by geographical zones, better frequency reuse, etc.

The idea to spread the clusters according to a partition can be extended to a non geographical space. Actually, the notion of cluster is still more important when the agregation (data fusion) is taken into acount. In Vlajic & Xia (2006), the authors propose a cluster grouping based on the similarity of the sensed data: The nodes which sense the same physical characteristics are naturally grouped allowing a maximal compression per data fusion. They propose then in Xia & Vlajic (2006) an algorithm, LNCA, for the multi-hop cluster formation consisting for each sensor in listening to the data transmitted by their neighbors. If the data are the same as their own data, they increment a counter and they insert the neighbor into a list. They broadcast then this counter with a time to live field *n* to limit the retransmission of the message to *n* hops and it is the node which has the highest value of this counter which is retain as a cluster head.

An original idea has been proposed by T.C. Henderson and his colleagues in Henderson et al. (2004) and Henderson et al. (1998). It consists in using the Türing's morphogenesis process to give to a very dense network a certain configuration. The idea consists in propagating the result of a certain function from sensors to sensors, this result being used in input of the function on the next sensor. By well choosing the function, a mechanism can be implemented to initialize a variable producing a totally predetermined global configuration. This method is expected for example to radio control robots. If the number of sensors is very large on a given surface, with a certain function, bands can be drawn which can be used as traces to guide robots. This morphogenesis could be used to find more complex configurations.

1.4 The multi-hop case

Apart the cases where the nodes group themselves by affinities (for example on the basis of similar sensed data like in Vlajic & Xia (2006) or LNCA in Xia & Vlajic (2006)) or implicitely like in Kawadia & Kumar (2003), or in a centralized fashion (like the extension of BCDCP, also centralized allowing multi hop communications in clusters thanks to routing trees within the clusters in Huang et al. (2006)), the multi hop cluster formation is doubly complicated: first the question is raised how to choose the cluster heads and, second, how to build the parentage between the ordinary nodes and their cluster heads.

In Kawadia & Kumar (2003), V. Kawadia and P.R. Kumar propose an algorithm integrating routing, power control and implicit clusterization, CLUSTERPOW and tunnelled CLUSTERPOW, for networks of which the node distribution is homogeneous. It is a multi hop routing algorithm where each node has several power levels and where it chooses the smallest possible one to reach its destination. Each power level defines then a cluster: to reach a far destination, the node must send the information by using its largest power level, which is the same to transmit to another cluster when the network is not evenly distributed.

Some approaches not considering the choice of the cluster heads aim only to split the whole network into clusters. Some consist in building spanning trees which are later split into subtrees, the important task being the good distribution of the clusters: Banerjee & Khuller (2001), Fernandess & Communication (2002).

Banerjee and Khuller serach in Banerjee & Khuller (2001) to constitute clusters of which the size is between k and 2k, except a single one allowed to be smaller, and such as the number of clusters a sensor belongs to is bounded. For this purpose, they build spanning trees on the network and, from the leaves of the tree, they take sub-trees with size between the two bounds. Two versions, centralized and distributed, are proposed. The problem of the cluster head election is not really the main concern of the authors.

In Fernandess & Communication (2002), the authors propose to make a partition into k hop clusters by building a minimum connected dominating set. They obtained next a spanning tree from this set. They add as leaves the nodes of the remaining part of the graph. This tree is later split into sub-trees with a diameter k. Building such a spanning tree gives more balanced clusters than other known techniques.

The same goal is targeted in Youssef et al. (2006) (algorithm MOCA). Youssef and his colleagues (among who there is Younis) put in Youssef et al. (2006) the problem of the necessity to have overlapping clusters, in order to facilitate the routing between clusters (among other reasons), and they define the concept of k-dominating set with overlap: any node is at most at a k hop distance and belongs to at least two clusters. The cluster heads elect themselves with a predetermined probability, and they broadcast an advertisement message, which is retransmitted at most k times. A node receiving this message answers even if it already belongs to a cluster. A sensor can thus belong to more than two clusters at a time. Note that it is always possible that the nodes be isolated and thus belong to only a single cluster; their own. It is the MOCA algorithm.

In Dai & Wu (2005), Dai and Wu propose three algorithms to build a k-connected k-dominating set. It is a set such as first any node is in this set or has at least k neighbors inside, and, second, if $k - 1$ nodes are removed, it remains connected. For the first algorithm, each node elects itself as a member of the k-connected k-dominating set with a given probability p. For example, with 200 nodes spread over a 1000×1000 surface and with $k = 2$, $p = 50\%$, this process leads to a 2-dominating set with a probability 98,2%. The second algorithm is deterministic and it consists in removing each node of the k-connected k-dominating set if there exists k disjoint backup paths between every couple (u, v) of its neighbors, via nodes having greater identifiers than v. The third algorithm combines both approaches: it consists for any node to be colored with a certain probability with a color given among k ones, and the deterministic condition is applied but between the nodes of a same color. The cluster heads are arbitrary chosen. This proposal aims to ensure a certain reliability.

The solution of Dai & Wu (2005) rather aims to ensure a certain reliability, but the approach aiming to build independant k-dominating sets is more suited to sensor networks because it leads to a more efficient use of the energy, at the expense of a certain lesser reliability. The work presented in Banerjee & Khuller (2001) and Fernandess & Communication (2002) are methods to partition a graph, but not to elect a cluster head from a given criterion, contrary to the papers McLaughlan & Akkaya (2007) and Nocetti et al. (2003). Nevertheless, in the case of these papers, clusters mades of nodes separated from their cluster heads by paths containing nodes belonging to other clusters can be obtained! To avoid that, Prakash and his colleagues propose in Amis et al. (2000) a heuristic which allows to build k-dominating sets using the address of the nodes as a criterion and made of two phases. The first one is analogous to the classical step of the broadcast of the highest value of the criterion in a d hop neighborhood. The second one consists in broadcasting in a k neighborhood the minimum of these maximums. That allows the cluster heads having not the highest value of the criterion in their k-neighborhood, and thus separated from their members by nodes belonging to other clusters, to gain members in their clusters.

Nevertheless, the choice of the cluster heads impacts the performance and should no be neglected. The simplest method is the one where each node elects itself as a cluster head independantly of its neighbors: for example with a certain probability (cf. Xiangning & Yulin (2007), RCC in Xu & Gerla (2002), Bandyopadhyay & Coyle (2003), EMCA in Qian et al. (2006), Wang et al. (2005), SWEET in Fang et al. (2008), McLaughlan & Akkaya (2007)), and then broadcasts messages which are retransmitted k times at maximum. In McLaughlan & Akkaya (2007), each node diffuses "alive" messages to its k hop neighborhood. The sensors elect cluster heads themselves with a probability which is decreased with the proximity of a cluster board (i.e. the board of the k hop neighborhood of a cluster head) and is increased with the number of neighbors. Then they broadcast to their k hop neighborhood a "dominator" message which, when it reachs a node situated at exactly k hops, triggers this later node to send a "board" message. This message allows the other sensors to determine their proximity to a cluster board.

In Bandyopadhyay & Coyle (2003), the authors propose a multi-hop algorithm where the sensors also elect themselves as cluster heads with a given probability p, then they advertize they are cluster heads. These advertisement messages are retransmitted at most k times. The authors calculate p to optimize the energy consumption in the system. k is fixed with a relationship obtained from the stochastic geometry and which is a function of the probability that the radius of a sphere centered on the cluster head and containing its Voronoï cell be larger than a certain value $r \times k$. An extension of LEACH to the multi-hop case (for the transmission between a sensor and its cluster head) is proposed in Qian et al. (2006): EMCA. The cluster heads are chosen in the same way as in LEACH. Then, they broadcast a message advertizing they are cluster heads. This message is retransmitted a given maximum number of times. A MAC method for the TDMA slots is also proposed.

The authors of Wang et al. (2005) propose a multi-hop cluster formation algorithm oriented towards the attributes. To make easier the data query, the clusters are first geographically defined and second they are defined within a same geographical zone by attributes (temperature, pressure, concentration, age,...). A cluster hierarchy embedded into each others is then defined, each cluster having its own cluster head: the hospital, the floor i of the hospital, the room j of this floor, the pressure sensor k of this room, etc. At the beginning, a node advertizes it is a general cluster head then this information is retransmitted through all the hierarchy by the others after a certain time which is a function of the residual energy. After this random delay, a sensor receiving this information advertizes it is a cluster head if there is still no cluster head in the hierarchy. The cluster heads transmit then the information of the composition of their clusters to the cluster head of higher level, which also gives a routing information used during the query. The idea to announce to be a cluster head after a certain random time inversely proportional to the residual energy is also proposed in Fang et al. (2008) (SWEET). A method proposed to be more efficient constists in comparing between the sensors a certain criterion: node identifier, residual energy, weights, etc. (cf. KHOPCA in Brust et al. (2008), CABCF in Liu et al. (2009), Rasheed et al. (2007), MaxMin in Amis et al. (2000),...).

Variants are proposed but, finally, the same method is always used: either a node elects itself with a given probability and it broadcasts an advertisement until k hops or it broadcasts weights until k hops. In Brust et al. (2008) (KHOPCA), Brust and his colleagues propose a mechanism which consists in decrementing a weight or changing it from MIN to MAX values depending on the values of the neighbors weights. This causes the weights to be spread so that they be separated by a good number of hops. the change from MIN to MAX is done in function of the neighboring weights, and thus not depending on a criterion like the energy of the node degree. In Liu et al. (2009), the authors propose CABCF where each node has a weight function of the residual energy, the degree and the distance to the sink. The nodes are then grouped into clusters step by step by combining themselves with larger weight sensors. The multi-hop communication is also set up by using this heuristic within the clusters.

It is possible that two nodes have the same criterion value. For this situation, the authors of Nocetti et al. (2003), propose an algorithm which consists in that the sensors having the highest degree and the smallest address elect themselves as cluster heads and broadcast an advertisement untill *k* hops.

This simple *k* hop broadcast is omnipresent in the literature, for example in Rasheed et al. (2007). but it is a problem because of the interdependance between the *k* hop neighborhoods. Actually, when building multi-hop clusters, the question arises sooner or later to know how to let a maximum distance between the cluster heads while ensuring any ordinary sensor to be at most at *k* hops wide a cluster head, that is how to build an optimal *k*-dominating independant set. Unfortunately, to find such a set is an NP-hard problem (cf. Amis et al. (2000)), that is why heuristics have been proposed.

It is intuitive that the nodes having the highest criterion value be elected cluster heads. There are two ways to implement that. Either the nodes exchange this criterion information so that each node gets the list of its neighbors and their criterion values, or a node broadcasts the couple of its identifier and its criterion value which is retransmitted by its neighbor if its own value is smaller or after having replaced the received value by its own if it is larger. In this case, all the nodes have finally a single information: the identifier of the node which has the highest value of the criterion in its *k*-hop neighborhood with this value but no more information on the neighborhood.

The drawback of the first approach is that some nodes become orphans and have no other solution than proclaim themselves cluster heads. Actually, let us consider the weights given on figure 1 and let us assume two-hop clusters. Applying this method leads nodes 5, 4 and 3 to know that the node 5 has the highest criterion in its two-hop neighborhood. Neither 4 nor 3 broadcasts any cluster head advertisement, but 5 does it. 5 is thus a cluster head of the cluster (5,4,3). 4 has not broadcast any cluster head advertisement, the same for 3 and 2, because it noticed it did not have the highest criterion value. The result is that 2 becomes alone. The only solution is to declare 2 cluster head of the cluster containing the only node (2). If there should be a cluster with only one node, it would be more intelligent to choose (5) and (4,3,2). In short, the more appropriate candidate in the neighborhood of 2 does not declare itself as a cluster head because it already belongs to another cluster but any node (e.g. node 2) counts on the node having the highest criterion value in its neighborhood (e.g. node 4) as a cluster head.

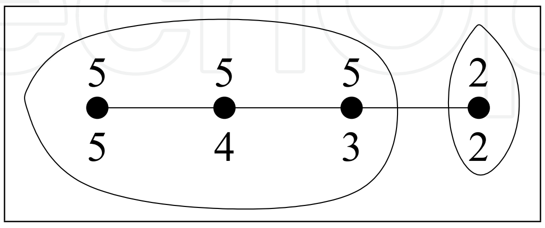


Fig. 1. Case of a bad cluster head selection

In the second case, where a single couple of identifier and criterion is broadcast and possibly overwritten by a node having a higher value, the choice of the cluster head leads to that each node A elects necessarily as a cluster head the node B which has the highest criterion in its *k*-hop neighborhood. Nevertheless, it is possible that B itself has already elected another node

C in its own *k*-hop neighborhood but not in A's neighborhood because C has a criterion value higher than the criterion value of B. In this case, a sensor elects a cluster head which does not consider as such a cluster head. On the example of figure 1, 2 would choose 4 as a cluster head which itself would choose 5.

To summarize, either a node does not elect its cluster head but it waits for that another node anounces itself as a cluster head, with the risk that this one is already a member of another cluster that is with the risk to be without cluster head and then to be obliged to be cluster head with a small criterion value, or it decides to elect another node with the risk that this latter is already in another cluster and thus the risk that it is a follower of a node which is not a cluster head. The whole problem comes from the interdenpendance of the *k*-hop neighborhoods which makes it NP-hard. To give a heuristic is exactly to distribute this problem by relaxing the independence and thus it is exactly to accept either a non optimality or inconsistencies. This fundamental problem has not really been considered in the literature. Scientists have focus their research mainly on finding a good criterion rather than on the method without realizing that an optimal criterion with a bad method could lead to a disastrous performance or to functional problems. It was urgent to consider this problem.

Prakash and his colleagues proposed then in Amis et al. (2000) a heuristic allowing to build *d*-dominating sets with the criterion of the node identifier and made of two phases. The first one is analogous to the classical broadcast of the highest criterion value in a *d*-neighborhood with overwritting. The second one consists in doing the same thing than in the first phase but by transmitting in a *d*-neighborhood the minimum of the exchanged values instead of the maximum. This gives to the cluster heads having not necessarily the highest value, and thus the cluster heads separated from their members by other nodes belonging to other clusters, to gain new members. This allows to solve the problem of the nodes having as cluster heads others which do not consider as such. On the example of figure 1, this algorithm leads to two clusters (5) and (4,3,2).

Of course, it would be naive to think that a NP-hard problem could be solved in a so simple way! this algorithm, by accepting that the minimum of some maximums are chosen, accepts not to be optimal, but since this minimum is chosen among maximums, the performance remains good. Nevertheless two other problems appear. First, as the algorithm has two steps, a phase where the maximums are exchanged until *d* hops followed by another one where the minimums are exchanged, it is possible to have a cluster head two hops away. Moreover, it is still possible that a node is separated from its cluster head by a father which belongs to another cluster. It is thus necessary to add rules after the phases "Max" and "Min" to avoid that.

The authors of Amis et al. (2000) decide that a node which finally received its own identifier at the end of the algorithm decides it is a cluster head: it is the rule 1. This node has then the highest criterion value in its *d*-hop neighborhood. They want also that, if a node does not find its identifier, and thus that another node would be a better cluster head, this node be chosen under the condition that it is in its *d* hop neighborood, and thus that it appears also during the "Max" phase. The node chooses then as a cluster head the node which appeared in both "Min" and "Max" phases, but, for reasons of a better balancing of the number of sensors in the clusters, they impose also that it is the smallest pair which is chosen if several are possible (because the algorithm tends to favor the cluster heads having the highest criterion value): it is the rule 2. At last, if it is in none of both preceding cases, a sensor chooses as a cluster head the node which appeared at the end of the "Max" phase: it is the rule 3.

This solution seems to solve enough problems to give satisfaction. Unfortunately, no validation has been given. In the next sections this heuristic is formally evaluated and it is shown how it still poses a problem. Nevertheless interesting lessons are drawn by this study and solutions are proposed.

2. The Maxi-Min d-cluster formation: election of cluster heads

The deployment of hierarchical sensor networks organized in clusters is of highest importance for applications requiring several hundreds of sensors. This actually allows to set up scalable protocols. Amis et al.'s proposal allows to build multi-hop hierarchical clusters with a bounded depth. The set of the cluster heads constitutes then a d-dominating set on the graph of the network. This notion is formalized in the following paragraphs.

Let $\mathcal{G} = \{V, E\}$ be a graph where E is the set of the edges and V the set of the vertices. In this context, the cluster heads constitute a subset S of V which is *d-dominating* with respect to the graph $\mathcal{G}$. A subset S of V is *d-dominating* when any vertex in E can join a vertex in S via edges in E in less than d hops. Amis et al. have proved that for $\mathcal{G}$, d and an integer k given, it is difficult to know if there exists a set of d-dominating subsets with a size smaller or equal to k. More precisely, the authors have proved that this problem is NP-hard. They propose an algorithm, the "Max-Min d cluster formation", which allows to build a d-dominating set and the tree associated to each cluster head.

To date, this algorithm is one of the very rare ones to propose a wireless network organization as multi-hop clusters and it is very important as already said in the previous section. Moreover, this algorithm is noticeable because the nodes exchange only few informations to build the d-hop dominating set. More precisely, the algorithm is divided into two steps. The first one allows to choose the d-dominating set and to let the simple nodes to know their cluster heads. The second one allows each node to know which node is its father, i.e. to know how to join its cluster head [1]. We first look at the selection of the d-dominating set, that is at the first part of the algorithm proposed by Amis et al. The clusters built with this algorithm depend on the addresses of the nodes. the cluster heads have often[2] the highest address. This means that the clusters formed by the algorithm are not the same for two networks which differ only by their node addresses. Moreover, there is no reason to select cluster heads in function of their addresses and it would certainly be more intelligent to use other criteria. Other criteria could be the node degree, its residual energy, etc. This led us to generalize the first part of this algorithm in order to build clusters of which the cluster heads have often the highest chosen criterion. The criterion becomes thus a parameter of the algorithm, as the maximal depth d. It is this generalized version which is presented here.

2.1 Notations and introduction to the algorithm

This part is extending the results published in CRAS Delye de Clauzade de Mazieux et al. (2006) (Compte Rendu à l'Académie des Sciences).

Let $\mathcal{G} = \{V, E\}$ be a graph with sets of vertices V and edges E . The 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 at least one member of S through a path of d edges in E.

[1] In fact, there is a misteake in this second part, as it will be shown in the next sections

[2] This notion will be specify later, see equation 1, p. 18

Let us consider $x \in V$, $\mathcal{N}_i(x)$ is the set of neighbors 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 denoted v^{-1}: $\forall x \in V \quad v^{-1}(v(x)) = x$.

The presented algorithm (cf. Delye de Clauzade de Mazieux et al. (2006)) generalizes the one proposed by Amis et al. 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 denote $W_k(x)$ 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 i given criterion) among its d hop neighbors ; a second phase, the Min phase, lets a node know whether it is a dominating node for one of its neighbor nodes. If it is the case, this node belongs to the S set. For a given criterion, the only dominating set is built from this very 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 2.1. *Let S be the set defined by:*
$S = \{x \in V, W_{2d}(x) = v(x)\}$ [3]

Theorem 2.1. *Each node $x \in V \setminus S$ can determine at least one node of S 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 at least once in each of the two phases), 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.*

[3] This definition is not the same as the one that is given in Amis et al. (2000) but both definitions are equivalent(see Th. 2.5 page 17).

- *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 Formal validation of the algorithm

It is necessary to check that all the definitions are coherent, i.e. a node chosen as a cluster head by another node is actually a cluster head (with respect to the construction of the set S), and that this node is in the d-hop neighborhood of the cluster head.
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 2.2. *Let us denote $M(x)$ the value $W_d(x)$.*

Theorem 2.2. $\forall\, x \in V \quad \forall\, y \in \mathcal{N}_d(x) \quad M(y) \geqslant v(x)$

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 $Max\,\{v(z),\ z \in\ \mathcal{N}_d(y)\} \geqslant v(x)$. □

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 2.3. *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. 2.1 it follows that it has to be proven that $W_{2d}(y) = v(y)$. The node y is among the d hop neighbors of x since $W_d(x)=v(y)$, so in the other way round, x is among the d hop neighbors 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. 2.2 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 2.3 proves that $y \in S$ and from the proof it appears that $y \in \mathcal{N}_d(x)$. □

Theorem 2.4. *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.2.3 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 2.4 proves that $y \in S$ because $v(y)$ appears in the Min phase. And since $v(y)$ appears at least once in the Max phase, then $y \in \mathcal{N}_d(x)$. So $y \in S \cap \mathcal{N}_d(x)$. □

Remark 1. *>From the first point of Th. 2.1, it seems reasonable to choose the k-dominating node corresponding to the smallest pair, when there are several ones. This choice leads to sets that are dominated by a smaller criterion value node.*

This definition of S (see Def. 2.1) is different from the definition given in Amis et al. (2000). 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 2.5. $S = S'$.

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. 2.2 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 Amis et al. (2000). 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 not 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:

$$\begin{aligned}
&x \in S \Leftrightarrow \\
&\mathcal{N}_d(x) = \varnothing \; or \\
&\qquad \exists y \in \mathcal{N}_d(x) \; v(x) = Max\,\{W_d(z),\, z \in \; \mathcal{N}_d(y)\}
\end{aligned}$$

Theorem 2.6. *Let us consider a graph (it may be finite or infinite) and d the maximal depth chosen, let us denote 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

The node degree $d(i)$ (i.e. number of neighbors) may be used as a criterion to select the cluster heads: the criterion may be the couple (node degree, node id) and a total order relation may be defined by:

$$\begin{aligned}
(d(x),x) > (d(y),y) \qquad &\Leftrightarrow \\
&(d(x) > d(y)) \quad or \quad (d(x) = d(y) \; and \; x > y)
\end{aligned}$$

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.

Simulations of the mechanism have been run for n nodes randomly and uniformly distributed distributed over a $100 \cdot 100$ surface, and a coverage radius R equal to 5.
It can be observed on figure 2 that the number of cluster heads converges towards a constant when the density of nodes increases. Actually, for a given area and a fixed transmission radius, the number of nodes a cluster can be constituted of is not bounded. Consequently, there is a density from which the number of cluster heads stops to increase when the total number of nodes increases. The figure 2 shows that to choose the degree of the nodes (curve "Node degree" on the figure) allows to obtain less cluster heads than the node identifier. The percentage of the number of cluster heads in function of the mean degree of a node is presented on figure 3.

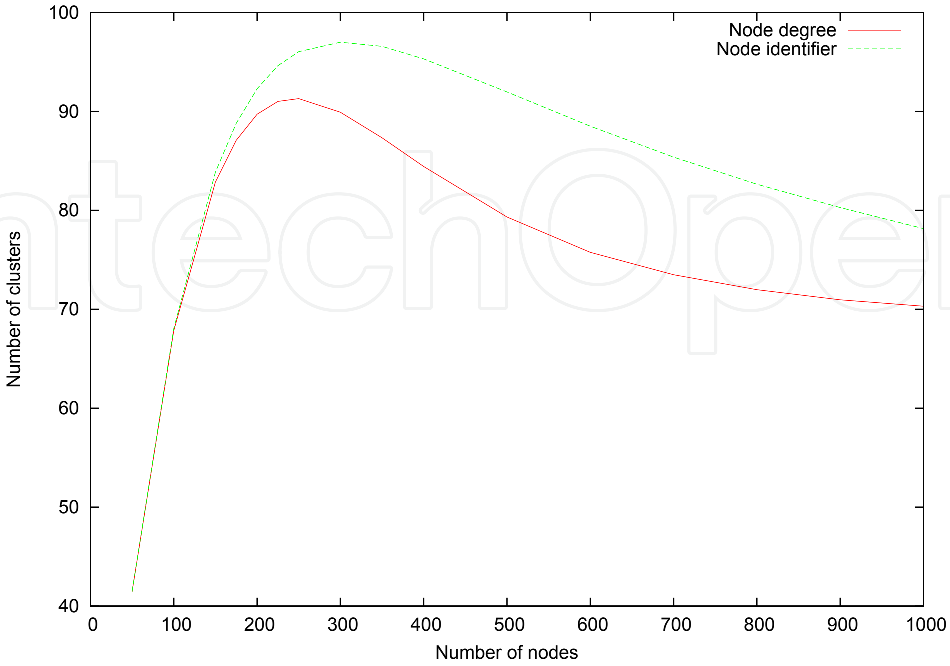


Fig. 2. Number of cluster heads for $S = 100 \cdot 100$, $R = 5$, $d = 3$

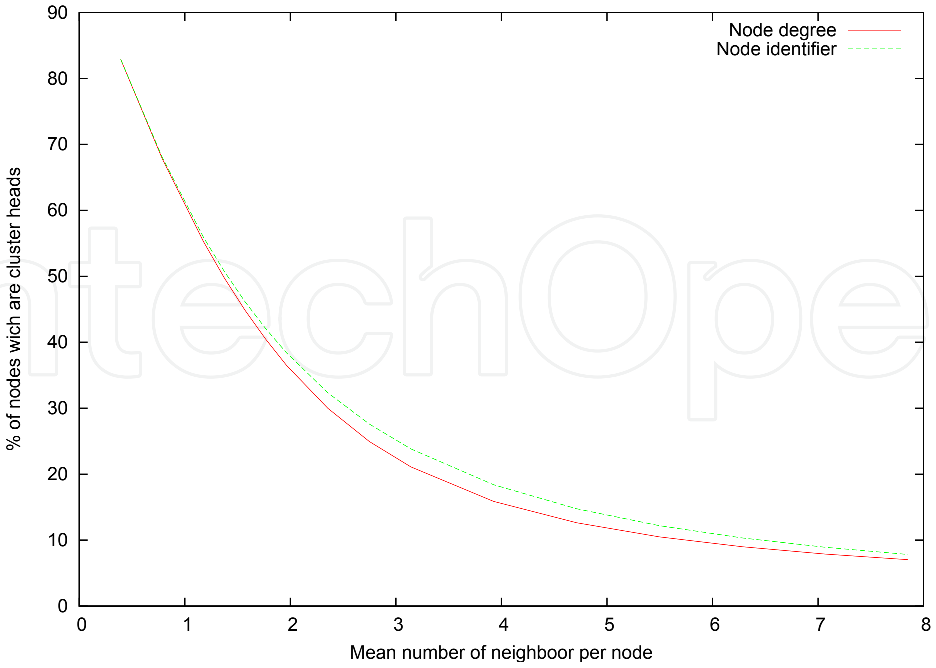


Fig. 3. Percentage of cluster heads for $S = 100 \cdot 100$, $R = 5$, $d = 3$

3. The Maxi-Min d-cluster formation: formation of the clusters

In the previous section, we proved that the nodes can determine a d-dominating set over the graph, for any 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 the formation of these clusters is studied. In paper Amis et al. (2000), the authors proposed a formation of the 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 Amis et al. (2000) 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 a father;
- *Rule 2:* Else, if x finds a pair ($v(y)$) in its *Winner* list (i.e. if $v(y)$ appears at least once in each of the two phases), then x selects y as a clusterhead [4]. 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 a father [5].
- *Rule 3:* Else, let the node y be such as $v(y){=}W_d(x)$. Then x selects y as a clusterhead [4]. x selects $S_k(x)$ as a father [5].

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 Amis et al. (2000) and it is quoted 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 neighboring 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 neighbors, it knows all the links between it and nodes in its cluster. Moreover it knows all the links between its cluster and the other neighboring 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.

We now show that cases exist where this condition is not always true because of the fact that loops may appear and we give a necessary and sufficient condition for these loops to occur. This necessary and sufficient condition is due to rule 2. We also show that to remove the loops

[4] We proved in the first part (cf. Th. 2.1 page 15.), 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.

[5] By definition, $S_k(x) \in \mathcal{N}_1(x)$, cf. page 15.

(by removing this rule 2) is not sufficient to allow the use of this cluster construction heuristic as proposed by their authors. Actually, removing the rule 2 leads to other problems: a node may have as a father a node belonging to another cluster than its own. We deduce than we can (and we must) keep the heuristic to select the cluster heads but the way the clusters are built must be set up differently from what they propose.

3.1 On an example where the algorithm leads to a bug

Let the parameter d be chosen as 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 a father and node 10 as a clusterhead. However, the node 5 has node 3 as father and node 11 as clusterhead.

	1	2	3	4	5	6	7	8	9	10	11
*Max*1	11	6	5	10	10	7	8	9	9	10	11
*Max*2	11	11	10	10	10	10	9	9	9	10	11
*Max*3	11	11	11	10	10	11	10	9	9	10	11
*Max*4	11	11	11	11	11	11	11	10	9	10	11
*Max*5	11	11	11	11	11	11	11	11	10	11	11
*Min*1	11	11	11	11	11	11	11	10	10	11	11
*Min*2	11	11	11	11	11	11	10	10	10	11	11
*Min*3	11	11	11	11	11	10	10	10	10	11	11
*Min*4	11	10	11	10	11	10	10	10	10	11	11
*Min*5	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

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

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*.

3.2 A necessary and sufficient condition for loops to appear

Note 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 let us 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 Amis et al. (2000). If i and j are two nodes, let us denote $d(i, j)$ 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 deduction proves ab absurdo that $c(i) \neq c(j)$. Assume that $c(i) = c(j) = l$, then $d(i, l) = d(j, l) + 1 \quad d(j, l) = d(i, l) + 1$ 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 to the d hop neighborhood of $c(i)$. Therefore, according to the equality true for all the nodes, $p(i)$ also is in the d hop neighborhood 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, let us continue by investigating whether removing the *Rule 2* could be appropriate and indeed it is proved as follows, that by removing the *Rule 2* there is no further 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 neighborhood 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):

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

which is absurd. The same kind of demonstration can be applied for any other loop for any given length.
Hence, if the *Rule 2* is removed, which is necessary, there is no more problem of loops.

3.3 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 loop 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.

	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

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

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 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. This type of problem thus 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.

3.4 Another proposal for the formation of the clusters

Let us start with the clusterheads: if the node i is a clusterhead, after application of the *Rule 1*, then node i informs its neighbors that it is a clusterhead. The neighbors who have not already chosen a clusterhead choose i as a clusterhead. Then, they also transmit a message to their neighbors saying that they are at one hop from the clusterhead i. The neighbors of these nodes which did not already choose a clusterhead then choose i as clusterhead by attaching themselves to one of the neighbors of i and proceed in the same way by informing their neighbors that they are two hops away 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. Because of the second part of the theorem 2.1, each non isolated node which is not cluster head is guaranteed to have a cluster head in its d-neighborhood.

4. On cluster modelling

Having a method to build clusters, it is natural to search to characterize these clusters. It is presented in this section results on cluster modelling. Bounds for the number of clusters are first given. Then, the size of the cluster is investigated in function of parameters of the network like the node density and their coverage radius. At last, the validity of the Voronoï model to model clusters is checked. Actually, in most of the papers dealing with clusters, they are modeled by the Voronoï cell centered in the cluster head. But is it valid? By the way, it is proved that the only quantity of interest when dealing with nodes distributed according to a Poisson process with intensity λ and a coverage radius R is $\lambda\pi R^2$

4.1 Analysis of the number of clusters

We searched to bound the number of cluster heads obtained with the MaxMin algorithm. Actually, to calculate exactly the average number is a very difficul problem related to the percolation theory. It can be shown that (cf. Delye (2007)):

$$\mathbf{E}\left[\text{Number of cluster heads in a surface } \mathcal{S}\right] \geq \lambda \cdot \mathcal{S} \cdot \exp\left(-\lambda\pi R^2\right)$$

In the case where the criterion is uniformly and independently distributed, and in the simplest case where the parameter d is equal to 1,

$$\mathbf{P}\left[O \text{ is cluster head}\right] \leq \left(1 + \sum_{i=1}^{\infty} \frac{1}{n} \frac{(\lambda \pi R^2)^n}{n!}\right) \exp\left(-\lambda \pi R^2\right)$$

In the case of a parameter $d > 1$, and still if the criterion is uniformly distributed, for a surface $\mathcal{S}$ and by denoting $E = \lambda \pi R^2$,

$$\mathbf{E}\left[\text{Number of cluster heads in} \mathcal{S}\right] \\ \leq \lambda \cdot \mathcal{S} \cdot \left(1 + \sum_{i=1}^{\infty} \frac{1}{n} \frac{E^n}{n!}\right) \exp\left(-E\right)$$

4.2 An empirical model of the size of the clusters

We have later searched to characterize the number of elements in a cluster. It is a very difficult problem which is to date not already solved. Actually, researchers face the problem to derive a simple law because of the strong dependence of the random variables in the considered process. This problem is related to the percolation theory. In this section, we begin by presenting the known results about coverage, connectivity and percolation. Then, we present our empirical work on the characterization of the size of clusters in a network, which gives at the same time interesting results on percolation.

4.2.1 Coverage, connectivity and percolation

Generally, it is common to consider that the sensors are spread over a plane surface according to a Poisson distribution and that they have a circular coverage, generally with the same radius. In the case of nodes distributed according to a Poisson process with a transmission radius R, we can then find the distribution of the law of the number N of neighbors of a node. It is the same distribution:

$$\mathbf{P}[N = k] = \frac{(\lambda \pi R^2)^k}{k!} \cdot e^{-\lambda \pi R^2}, \mathbf{E}[N] = \lambda \pi R^2 \tag{1}$$

Another newer model, *the Blinking Poisson Model*, was introduced in 2004 by Dousse et al. in the paper Dousse et al. (2004). The idea is to consider a distribution of sensors following a Poisson process with rate λ and a transmission radius R_i for each node. The R_i are independent of the Poisson process and their average is $\mathbf{E}[R_i]$. The sensors switch on (*on* period) and off (*off* period) independently from each other. It is assumed that *on* and *off* periods are independent. The period *on* is distributed according to any distribution with mean t_{on} The *off* period can be distributed according to a deterministic or exponential law.

This short state of the art does not pretend to be exhaustive. The reader can refer to the works of Werner Wendelin, Oded Schramm, Gregory Lawler, François Baccelli, Bartek Blaszczyszyn, Patrick Thiran, etc. for more details.

The problem of coverage

The most interesting paper dealing with this subject is Philips et al. (1989) dating from 1989. The authors study (among other things) the probability that each point of the plane be covered when the used model is a "Poisson blob-model" with parameter λ and R. To do this, they consider a finite surface A and they make tend this surface towards the infinity. The results are the following ones :

Theorem 4.1.

$$\forall \varepsilon > 0 \quad R = \sqrt{\frac{(1-\varepsilon)\ln A}{\pi\lambda}} \Longrightarrow \lim_{A\to\infty} Pr[A \textit{ is covered}] = 0$$

Theorem 4.2.

$$\forall \varepsilon > 0 \quad R = \sqrt{\frac{(1+\varepsilon)\ln A}{\pi\lambda}} \Longrightarrow \lim_{A\to\infty} Pr[A \textit{ is covered}] = 1$$

To demonstrate the first theorem, the authors build a grid of points on the surface A. These are spaced by twice the communication radius. This way, these points are covered, or not, independantly. These points are such that there is $A/4R^2$ points in a surface included in A. So, the probability that there is 0 point not covered is $(1-\lambda\pi R^2)^{A/4R^2}$ which tends towards 0 when A tends towards the infinity.

These are important theorems to dimension a sensor network. Indeed, it is necessary that the area covered by the sensors is good. The last theorem shows that for a given surface A, there must be a given number of neighbor nodes in average to ensure coverage. More precisely, knowing that the average number of neighbors under these conditions is $\lambda\pi R^2$, it is immediate to see it must be a little more than $\ln(A)$ neighbors to be almost sure that the surface is covered.

The problem of connectivity
In the same paper Philips et al. (1989), the authors show also a theorem about the connectivity. Connectivity and coverage must not be confused. The following theorem is proved:

Theorem 4.3.

$$\forall \varepsilon > 0 \quad R = \sqrt{\frac{(1-\varepsilon)\ln A}{\pi\lambda}}$$
$$\Longrightarrow \lim_{A\to\infty} Pr[\textit{the network is connected}] = 0$$

This proves that if the average number of neighbors is given, then it is sure for a large enough surface to have a network disconnected. The authors did not succeed to demonstrate that when the number of neighbors were on average a little larger than $\ln(A)$ connectivity was ensured. The consequence of this theorem is that it can not exist magic number of neighbors. In particular, 6 is not a magic number for the network rate, contrary to what Kleinrock and Sylvester claimed in Kleinrock & Silvester (1978) in 1978.

Using "slotted ALOHA" protocols and by requiring that the transmission power of the nodes be the same for all, Kleinrock and Sylvester Kleinrock & Silvester (1978) suggested that the number six is considered a magic number. Later in 1984, the magic number changed and 8 became the newly elected one Takagi & Kleinrock (1984). In this same paper, Tagaki and Kleinrock also found two other magic numbers (5 and 7) considering other transmission protocols. Considering that the nodes can adapt their transmission radius, the authors of Hou & Li (1986) proposed in 1986 the magic numbers 6 and 8. This is the pantheon of a belief which became false in 1989!

Indeed, none of these analyses dealt with the problem of network connectivity. When Tantawi et al. looked à this problem in 1989 Philips et al. (1989), they proved that no number can be magic. The authors showed that whatever the average number of neighbors is chosen, the network will almost surely be disconnected if this number is constant ...

The authors of Gupta & Kumar (1998) deal with the problem of connectivity on a finite circular surface with unit area in which N nodes are randomly placed. The node density is then $\lambda = N$. The transmission radius of a node n is r. The authors show that if $\pi r^2 = (log(N) + c(N))/N$ then the network is asymptotically connected (ie. when N tends to the infinity) with a probability 1 if and only if $c(N)$ tends also to the infinity. This leads the authors of Shakkottai et al. (2003) to say that the transmission radius must be of the order of $\sqrt{log(N)/N}$ for the network to be connected.

The authors of Shakkottai et al. (2003) study a network of sensors placed on a unit area square surface. When n nodes constitute the network (n is supposed to be a squared number), they are placed on a grid such as the distance between two nodes able to communicate is $\sqrt{1/n}$. When the transmission radius is of the order of $\sqrt{1/n}$, the connectivity is ensured. These authors continue their study by supposing that the nodes are on with probability $p(n)$. They show then that the connectivity is asymptotically ensured when $\sqrt{p(n)}r(n) \sim \sqrt{log(n)/n}$. Moreover, they show that the diameter of such a network where the nodes can crash is of the order of $\sqrt{n/log(n)}$.

To choose the good number of neighbors is important. Indeed, this choice impacts not only the network connectivity but also its capacity: the presence of a large number of links between the nodes is not necessarily advantageous. Indeed, if a link exists between i and j, it is an advantage in terms of energy consumption since i can send a packet to j in a single hop. However, when i sends a message to another neighbor, it causes interferences at j which would not have existed without the link that connects them. Therefore there is a trade-off. More specifically, when the transmission radius increases, the number of retransmissions decreases, but the value of the interferences increases. In the paper Gupta et al. (2000), P. Gupta and P. R. Kumar showed that the number of retransmissions increased as $O(1/r)$ (when r increases) but that the interferences were "only" on the order of $O(r^2)$. Thus, the product of both quantities leads to assert that the "net" effect is about $O(r)$. This means that it is better to choose a radius of little value. However, if the radius is too low, then of course the network is disconnected!

For the moment, the best results are those of Xue Feng and P. R. Kumar presented in 2004 in the paper Xue & Kumar (2002). In a network with n nodes randomly placed (uniformely), the number of neighbors of each node must be of the order of $\Theta(log(n))$ so that the network is connected. More precisely, the network is asymptotically disconnected when this number is less than $0,075 \cdot \log n$ and is asymptotically connected when this number is greater than $5.1774 \cdot \log n$ neighbors.

The problem of percolation

It is dealt here with an issue that seems very simple and raised in 1963 by E. N. Gilbert, the issue of critical density in percolation in a network of clusters. Gilbert is one of the first to propose a modelling of wireless networks. His model is a particular case of modelling with Boolean networks. He deals with nodes placed in the (infinite) plan according to a two dimensions Poisson process. He demonstrates that in such a plan, there is a critical threshold beyond which the probability to belong to an infinite size cluster is not zero. It is said in this case that there is percolation.

Gilbert introduced in 1961 Gilbert (1961) a modelling of these networks using a graph formalism. The vertices of this gaph are the nodes. All the nodes are supposed to be in a same plan (dimension 2). He assumes that two nodes can communicate if and only if their distance is less than a given value R. An edge exists in the graph of the vertices if and only if the respective nodes of each vertice can directly communicate. Gilbert builds such a network with a Poisson process with intensity λ, on an infinite plan. Each connected component is called cluster.

Let be the quantity $E = \lambda\pi R^2$, expectation of the number of points in a circle of radius R and $P(N)$ the probability that a node belongs to a cluster of size larger than $N - 1$. Gilbert shows the following theorem :

Theorem 4.4.

$$\exists E_c \in \mathbb{R} \quad \forall E \in \mathbb{R} \quad \begin{cases} P(\infty) = 0 & \textit{if } E < E_c \\ P(\infty) > 0 & \textit{if } E > E_c \end{cases}$$

He bounds also E_c :

$$1.64 \approx \frac{1}{\frac{1}{3} + \frac{\sqrt{3}}{2\pi}} \leq E_c \leq 8\pi \log_e(2) \approx 17.4$$

In fact, Gilbert wrote 1.75 instead of 1.64, but it is a typo in the paper. He shows also by simulation that $E_c \approx 3.2$. He suggests a beginning for a demonstration which would help to prove that $E_c \leq \dfrac{26\pi}{3\sqrt{3}} \log_e 2 \approx 10.9$ but he does not succeed to conclude.

Kirkook and Wayne Kirkwood & Wayne (1983) and Hall Hall (1985) showed that $2.186 < E_c < 10.588$.

In 1989, Tantawi et al. Philips et al. (1989) proved that the critical value E_c is in the interval : $2.195 < E_c < 10.526$. The demonstration, non explicitly given in the paper, uses an analogy with the $M/D/1$ queue. The instability of this system corresponds to the existence of an infinite component.

These results are summarized in Fig. 4. The probability to obtain an infinite size cluster is zero for R and λ under the point of the curve at $E_c = 2.195$, and it is non zero for all points above $E_c = 10.588$ To the best of our knowledge, there exists no better bounds than the ones given by Tantawi et al. This very simple problem, dating from 1963, is not already solved.

The authors of Dousse et al. (2004) give the results about percolation for the previously described model. They show that there exists a critical density λ_c, function of R, λ and π_{on} such as the network is constituted almost surely of a unique infinite component for $\lambda > \lambda_c$ and almost surely of an infinity of finite components for $\lambda < \lambda_c$. In addition, if λ^* denotes Gilbert's critical density for R given, then these authors show that $\lambda^* = \pi_{on}\lambda_c$. Consequently, it means that if the sensors are placed according to a two dimension Poisson process with intensity λ and if they switch on and off each independently of one another according to

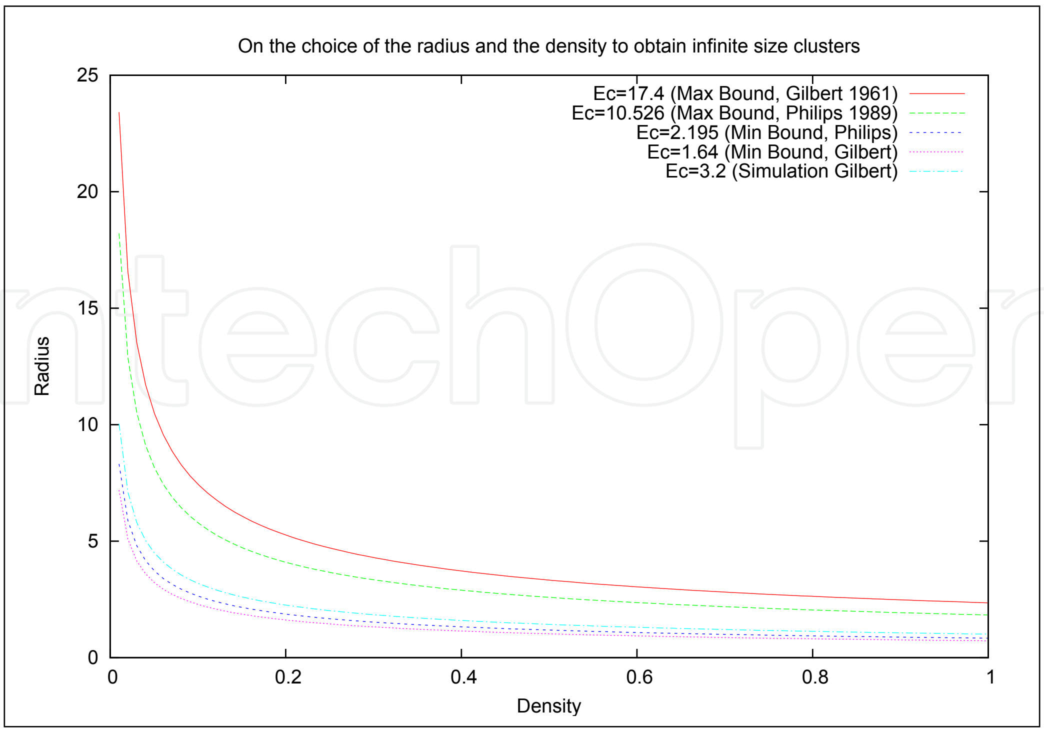


Fig. 4. Synthesis of the results

on and off periods with a ratio π_{on}, then a communication is possible between two sensors almost surely if the density of the Poisson process is strictly greater than Gilbert's constant: $\lambda > \lambda_c = \frac{\lambda^*}{\pi_{on}}$. From the viewpoint of percolation, it is as if there had been a Poisson process with density $\lambda\pi_{on}$.

Moreover, the authors study the transmission delay between two nodes X and Y belonging to the infinite component, where $\lambda > \lambda_c$. Indeed, this delay is theoretically zero when $\pi_{on} = 1$ (classical model) but this is no more true with the Blinking Poisson Model since sensors switch on independently from each other. The result they demonstrate is the following:

$$\exists \eta > 0 \quad (1-\varepsilon)\eta < \frac{T(X,Y)}{|X-Y|} < (1+\varepsilon)\eta$$

where $T(X,Y)$ is the time necessary to transmit and $|X-Y|$ their euclidian distance. The result is true for $|X-Y|$ sufficiently high. This result shows that under this transmission model (which does not take into account, of course, the interferences), the time needed to deliver a message increases linearly with the Euclidean distance. The value of η depends only on the parameters of the model and may therefore be determined by simulation.

4.2.2 Evaluation by simulation of the cluster size: an empirical result

Here, we present an empirical work on the the size of the clusters in terms of number of nodes per cluster. This, by the way, gives a result on percolation in sensor networks: the size of the cluster diverges above a certain density of nodes. placed on the plan through a Poisson process with rate λ and with a transmission radius R. The number of nodes N is a function of λ and R, but, more precisely, a function only of the quantity $E = \lambda\pi R^2$ proposed by Gilbert.

This can be verified on Fig. 6. This is exactly to say that $N = f(\lambda, R) = g(\lambda\pi R^2)$

Note that, the cluster size seems not to diverge at $E_c = 2.3$ as thought Gilbert but rather at $E_c = 4.4$. This is most certainly due to the board effects on small surfaces chosen by him. However, we verified that the divergence is well within the range proposed by Gilbert.

The inverse of the natural logarithm of the number of nodes in function of E can be approximated by a straight line. We determined its coefficients empirically. This allows therefore approaching the number of nodes by the following formula:

$$N = \exp(1/(-0.155E + 0.787))$$

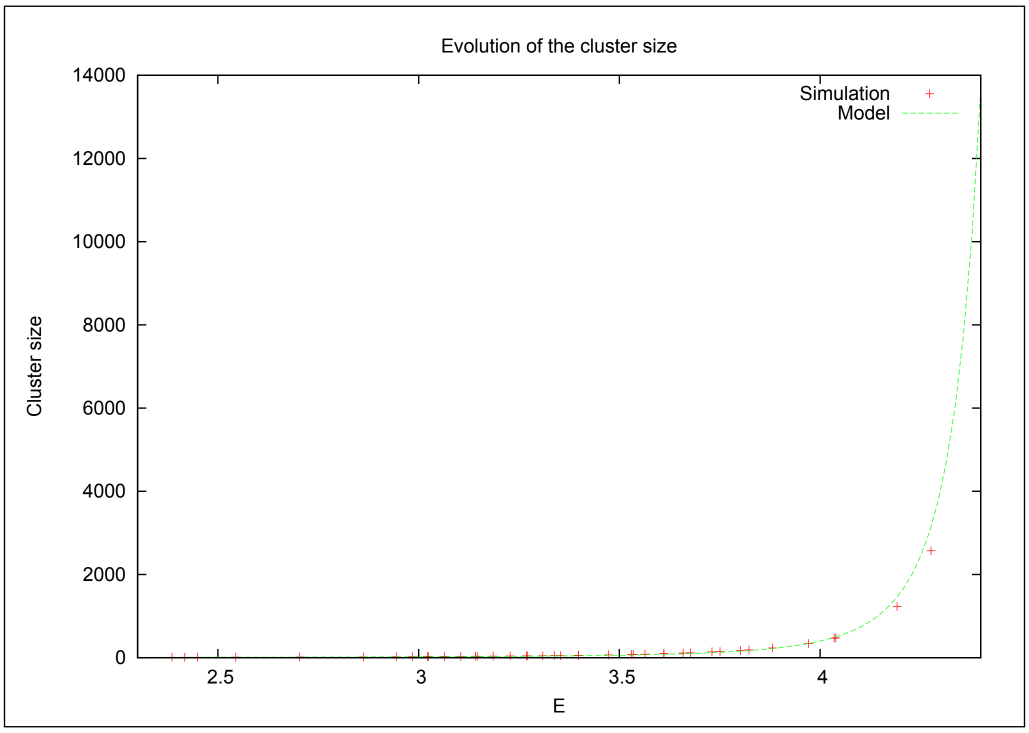


Fig. 5. Cluster size in function of $\lambda\pi R^2$

Comparisons of simulation and heuristic in two dimensions and in three dimensions can be found on the figures 5 and 6.

4.3 Voronoï's modelling

The Voronoï's theory Voronoï (1907) can be used to obtain analytical results in ad-hoc and sensor networks. Actually, the known results of this theory are often applied to model these networks: the comparison between the analytical results given by a Voronoï modelling of the network and the ones obtained by simulation may be interesting. For example, the authors of Bandyopadhyay & Coyle (2003) give an analysis of the performance of their cluster formation algorithm. All their work assumes that the clusters formed with their algorithm can be modellized by Voronoï cells.

It is nevertheless important to check if it is acceptable to consider that a cluster is modelled by a Voronoï cell of which the seed is the cluster head. Actually, a Voronoï tessellation is the

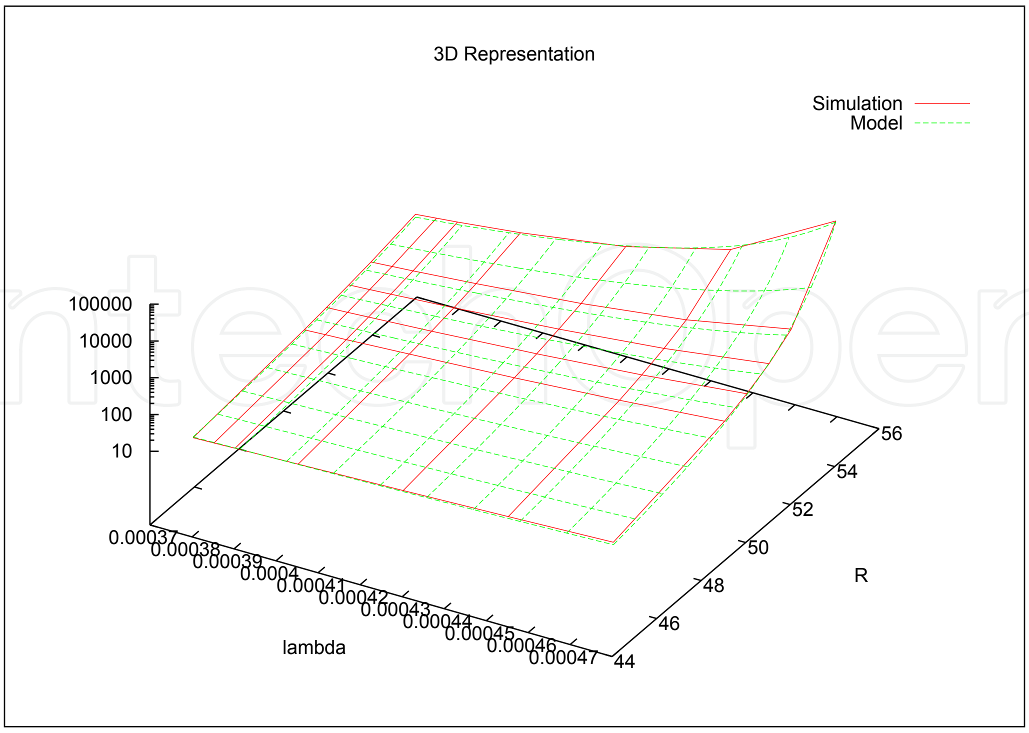


Fig. 6. 3D Comparison

partition in convex polygons generated by seeds: a polygon contains exactly one seed and all the points inside the polygon are closer to this seed than any other seed. Consequently, the simple hop cluster modelling with a Voronoï model is correct for a certain cluster formation policy. This is not at all obvious for all the multi-hop clusters of which nodes can be attached to a cluster head but belong to the Voronoï cell of another cluster head.

This highlights the importance of the choice of the cluster head for the cluster formation mechanism. It seems actually that a Voronoï modelling is more or less false depending on the spatial distribution of the cluster. First of all we assumed that the distribution of the cluster heads is uniform. It is for example the case in the paper Bandyopadhyay & Coyle (2003). Without a generalization of our work, the presented results are thus a priori restricted to this context.

Moreover, it is obvious that the cluster head choice mechanism once the clusters built is also of the highest importance. Let us consider for example cluster heads distributed according to a Poisson process with a density $p \cdot \lambda$ on a surface and a cluster formation policy such that the clusters are single-hop and have a bounded number of children k. It is obvious that some nodes cannot be attached to the nearest cluster head because it has already reached its maximal number of children. Then they can join another cluster head but they do not belong to its Voronoï cell. It is clear that for this policy, the higher the value of λ and the smaller p is, the falser the Voronoï modelling is.

We must then choose the cluster formation policy to partially answer the question. The policy we use is the policy we call canonical: a node which is a neighbor of a cluster head joins the nearest cluster head (the probability that two cluster heads are at the same distance r is $o(\lambda r)$).

This node is then said clustered. A node which has no cluster head among its neighbors joins its nearest clustered neighbor. This policy is debatable but we think it is both more favorable to the use of the Voronoï modelling by choosing the nearest cluster head and more realistic by choosing the nearest clustered neighbor. We also could choose a policy for which the clustered neighbor having the nearest cluster head would have been chosen but in practice it is not possible without the use of a triangulation. Note also that it is not easy to estimate the distance to a neighbor (and thus the nearest one) from the reception power because the wireless medium is by nature quite instable.

The Voronoi model is good for a node x if x belongs to its Voronoï cluster. In this case, it is set $V(x) = 1$. The accuracy of the modelling of the clusters by Voronoï cells can be assessed by observing the percentage of nodes for which the Voronoï model is good. This criterion is *a priori* a function of R, λ and p where R is the transmission radius, λ is the density of the network and p the ratio between the number of cluster heads and the number of nodes, $p\lambda\pi R^2$ being the density of cluster heads. The criterion is denoted by $C = f(\lambda, R, p)$.

We have first evaluated the proportion of ordinary nodes belonging to a cluster. Actually, some nodes can belong to a strongly connected component while there is no cluster head in this component. We have evaluated the percentage of nodes for which the Voronoï modelling is good, that is for which the Voronoï cell in which they are is well centered on their cluster head. These results allow to evaluate in which conditions the Voronoï modelling is acceptable. We have found that it is true for a large interval of densities. A node is said to belong to its Voronoï cluster if and only if its cluster head is the seed of the Voronoï cell which it belongs to.

4.3.1 Probability for a node to belong to a cluster: a simulation study

The cluster formation is done according to the canonical policy above described: first a single hop cluster is formed with the cluster heads and their neighbors. The neighbors of the cluster head can then build a two hop cluster by associating their neighbors, and so on. Two types of nodes exist once the clusters constructed: those actually attached (possibly indirectly via several hops) to a cluster (clustered nodes) and those who are not clustered.

We simulated a Poisson process on a surface area $A = 10000 \cdot 10000$. The transmission radius R and the density λ of the Poisson process are the two main parameters. The percentage of cluster heads is another fundamental parameter, as it will be explained later and is denoted by p. A node is a cluster head with probability p. Then, we know that the cluster heads are distributed according to a Poisson process with intensity $\lambda \cdot p$ while the other nodes are distributed according to a Poisson process with intensity $\lambda \cdot (1 - p)$. In addition, another result states that both processes are independent from the Poisson process with intensity λ. The cluster heads constitute a set denoted S_0 while the set of the other nodes is denoted S'_0.

For the parameters λ,R and p, the probability for a node belonging to S'_0 to be clustered is denoted $\mathbf{P}_{\lambda,R,p}[C(x) = 1]$. This probability has been simulated for

$$\lambda = \{0.001\ 0.0012\ 0.0014\ 0.0016\ 0.0018\ 0.002\},$$
$$R = \{5\ 15\ 25\ 35\ 45\ 55\ 65\ 75\ 85\ 95\}$$

and

$$p = \{0.05\ 0.1\ 0.15\ 0.2\ 0.25\ 0.3\ 0.35\ 0.4\ 0.45\ 0.5\}.$$

As expected, (cf. Sec.4.3.1), $\mathbf{P}_{\lambda,R,p}[C(x) = 1]$ is only a function of $E = \lambda\pi R^2$ and p, probability for a node to be cluster head. The probability $\mathbf{P}_{\lambda,R,p}[C(x) = 1]$ is thus denoted $\mathbf{P}_{\lambda\pi R^2,p}[C(x) = 1]$. The simulations of this probability are given on figure 7. It can be observed that, wathever the value of the parameter p, $\mathbf{P}_{\lambda\pi R^2,p}[C(x) = 1]$ is an increasing function of $\lambda\pi R^2$, which was expected. It can be noticed that, for the "magic number" 6 and $p \geq 0.05$, the probability that a node is clustered is greater than 95%.

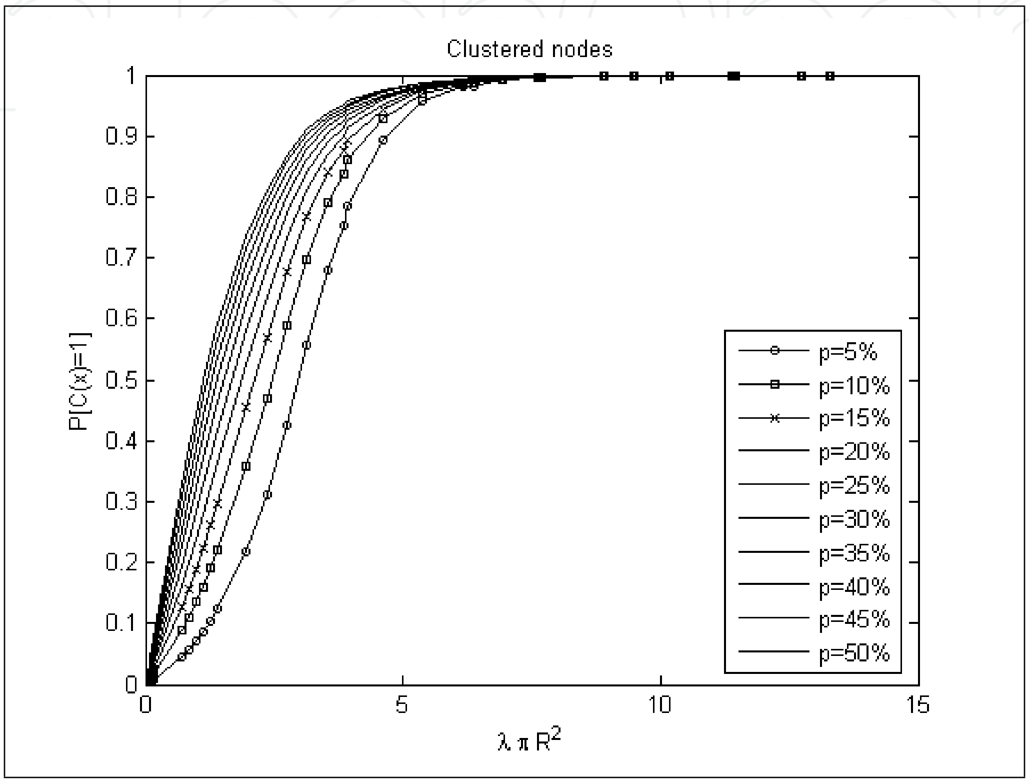


Fig. 7. Probability for a node belonging to S'_0 to be clustered

Figure 7 gives then limits for the use of the modelling of these clusters by Voronoï cells. Actually, a node must at least be clustered with a good probability to be used in a Voronoï model. The Voronoï modelling must thus not be used for a wireless network distributed according to a Poisson process with intensity λ and a communication radius R such as $\lambda\pi R^2 \leq 5$ which is a lower bound for $\mathbf{P}_{\lambda,R,p}[C(x)=1] \geq 85\%$ when $p \geq 5\%$.

4.3.2 Probability for a node to belong to a cluster: analytical results

In this section, the probability that a node, which is not a cluster head, is clustered is expressed and approximated.

The probabilities ψ and Ψ

We want to evaluate the probability $\psi(X)$ that a node X is clustered knowing that it is not a cluster head. Let $\Psi(S)$ be the probability that a surface S contains at least a clustered node. By observing that a node X is clustered if and only if the open subset of center X and radius R (denoted $B(X,R)$) contains at least a clustered node, the following equality is true: $\psi(X) = \Psi(B(X,R))$.

$\Psi(S) = \psi(X)\nu(S)$ cannot be written since $\psi(X) = \Psi(B(X,R))$. Moreover, a node is clustered if and only if there exists a path from this node to its cluster head. These cluster heads constitute a two dimension Poisson process with density $p\lambda$, that is the number of cluster heads in a surface R follows a Poisson law with parameter $p \cdot \lambda \cdot \nu(S)$.

Another way to calculate the probability that a node is clustered is to consider it is clustered if and only if at least one of its one hop neighbors is a cluster head or at least one of its two hop neighbors is a cluster head, and so on. We will search a lower bound of this probability by calculating the probability that at least one of its neighbors at less than two hops is a cluster head.

Calculation of the supplementary surface $S(n,R)$ brought by n nodes in a disk with radius R to the ring $A(X,R,2R)$

Let be a disk with centre X and radius R. Let's assume that n nodes are uniformly distributed inside. These nodes bring a supplementary surface $S(n)$ to $B(X,R)$ which is a portion of the ring $A(X,R,2R)$ (surface of the disk of radius $2R$ centered in X minus the disk of radius R centered in X). What is the supplementary surface $S(n)$? When $n = 0$, $S(0)$ should be 0. When n tends towards the infinity, all infinitesimal element of $B(X,R)$ contains exactly one node. In this case, $\lim_{n\to\infty} S(n) = 3\pi R^2$, area of the ring. It can be shown that (cf. Delye (2007)):

$$S(n,R) = 3\pi R^2 - 2\pi \int_{r=R}^{2R} \left(1 - \frac{I(r,R)}{\pi R^2}\right)^n r dr \tag{2}$$

with

$$I(r,R) = 2 \cdot R^2 \cdot \arccos\left(\frac{r}{2R}\right) - r \cdot \sqrt{R^2 - \frac{r^2}{4}} \tag{3}$$

When thinking about the quantity $P_n(R) = S(n,R)/(3\pi R^2)$ from a probabilistic viewpoint, it can be conjectured that this quantity does not depend on R. Indeed, $P_n(R)$ represents the percentage of the surface of the ring $A(R,2R)$ brought by the n nodes uniformly placed on the disk $B(0,R)$. Actually, it can be shown (cf. Delye (2007)):

$$P_n(R) \quad = \quad 1 - \frac{2}{3} \int_{u=1}^{2} \left(1 + \frac{u}{\pi} \cdot \sqrt{1 - \frac{u^2}{4}} - \frac{2}{\pi} \cdot \arccos\left(\frac{u}{2}\right)\right)^n udu$$

This last equation thus shows that $P_n(R)$ is independent of R. It can also be shown (cf. Delye (2007)) that the probability that a node has n_2 two hop neighbors is:

$$\mathbf{P}[n_2 = k] \quad = \quad \sum_{i=0}^{\infty} \frac{(3P_iE)^k}{k!} e^{-3P_iE} \frac{E^i}{i!} e^{-E}$$

Its expecation is then:

$$\mathbf{E}[n_2] \quad = \quad \sum_{i=0}^{\infty} P_i \frac{E^i}{i!} e^{-E} 3E$$

Calculation of the probability to be clustered for two-hop clusters

Let be Φ a two hop Poisson process with intensity λ. Let be R the transmission radius and p the probability that a node is cluster head. Let be $x = E = \lambda\pi R^2$ the mean number of neighbors and $\psi(x)$ the probability for a node which is not a cluster head, to be clustered. This probability can be seen as the sum of two other ones :

- the probability to be clustered and that there is at least a cluster head among the neighbors : it is probability $p1$;
- the probability to be clustered and that there is no cluster head among the neighbors : it is probability $p2$.

For the one hop case, the process of the cluster heads is a two dimension Poisson process Φ_C with density $p\lambda$ and which is independent of the first process. Then :

$$p1 = \mathbf{P}^{\circ}\left[\Phi_C(B'(X,R)) = 0\right] = 1 - e^{-p\lambda\pi R^2} = 1 - e^{-px}$$

And for the two hop case, it can be shown (cf. Delye (2007)):

$$p2 = e^{-x}\left[1 - e^{-(1-p)x} - \sum_{i=1}^{\infty}\frac{((1-p)x)^i}{i!}e^{-p\lambda S(i)} + \epsilon\right]$$

with

$$\epsilon = \sum_{i,j=1}^{\infty}\frac{((1-p)x)^i}{i!}\frac{(\lambda S(i)(1-p))^j}{j!}e^{-\lambda S(i)}\psi(i,j)$$

We did not succeed to calculate exactly $\psi(i,j)$. We can only give a lower bound of the probability of clustering of a node :

$$\begin{aligned}&\mathbf{P}\left[C(X) = 1\right] \geq \\ &\quad 1 - e^{-px} + e^{-x}\left[1 - e^{-(1-p)x} - \sum_{i=1}^{\infty}\frac{((1-p)x)^i}{i!}e^{-p\lambda S(i)}\right]\end{aligned}$$

4.3.3 Evaluation of the Voronoï model

Here the probability that a node belongs to its Voronoï clusteri is presented. We simulated cluster head distributions and the "canonical" policy for each one of the following triplets :

$$\begin{aligned}\lambda =& \{0.001\ 0.0012\ 0.0014\ 0.0016\ 0.0018\ 0.002\},\\ R =& \{5\ 15\ 25\ 35\ 45\ 55\ 65\ 75\ 85\ 95\}\end{aligned}$$

and

$$p = \{0.05\ 0.1\ 0.15\ 0.2\ 0.25\ 0.3\ 0.35\ 0.4\ 0.45\ 0.5\}.$$

These values lead to $6 \cdot 10 \cdot 10 = 600$ parameters. The average of the criterion is obtained on 1000 simulations for each parameter.

Figure 8 shows the probability that a node belonging to S'_0 is in its Voronoï cluster in function of E and p where $E = \lambda\pi R^2$ and p. E is the average of the number of neighbors per node. The

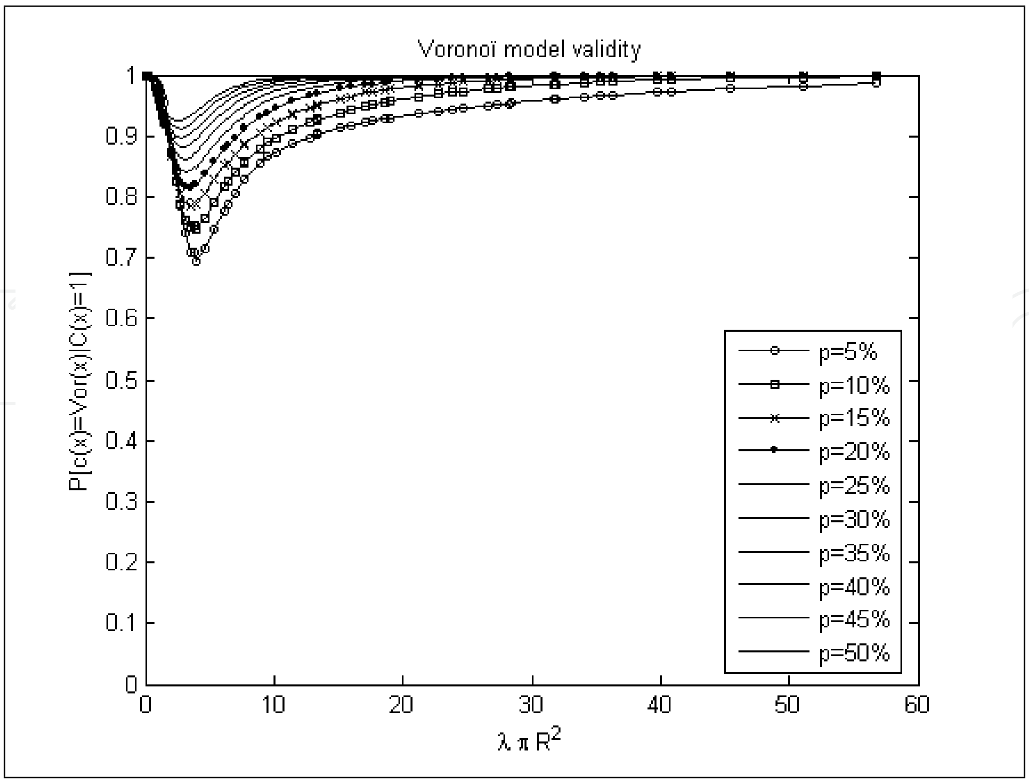


Fig. 8. Probability for a node S_0' to belong to its Voronoï cluster

validity of the Voronoï model is only function of E and p. $C = f(E, p)$. $V(E, p)$ is equal to 1 when E is small since there is only a single node per cluster, the cluster head itself. When E is large, $V(E, p) = 1$ since every node is connected to its Voronoï cluster because the density is very large. Since the density of the cluster heads is larger than 5% the probability that a node belongs to its Voronoï cluster is larger than 72%.

5. An address assignment mechanism

Addressing nodes is an important step which itself consumes energy and we searched an addressing mechanism allowing to economize energy compared to "naive" protocols like the Cluster Tree Protocol proposed by the Zigbee Alliance.

In Weniger & Zitterbart (2004), the authors define a classification of the different addressing mechanisms. This classification is used in all the papers dealing with this subject. They are separated into two families: the "statefull" protocols and the "stateless" ones. The protocols of the second family do not use allocation tables like the protocols of the first family but they use random addresses or addresses based on a serial number. The protocols of the first type are classified into two subsets: the ones using a centralized allocation table (*Centralized Autoconfiguration CAC*) and the protocols using a distributed allocation table. The protocols MANETconf, Boleng's and Prophet Allocation Zhou et al. (2003) belongs to this last category. At last, a third hybrid family is proposed, in which are the protocols HCQA Yuan Sun et al. (2003) and PACMAN.

We have proposed an address assignment algorithm which is based neither on probabilistic considerations or serial numbers, nor on an address table storage (distributed or not). Moreover, this protocol minimizes the number of exchanges allowing to obtain an address: when a node wants to obtain an address from another one, a single exchange is necessary between these two nodes. An economy of emission, reception and storage is thus gained. This work has been presented in Delye de Clauzade de Mazieux et al. (2009). At the same time, the ZigBee Alliance retained an algorithm very close to this one. This algorithm is based only on a single constaint: the a priori knowledge of the maximum number of children of the vertices in the graph. The idea is the following.

We consider a tree structure (i.e. the cluster has physically a tree structure). We have designed a distributed addressing algorithm on this tree. For a node i of this tree, let $@_i$ denote its address and $e_i(t)$ the number of its children at time t. Let d be a fixed integer. Assume the highest degree of the root is $d-1$ and that the other nodes have degrees less or equal to d. This means that all the nodes have at most $d-1$ children. The root is assigned the address 0. Assume that following an event, node j, without address, queries node i at time t in order to obtain an address. Since $e_i(t) < d-1$, the node i increments the number of its children ($e_i(t+dt) = e_i(t)+1$) and the node i attributes to the node j the address $@_j = d * @_i + e_i(t+dt)$.

This addressing mechanism has interesting properties. First, in terms of efficiency, as already noticed, it is more efficient than the Cluster Tree Protocol. Second, it allows to set up self-routing. Actually, from the only knowledge of the destination address and its own address, every node can determine to which next hop to send the packet to be routed. It is very similar to what allowed once the Banyan networks. The interested reader can refer to Delye de Clauzade de Mazieux et al. (2009) for more details.

6. References

Abbasi, A. A. & Younis, M. (2007). A survey on clustering algorithms for wireless sensor networks, *Comput. Commun.* **30**(14-15): 2826–2841.

Amis, A. D., Prakash, R., Thai, H. P., Dung, V. & Huynh, T. (2000). Max-min d-cluster formation in wireless ad hoc networks, *Proceedings of IEEE INFOCOM 2000*, Dan Panorama Hotel, Tel-Aviv, Israel, pp. 23–41.

Bandyopadhyay, S. & Coyle, E. (2003). An energy efficient hierarchical clustering algorithm for wireless sensor networks, *Proceedings of the Twenty-Second Annual Joint IEEE Computer and Communications Societies. IEEE INFOCOM*, pp. 1713–1723.

Banerjee, S. & Khuller, S. (2001). A clustering scheme for hierarchical control in multi-hop wireless networks, *Proceedings of IEEE Infocom*, Anchorage, Alaska.

Basagni, S. (1999a). Distributed and mobility-adaptive clustering for multimedia support in multi-hop wireless networks, *Proceedings of the IEEE 50th International Vehicular Technology Conference, VTC 1999-Fal*, Amsterdam, The Netherlands, pp. 889–893, vol. 2.

Basagni, S. (1999b). Distributed clustering for ad hoc networks, *Proceedings of the 1999 International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN'99)*, Ed. A.Y. Zomaya, D.F. Hsu, O. Ibarra, S. Origuchi, D. Nassimi, M. Palis, IEEE Computer Society, Perth/Fremantle, Australia, pp. 310–315.

Basagni, S., Mastrogiovanni, M. & Petrioli, C. (2004). A performance comparison of protocols for clustering and backbone formation in large scale ad hoc network, *Proceedings of*

The 1st IEEE International Conference on Mobile Ad Hoc and Sensor Systems, MASS 2004, Fort Lauderdale, FL.

Bettstetter, C. (2004). The cluster density of a distributed clustering algorithm in ad hoc networks, *Proceedings of IEEE Intern. Conf. on Communications (ICC)*, Paris, France.

Brust, M., Frey, H. & Rothkugel, S. (2008). Dynamic multi-hop clustering for mobile hybrid wireless networks, *Proceedings of the 2nd international Conference on Ubiquitous information Management and Communication*, ACM, New York, NY.

Chan, H. & Perrig, A. (2004). ACE: An emergent algorithm for highly uniform cluster formation, *Proceedings of the European Workshop on Wireless Sensor Networks (EWSN 2004)*.

Chang, R. & Kuo, C. (2006). An energy efficient routing mechanism for wireless sensor networks, *Proceedings of the 20th international Conference on Advanced information Networking and Applications - Volume 02*.

Chatterjee, M., Das, S. K. & Turgut, D. (2001). Wca: A weighted clustering algorithm for mobile ad hoc networks, *Cluster Computing (Special Issue on Mobile Ad hoc Networks)* **5**: 193–204.

Chen, H., Wu, C.-S., Chu, Y.-S., Cheng, C.-C. & Tsai, L.-K. (2007). Energy residue aware (era) clustering algorithm for leach-based wireless sensor networks, *Proceedings of Second International Conference on Systems and Networks Communications*, pp. 40–40.

Chinara, S. & Rath, S. (2008). Mobility based clustering algorithm and the energy consumption model of dynamic nodes in mobile ad hoc network, *Proceedings of International Conference on Information Technology*, pp. 171–176.

Dai, F. & Wu, J. (2005). On constructing k-connected k-dominating set in wireless networks, *Proceedings of the 19th IEEE international Parallel and Distributed Processing Symposium (Ipdps'05)*, IEEE Computer Society, Washington, DC.

Delye, A. (2007). Etude théorique des clusters multi-sauts dans les réseaux de capteurs sans fil, *Thèse de Doctorat de L'Université Pierre et Marie Curie (Paris VI)*.

Delye de Clauzade de Mazieux, A., Marot, M. & Becker, M. (2006). Construction d'un ensemble d-dominant sur un graphe en utilisant un critère donné, *C.R. Mécanique, Académie des Sciences, Elsevier (10.1016/j.crme.2006.09.001)* **334**.

Delye de Clauzade de Mazieux, A., Marot, M. & Becker, M. (2009). Caac mechanism a cluster address auto-configuration mechanism, *WIRE Special Journal Issues on 'Wireless Networks: Performance Modelling and Evaluation' & 'Wireless Networks: Mobility Management and QoS'. To Appear*.

Demirbas, M., Arora, A. & Mittal, V. (2004). Floc: A fast local clustering service for wireless sensor networks, *Proceedings of Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks (DIWANS/DSN)*.

Deosarkar, B., Yadav, N. & Yadav, R. (2008). Clusterhead selection in clustering algorithms for wireless sensor networks: A survey, *Proceedings of the International Conference on Computing, Communication and Networking, 2008. ICCCn 2008.*, pp. 1 –8.

Depedri, A., Zanella, A. & Verdone, R. (2003). An energy efficient protocol for wireless sensor networks, *Proceedings of the Conference on Autonomous Intelligent Networks and Systems (AINS 2003)*, Menlo Park, CA.

Ding, P., Holliday, J. & Celik, A. (2005). Distributed energy efficient hierarchical clustering for wireless sensor networks, *Proceedings of the IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS'05)*, Marina Del Rey, CA.

Dousse, O., Mannersalo, P. & Thiran, P. (2004). Latency of wireless sensor networks with uncoordinated power saving mechanisms, *Proceedings of Mobihoc, 2004*, pp. 109–120.

Fan, Z. & Zhou, H. (2006). A distributed weight-based clustering algorithm for wsns, *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing, 2006. WiCOM 2006.*, pp. 1 –5.

Fang, S., Berber, S. & Swain, A. (2008). Performance of a clustering algorithm for high density wireless sensor networks, *Proceedings of the 2008 IEEE Region 10 Conference. TENCON 2008*, pp. 1 –6.

Fernandess, Y. & Communication, D. (2002). K-clustering in wireless ad hoc networks, *Proceedings of the second ACM international workshop on Principles of mobile computing, POMC âĂŹ02*, ACM Press, pp. 31–37.

Gerla, M., Kwon, T. J. & Pei, G. (2000). On-demand routing in large ad hoc wireless networks with passive clustering, *Proceedings of the 2000 IEEE Wireless Communications and Networking Conference, 2000. WCNC.*, Vol. 1, pp. 100 –105 vol.1.

Gilbert, E. N. (1961). Random plane networks, *SIAM J.* **9**: 533–543.

Guo, S.-j., Zheng, J., Qu, Y.-G., Zhao, B.-H. & Pan, Q.-K. (2007). Clustering and multi-hop routing with power control in wireless sensor networks, *The Journal of China Universities of Posts and Telecommunications* **14**(1): 49 – 57.
URL: *http://www.sciencedirect.com/science/article/B8H14-4NHNFT2-B/2/fe068a0f1badd13663dd3677da5a8238*

Gupta, G. & Younis, M. (2003a). Fault-tolerant clustering of wireless sensor networks, *Proceedings of the 2003 IEEE Wireless Communications and Networking, 2003. WCNC 2003.*, Vol. 3, pp. 1579 –1584 vol.3.

Gupta, G. & Younis, M. (2003b). Load-balanced clustering of wireless sensor networks, *Proceedings of the IEEE International Conference on Communications, 2003. ICC '03.*, Vol. 3, pp. 1848 – 1852 vol.3.

Gupta, P. & Kumar, P. R. (1998). Critical power for asymptotic connectivity in wireless networks, *Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W.H. Fleming, Ed. W.M.McEneaney , G. Yin, Q Zhang, Birkhauser, Boston*, pp. 547–566.

Gupta, P., Member, S. & Kumar, P. R. (2000). The capacity of wireless networks, *IEEE Transactions on Information Theory* **46**: 388–404.

Gupta, S. & Dave, M. (2008). Real time approach for data placement in wireless sensor networks, *Proceedings of the World Academy of Science, Engineering and Technology*, Vol. 28.

Hall, P. (1985). On continuum percolation, *Ann. Probab.* **9**: 1250–1266.

Handy, M., Haase, M. & Timmermann, D. (2002). Low energy adaptive clustering hierarchy with deterministic cluster-head selection, *Proceedings of the 4th International Workshop on Mobile and Wireless Communications Network, 2002.*, pp. 368 – 372.

Hao, X., Kang, Y. & Wang, Y. (2008). Geographical-based multihop clustering algorithm for distributed wireless sensor network, *Proceedings of the 7th World Congress on Intelligent Control and Automation, 2008. WCICA 2008.*, pp. 3304 –3309.

Heinzelman, W. B. (2000). *Application-specific protocol architectures for wireless networks*, PhD thesis. Supervisor-Chandrakasan, Anantha P. and Supervisor-Balakrishnan, Hari.

Heinzelman, W., Chandrakasan, A. & Balakrishnan, H. (2002). An application-specific protocol architecture for wireless microsensor networks, *Wireless Communications, IEEE Transactions on* **1**(4): 660 – 670.

Henderson, T., Dekhil, M., Morris, S., Chen, Y. & Thompson, W. (1998). Smart sensor snow, *Proceedings of the 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems, 1998.*, Vol. 3, pp. 1377 –1382 vol.3.

Henderson, T., Venkataraman, R. & Choikim, G. (2004). Reaction-diffusion patterns in smart sensor networks, *Proceedings of the 2004 IEEE International Conference on Robotics and Automation, 2004.*, Vol. 1, pp. 654 – 658 Vol.1.

Hou, T.-C. & Li, V. (1986). Transmission range control in multihop packet radio networks, *IEEE Transactions on Communications,* **34**(1): 38 – 44.

Huang, G., Li, X. & He, J. (2006). Dynamic minimal spanning tree routing protocol for large wireless sensor networks, *Proceedings of the 2006 1ST IEEE Conference on Industrial Electronics and Applications*, pp. 1 –5.

Huang, K.-C., Yen, Y.-S. & Chao, H.-C. (2007). Tree-clustered data gathering protocol (tcdgp) for wireless sensor networks, *Proceedings of Future Generation Communication and Networking (FGCN 2007)*, Vol. 2, pp. 31 –36.

Jang, K. Y., Kim, K. T. & Youn, H. Y. (2007). An energy efficient routing scheme for wireless sensor networks, *Proceedings of the International Conference on Computational Science and its Applications, 2007. ICCSA 2007.*, pp. 399 –404.

Jung, S.-M., Han, Y.-J. & Chung, T.-M. (2007). The concentric clustering scheme for efficient energy consumption in the pegasis, *Proceedings of the 9th International Conference on Advanced Communication Technology*, Vol. 1, pp. 260 –265.

Kawadia, V. & Kumar, P. (2003). Power control and clustering in ad hoc networks, *Proceedings of the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. INFOCOM 2003. IEEE Societies*, Vol. 1, pp. 459 – 469 vol.1.

Kim, N., Heo, J., Kim, H. S. & Kwon, W. H. (2008). Reconfiguration of clusterheads for load balancing in wireless sensor networks, *Comput. Commun.* **31**(1): 153–159.

Kirkwood, J. R. & Wayne, C. E. (1983). *Percolation in continuous systems*.

Klaoudatou, E., Konstantinou, E., Kambourakis, G. & Gritzalis, S. (2008). Clustering oriented architectures in medical sensor environments, *Proceedings of the Third International Conference on Availability, Reliability and Security, 2008. ARES 08.*, pp. 929 –934.

Kleinrock, L. & Silvester, J. (1978). Optimum transmission radii for packet radio networks or why six is a magic number, *Proceedings of the National Telecommunications Conference, Birmingham, Alabama*, pp. 4.3.2–4.3.5.

Kumarawadu, P., Dechene, D., Luccini, M. & Sauer, A. (2008). Algorithms for node clustering in wireless sensor networks: A survey, *Proceedings of the 4th International Conference on Information and Automation for Sustainability, 2008. ICIAFS 2008.*, pp. 295 –300.

Kwon, T. J. & Gerla, M. (1999). Clustering with power control, *Proceedings of the IEEE Military Communications Conference Proceedings, 1999. MILCOM 1999.*, Vol. 2, pp. 1424 –1428 vol.2.

Li, L., Dong, S.-S. & Wen, X.-M. (2006). Adaptive clustering for mobile wireless networks, *The Journal of China Universities of Posts and Telecommunications* **13, issue 3**: 71–75.

Liang, Y. & Yu, H. (2005). Energy adaptive cluster-head selection for wireless sensor networks, *Proceedings of the Sixth International Conference on Parallel and Distributed Computing, Applications and Technologies, 2005. PDCAT 2005.*, pp. 634 – 638.

Lijun, L., Hongtao, W. & Peng, C. (2006). Discuss in round rotation policy of hierarchical route in wireless sensor networks, *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing, 2006. WiCOM 2006.*, pp. 1 –5.

Lin, C. & Gerla, M. (1997). Adaptive clustering for mobile wireless networks, *Selected Areas in Communications, IEEE Journal on* **15**(7): 1265 –1275.

Lindsey, S. & Raghavendra, C. (2002). Pegasis: Power-efficient gathering in sensor information systems, *Proceedings of the IEEE Aerospace Conference, 2002.*, Vol. 3, pp. 3–1125 – 3–1130 vol.3.

Liu, C.-M., Lee, C.-H. & Wang, L.-C. (2007). Distributed clustering algorithms for data-gathering in wireless mobile sensor networks, *J. Parallel Distrib. Comput.* **67**(11): 1187–1200.

Liu, Y., Gao, J., Zhu, L. & Zhang, Y. (2009). A clustering algorithm based on communication facility in wsn, *Proceedings of the 2009 WRI International Conference on Communications and Mobile Computing*, IEEE Computer Society, Washington, DC, USA, pp. 76–80.

Loscri, V., Morabito, G. & Marano, S. (2005). A two-levels hierarchy for low-energy adaptive clustering hierarchy (tl-leach), *Proceedings Vehicular Technology Conference, 2005. VTC-2005-Fall. 2005 IEEE 62nd*, Vol. 3, pp. 1809 – 1813.

McLaughlan, B. & Akkaya, K. (2007). Coverage-based clustering of wireless sensor and actor networks, *Proceedings of the IEEE International Conference on Pervasive Services*, pp. 45 –54.

Muruganathan, S., Ma, D., Bhasin, R. & Fapojuwo, A. (2005). A centralized energy-efficient routing protocol for wireless sensor networks, *Communications Magazine, IEEE* **43**(3): S8 – 13.

Nagpal, R. & Coore, D. (1998). An algorithm for group formation in an amorphous computer, *Proceedings of the Tenth International Conference on Parallel and Distributed Systems (PDCS)*.

Nam, D.-H. & Min, H.-K. (2007). An efficient ad-hoc routing using a hybrid clustering method in a wireless sensor network, *Proceedings of the Third IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, 2007. WiMOB 2007.*, pp. 60 –60.

Nocetti, F. G., Gonzalez, J. S. & Stojmenovic, I. (2003). Connectivity based k-hop clustering in wireless networks, *Telecommunication Systems* **22**: 1–4.

Philips, T., Panwar, S. & Tantawi, A. (1989). Connectivity properties of a packet radio network model, *Information Theory, IEEE Transactions on* **35**(5): 1044 –1047.

Qian, Y., Zhou, J., Qian, L. & Chen, K. (2006). Highly scalable multihop clustering algorithm for wireless sensor networks, *Proceedings of the 2006 International Conference on Communications, Circuits and Systems Proceedings*, Vol. 3, pp. 1527 –1531.

Qing, L., Zhu, Q. & Wang, M. (2006). Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor networks, *Comput. Commun.* **29**(12): 2230–2237.

Rajiullah, M. & Shimamoto, S. (2007). An energy-aware periodical data gathering protocol using deterministic clustering in wireless sensor networks (wsn), *Proceedings of the Wireless Communications and Networking Conference, 2007.WCNC 2007. IEEE*, pp. 3014 –3018.

Rasheed, T., Javaid, U., Meddour, D.-E., Reynaud, L. & Al Agha, K. (2007). An efficient stable clustering algorithm for scalable mobile multi-hop networks, *Proceedings of the 4th IEEE Consumer Communications and Networking Conference, 2007. CCNC 2007.*, pp. 89 –94.

Satapathy, S. & Sarma, N. (2006). Treepsi: tree based energy efficient protocol for sensor information, *Proceedings of the 2006 IFIP International Conference on Wireless and Optical Communications Networks*, pp. 4 pp. –4.

Selvakennedy, S. & Sinnappan, S. (2007). An adaptive data dissemination strategy for wireless sensor networks, *Int. J. Distrib. Sen. Netw.* **3**(1): 23–40.

Shakkottai, S., Srikant, R. & Shroff, N. (2003). Unreliable sensor grids: coverage, connectivity and diameter, *Proceedings of the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. INFOCOM 2003. IEEE Societies*, Vol. 2, pp. 1073 – 1083 vol.2.

Takagi, H. & Kleinrock, L. (1984). Optimal transmission ranges for randomly distributed packet radio terminals, *IEEE Trans. Commun* **32**: 246–257.

Tian, Y., Wang, Y. & Zhang, S.-F. (2007). A novel chain-cluster based routing protocol for wireless sensor networks, *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007.*, pp. 2456 –2459.

Tillapart, P., Thammarojsakul, S., Thumthawatworn, T. & Santiprabhob, P. (2005). An approach to hybrid clustering and routing in wireless sensor networks, *Proceedings of the 2005 IEEE Aerospace Conference*, pp. 1 –8.

Vlajic, N. & Xia, D. (2006). Wireless sensor networks: To cluster or not to cluster?, *Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks*, IEEE Computer Society, Washington, DC, USA, pp. 258–268.

Voronoï, G. (1907). Nouvelles applications des paramètres continus Ãă la théorie des formes quadratiques, *Journal für die Reine und Angewandte Mathematik* pp. 133:97–178.

Wang, H., Yu, X., Kong, D., Yan, X. & Ma, X. (2007). Route protocol of wireless sensor networks based on dynamic setting cluster, *Proceedings of the International Conference on Information Acquisition, 2007. ICIA '07.*, pp. 112 –117.

Wang, K., Ayyash, S. A., Little, T. D. C. & Basu, P. (2005). Attribute-based clustering for information dissemination in wireless sensor networks, *Proceedings of The 2nd Annual IEEE Communications Society Conf. on Sensor and Ad Hoc Communications and Networks (SECON'05)*.

Wen, C.-Y. & Sethares, W. A. (2005). Automatic decentralized clustering for wireless sensor networks, *EURASIP J. Wirel. Commun. Netw.* **2005**(5): 686–697.

Weniger, K. & Zitterbart, M. (2004). Adress autoconfiguration in mobile ad hoc networks: Current approches and future directions, *IEEE network* **18**(4): 6–11.

Xia, D. & Vlajic, N. (2006). Near-optimal node clustering in wireless sensor networks for environment monitoring, *Proceedings of the Canadian Conference on Electrical and Computer Engineering, 2006. CCECE '06.*, pp. 1825 –1829.

Xiangning, F. & Yulin, S. (2007). Improvement on leach protocol of wireless sensor network, *Proceedings of the International Conference on Sensor Technologies and Applications, 2007. SensorComm 2007.*, pp. 260 –264.

Xu, K. & Gerla, M. (2002). A heterogeneous routing protocol based on a new stable clustering scheme, *Proceedings of the MILCOM 2002 Conference.*, Vol. 2, pp. 838 – 843 vol.2.

Xue, F. & Kumar, P. R. (2002). The number of neighbors needed for connectivity of wireless networks, *Wireless Networks* **10**: 169–181.

ya Zhang, W., ze Liang, Z., guang Hou, Z. & Tan, M. (2007). A power efficient routing protocol for wireless sensor network, *Proceedings of the 2007 IEEE International Conference on Networking, Sensing and Control,*, pp. 20 –25.

Ye, M., Li, C., Chen, G. & Wu, J. (2005). Eecs: an energy efficient clustering scheme in wireless sensor networks, *Proceedings of the 24th IEEE International Performance, Computing, and Communications Conference, 2005. IPCCC 2005.*, pp. 535 – 540.

Yiming, F. & Jianjun, Y. (2007). The communication protocol for wireless sensor network about leach, *Proceedings of the International Conference on Computational Intelligence and Security Workshops, 2007. CISW 2007.*, pp. 550 –553.

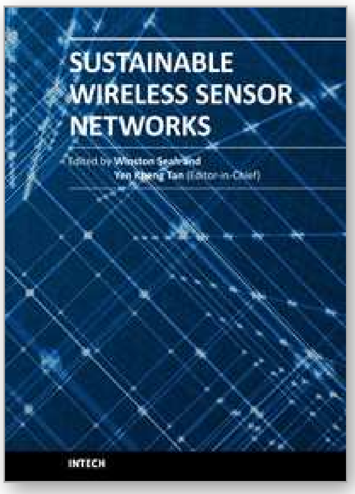
Younis, O. & Fahmy, S. (2004). Distributed clustering in ad-hoc sensor networks: a hybrid, energy-efficient approach, *Proceedings of the Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies, INFOCOM 2004.*, Vol. 1, p. 640.

Youssef, A., Younis, M., Youssef, M. & Agrawala, A. (2006). Wsn16-5: Distributed formation of overlapping multi-hop clusters in wireless sensor networks, *IEEE Proceedings of the Global Telecommunications Conference, 2006. GLOBECOM '06.*, pp. 1 –6.

Yuan Sun, E., Sun, Y. & Belding-Royer, E. M. (2003). Dynamic address configuration in mobile ad hoc networks, *Technical report*, Computer Science, UCSB, Tech. Rep.

Zhang, H. & Arora, A. (2003). Gs3: scalable self-configuration and self-healing in wireless sensor networks, *Computer Networks* pp. 459–480.

Zhou, H., Ni, L. & Mutka, M. (2003). Prophet address allocation for large scale manets, *Proceedings of the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. INFOCOM 2003. IEEE Societies*, Vol. 2, pp. 1304 – 1311 vol.2.



Sustainable Wireless Sensor Networks

Edited by Yen Kheng Tan

ISBN 978-953-307-297-5
Hard cover, 574 pages
Publisher InTech
Published online 14, December, 2010
Published in print edition December, 2010

Wireless Sensor Networks came into prominence around the start of this millennium motivated by the omnipresent scenario of small-sized sensors with limited power deployed in large numbers over an area to monitor different phenomenon. The sole motivation of a large portion of research efforts has been to maximize the lifetime of the network, where network lifetime is typically measured from the instant of deployment to the point when one of the nodes has expended its limited power source and becomes in-operational â€“ commonly referred as first node failure. Over the years, research has increasingly adopted ideas from wireless communications as well as embedded systems development in order to move this technology closer to realistic deployment scenarios. In such a rich research area as wireless sensor networks, it is difficult if not impossible to provide a comprehensive coverage of all relevant aspects. In this book, we hope to give the reader with a snapshot of some aspects of wireless sensor networks research that provides both a high level overview as well as detailed discussion on specific areas.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821