A dynamic clustering construction for wireless sensor networks

Pamba Capo-Chichi Eugène, David Martins, Hervé Guyennet, Violeta Felea

▶ To cite this version:

Pamba Capo-Chichi Eugène, David Martins, Hervé Guyennet, Violeta Felea. A dynamic clustering construction for wireless sensor networks. CTS'09, int. symposium on Collaborative Technologies and Systems, 2009, United States. pp.565–570. hal-00563304

HAL Id: hal-00563304
https://hal.science/hal-00563304

Submitted on 4 Feb 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Dynamic Clustering Construction for Wireless Sensor Networks

Eugene Pamba Capo-Chichi *, David Martins *, Hervé Guyennet *and Violeta Felea *
*Computer Science Department
University of Franche-Comté, France
Email: {mpamba,dmartins,hguyennet,vfelea}@lifc.univ-fcomte.fr

***Abstract*—Research in sensor networks has focused on development of energy efficient and secure infrastructures. In this article, we introduce a new approach to organize sensor networks in clusters in order to reduce energy dissipation. Our contribution is an heuristic to define the number of clusters and also an efficient manner to choose cluster heads by minimizing the distance between the cluster heads and its cluster nodes. Inspired from LEACH, a well-known TDMA cluster-based sensor network architecture, we introduce a new method for building and maintaining clusters using the paradigm of a soccer team. In this work, a new algorithm called OH-Kmeans, based on the K-means algorithm, is used to find dynamically the number of clusters and form them guaranteeing direct transmission between the cluster heads and cluster nodes.**

***Index Terms*—wireless sensor networks, clustering, security, energy-efficient algorithms**

I. Introduction

Facilities of deployment and reduced costs have encouraged the use of wireless sensor networks in many applications. Advances in processor, memory and radio transmission technology have also enabled the possibility to develop distributed networks with sensors. However this kind of networks is limited by the critical problem of the energy consumption [ASSC02].

Because sensors have reduced sizes, their batteries are small and are thus limited in energy. We have to minimize energy consumption in order to guarantee the lifetime of the network. The most consuming activity of sensors is radio communication [HNNB06]. Two main methods are used in solving the problem of energy consumption: data aggregation and mode states scheduling (switch to sleeping mode for example) because these methods reduce energy spent in radio communications. Data aggregation is any process in which information is gathered and expressed in a summary form with minimum information loss [KEW02]. This method is also used in order to reduce energy consumption in wireless sensor networks by eliminating redundant data transmissions.

In this article, we introduce another approach of hierarchical organization of sensors. Indeed, to reduce the number of radio communications, we have to organize the sensor field in clusters. In the context where sensors are geographically close to each other, there is a strong probability that they sense the same information, so it is interesting to send minimum redundant information to the sink to save energy in the network.

Clustering is the classification of objects into groups, known as clusters, based on similarity. This similarity is often the proximity according to some defined distance measure. The objects, that are being clustered here, are sensor nodes. In each cluster, a sensor, called cluster head, gathers data from the others. In this paper, we define an algorithm to choose the number of clusters and form them as to minimize the distance between the cluster heads and the other cluster nodes. This approach uses the barycenter concept and a new algorithm named OH-Kmeans based on the K-means algorithm. This work is inspired from LEACH but according to [Hei00], LEACH offers no guarantee about placement and/or number of cluster head nodes; the aim of this article is to give these guarantees.

This paper is organized as follows: Section 2 is the related work focused on the presentation of a well known clustering algorithm for wireless sensor networks named LEACH and its versions such as LEACH-F and LEACH-C. In Section 3 we present a new model of energy management in wireless sensor networks based on the paradigm of the soccer game using a new algorithm inspired from the K-means algorithm called as OH-Kmeans. Section 4 shows simulation results of the OH-Kmeans algorithm. Finally, we conclude.

II. Related work

Several works in sensor networks intend to solve the problem of energy waste in sensor networks. One of these approaches is a protocol known as LEACH (Low-Energy Adaptive Clustering Hierarchy) [Hei00] which proposes energy-efficient communications.

LEACH is a protocol based on clustering: a node elects itself to be cluster head with some probability and broadcasts an advertisement message to all the other nodes in the network. A node is selected to join a cluster based on the power radio signal received from the cluster head. Being cluster head is more energy consuming than not being one. All nodes in the network may be cluster head during some period of time. The protocol is a two-phase model: the set-up and the steady-state phase. The set-up phase allows to organize the clusters.

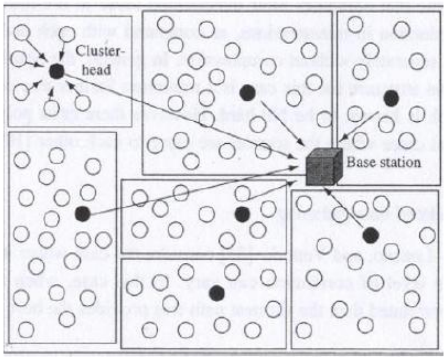

Fig. 1. LEACH: protocol based on clustering [HK05]

The steady-state phase consists of several cycles where all nodes have their communication slots periodically. The nodes send their data to the cluster head that aggregates data and sends it to the base station at the end of each cycle. After a certain amount of time, the TDMA round ends and the network re-enters the set-up phase. Experimental results show that the hierarchical architecture of LEACH [Hei00] optimizes communication cost compared to direct communication.

A version of LEACH called LEACH-C (LEACH-Centralized) [Hei00] uses a centralized cluster formation algorithm to form clusters.

Centralized approaches allow to have a global view of the network and consequently identify relevant cluster heads and number. Even though all information is available in this kind of approaches, determining an adequate value of the number of clusters is still a NP-hard problem.

LEACH-C uses simulated annealing to search for a near-optimal number of clusters. The protocol uses the same steady-state protocol as LEACH. During the set-up phase, the base station receives information from each node about their current location and energy level. After that, the base station runs the centralized cluster formation algorithm to determine cluster heads and clusters for that round. LEACH-C chooses cluster heads randomly but the base station makes sure that only nodes with "enough" energy are participating in the cluster head selection. Once the clusters are created, the base station broadcasts the information to every network node.

There is another version of LEACH named LEACH-F (LEACH with Fixed clusters) [Hei00] which forms clusters and then fixes them. There is no set-up overhead at the beginning of each round; this is an interesting advantage because overhead is one of the reasons for energy waste in sensor networks. To form clusters, LEACH-F uses the same centralized cluster formation algorithm as LEACH-C. The fixed clusters in LEACH-F do not allow new nodes to be added to the system and as such, do not adjust their behavior in case of nodes failure.

K-means is a well known algorithm that solves the clustering problem. The aim of this algorithm is to organize a given data set into a certain number of clusters (assume K clusters). The number of K is fixed at the beginning. The algorithm is composed of the following steps:

1) *Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.*
2) *Assign each object to the closest centroid using a distance metric.*
3) *When all objects have been assigned, compute the positions of the new K centroids as barycenters of the clusters resulting from the previous step.*
4) *Repeat Steps 2 and 3 until the obtained centroids are the same as in the previous iteration.*

Since choosing centroids is very important, the best option is to place them, as much as possible, far away from each other. Indeed, in LEACH, node elects itself to be cluster head and consequently, we may obtain close centroids. This clustering method needs improvement in order to assure shrewd choice of the cluster heads and of their number.

III. System model

In this approach, we assimilate a sensor network to a soccer team and we use soccer principles to organize the network so as to minimize the energy consumption. In this model, the network is called

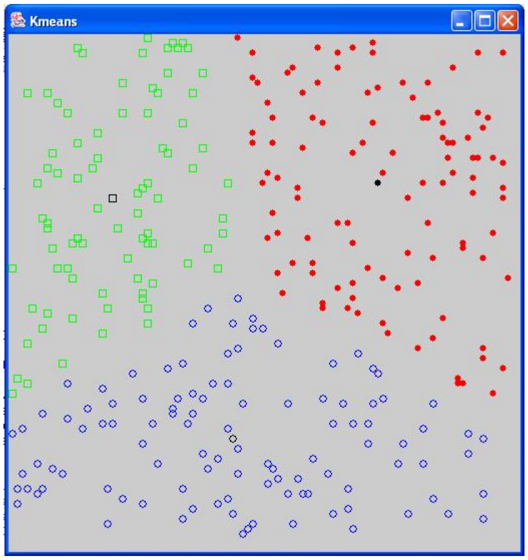

Fig. 2. Cluster construction using the K-means algorithm with K=3

team, sensors are the **players**, the sink is the **goal** and the information to send to the sink is the **ball**. To explain further in details, the team needs to be organized around some particular players who can facilitate ball exchanges. A good method for an interesting deployment of sensor networks is the clustering which reveals already interesting in [HCB00]. Clustering permits to avoid sending redundant information to the sink because nodes geographically close collect the same information with strong probability and by clustering, the cluster head can aggregate redundant data. Clustering used in LEACH [HCB00] reveals to be effective but on the other hand, LEACH does not offer any guarantee about placement and number of cluster head nodes. Our contribution is meant to introduce another clustering method which gives these guarantees. In the following lines we present this new clustering method for sensor networks.

A. Our clustering method

Generally, clustering is carried out in the following way: nodes are divided into clusters and in each cluster, one node, named cluster head is responsible not only for the cluster formation but also for its maintenance. In this way, several decisions need to be made: determination of the number of cluster heads, the choice of the cluster heads between the network nodes and the cluster organisation. In our model, a cluster's head is named **captain**. The choice of the number of captains in the network is very important but not easy. If we consider K the number of captains, there may be several methods to choose K:

- K is fixed according to the knowledge of the field and the application,
- K is fixed in an empirical way: testing various values of K may help choosing K in order to optimize a validity and quality of the regrouping. In this work, a criterion of quality is minimization of energy consumption by minimizing the distance between the captain and the cluster nodes.

In this article, we propose a new way to choose K. Sensors are characterized by a *Sensing Range (SR)* and a *Communication Range (CR)*. According to [GCSRS08], the Sensing Range is the monitored area of a sensor which is modeled as a disk of radius SR, centered on the node itself. Two sensors are communicating neighbours, or simply neighbours, if and only if the distance between them is at most CR.

In order to be able to relay information, nodes must communicate with their neighbours. In this way, we will compare the deployment area to a rectangle whose surface is equal to length*width. Let CR be the transmission range, an interesting K seems to be:

$$K = rounded(length/(CR*2))*rounded(width/(CR*2))$$

where CR*2 is the diameter of the circle formed with the radius CR.

After having determined K, we can apply the K-means algorithm with some modifications like the choice of K presented previously. To form clusters, we introduce a new algorithm called OH-Kmeans (One Hop Kmeans) which uses the K-means algorithm. In this work, we consider that the sink has an unlimited energy power and a good computing power to run the OH-Kmeans algorithm. The aim of this algorithm is to guarantee one hop exchange between captain and players. The following lines define the different algorithms used by the OH-Kmeans algorithm.

The first function named `Kmeans` is an implementation of the K-means algorithm.

The second function, named `OneHop`, counts the number of nodes, one hop at most of the cluster head and returns this number.

The third function `TestKInf` verifies that it is possible to decrease K keeping one hop between a cluster head and its cluster nodes and returns null if it is impossible to do so.

The OH-Kmeans algorithm uses these previous functions and changes K dynamically.

This algorithm inspired from K-means named OH-Kmeans (One Hop Kmeans) allows to find a number of K clusters which guarantees a direct information transmission (one hop) between captains and players. The sink needs to know the position of all players at

Algorithm 1 Kmeans: TableofCluster Kmeans(int K)

```
  Step 1 - Choose K points randomly into the space represented by the network that are being clustered. These points represent initial captains (they may be virtual).
  Step 2 - Assign each player to the closest captains using the euclidian distance metric.
  Step 3 - When all players have been assigned, compute the positions of the new K captains as barycenters of the clusters resulting from the previous step.
  Step 4 - Repeat Steps 2 and 3 until the obtained captains are the same as in the previous iteration.
  return TableofClustersCreated containing each cluster with its captain
```

Algorithm 2 OneHop: int OneHop (TableofCluster)

```
  $compteur \leftarrow 0$
  for all $nodes\ n_i$ of TableofCluster in each $cluster_i$ do
    if $Euclidean\_distance(clusterhead_i, n_i) > CR$ then
      $compteur \leftarrow compteur + 1$
    end if
  end for
  return $compteur$
```

the beginning. During the execution of OH-Kmeans, captains are chosen and they can send messages.

Algorithm 3 TestKInf: TableofCluster TestKInf(int K)

```
  let tab be a table of clusters
  $tab \leftarrow Kmeans(K)$
  if $(OneHop(tab) == 0)$ then
    return $tab$
  else
    return $null$
  end if
```

Algorithm 4 OH-Kmeans (K, communication range)

```
  $CR \leftarrow$ communication range
  Choice of K
  $TableofCluster \leftarrow$ Kmeans(K)
  $test \leftarrow$ OneHop($TableofCluster$)
  if $(test \neq 0)$ then
    while $(test \neq 0)$ do
      $K \leftarrow K + 1$
      $TableofCluster \leftarrow$ Kmeans(K)
      $test \leftarrow$ OneHop($TableofCluster$)
    end while
  else
    $out \leftarrow$ false
    while $(not\ out)$ do
      $K \leftarrow K - 1$
      $newtable \leftarrow$ TestKInf(K)
      if $(newtable \neq null)$ then
        $TableofCluster \leftarrow newtable$
      else
        $out \leftarrow$ true
      end if
    end while
  end if
```

Once the initial captains are selected, they send messages to players to create clusters. Nodes choose the closest captain (at one hop) and send to it the information (position and energy level). These positions enable to maintain the cluster organisation (details will be given in the next paragraph which is related to the captain rotation).

Becoming captain is a very energetic task because the captain receives all information that is why it is very important to change the captain after a period.

B. Captain rotation

Because all balls in the network are sent through the captain, the latter is an energy consuming node, which makes its choice very important. At the end of each period the captain aggregates information before sending it to the *goal*. In this section we define the manner in which a captain is chosen.

When a captain receives messages from players which want to join clusters at the beginning of a period, it receives their positions and their energy level to choose the future captain. The captain chooses the best future captain among them depending on the criterion which follow:

- **The energy level**
  This energy level corresponds to the level of sensor's battery because captains must exchange

information with the other nodes which are not captain of cluster and this operation requires a considerable level of energy.

- **The number of times that it was captain**
  In order to spare the captain's energy, we need to consider the number of times a node has already been captain.
- **The density of nodes in a given range**
  To have several nodes in a given range can facilitate the exchange of information. If nodes are close to the captain, information is sent easily, in one hop transmissions.
- **Proximity compared to the sink**
  Information collected by the cluster's captain must be, after their aggregation, sent to the sink thus it would be less expensive to choose a captain close to the sink.

This work is a mixture between LEACH-C and LEACH-F because at the beginning the system is centralized but when OH-Kmeans stops and the good clusters are formed, the system becomes stable. LEACH-F does not allow new nodes to be added but in this model, we can add new nodes if they send their position to the nearest captain. In this model, compared to LEACH-C, it is not the sink which chooses the future captain but the previous captain. After having created the clusters, it is important to use collective intelligence of all players in order to organize the game. After cluster construction, the steady state phase is the same as in LEACH.

IV. Experimental results

OH-kmeans and K-means have been simulated using Java platform. Results of simulations are presented here. All experiments have been done on a 100*100 surface with sensor's communication range equal to 15.

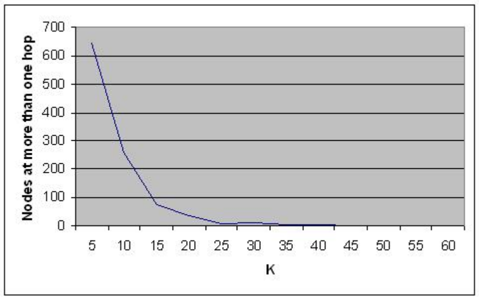

Fig. 3. Number of nodes at more than one hop for different values of K using the K-means algorithm

We obtained Figure 3 for 1000 nodes deployed on a 100*100 surface. We can see that when the number of captains, K, in the network increases, the number of nodes, with a distance greater than one hop, decreases. Then we decided to implement an algorithm which allows to modify K dynamically. This new algorithm, called OH-Kmeans, allows to find the adapted value of K which guarantees one hop exchange between each captain and nodes in its cluster.

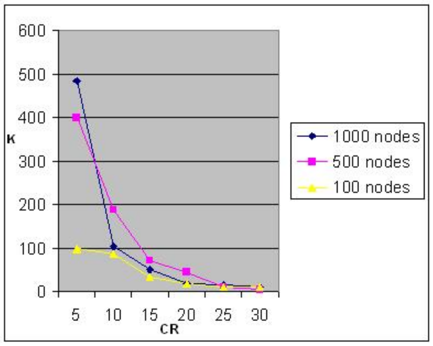

Fig. 4. Variation of K with different values of communication range using the K-means algorithm

In Figure 4, we varied the communication range between 5 and 30, for 100, 500, 1000 deployed nodes on a 100*100 surface and we remarked that when the communication range increases, the number of captains K (cluster-heads) which allows to have one hop exchange decreases.

This graphics shows that CR is an important factor just like K, the number of cluster heads, in the clustering technic.

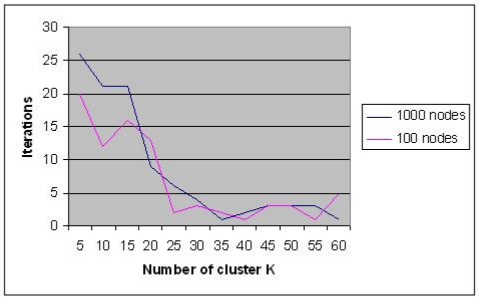

Fig. 5. Number of iterations for different values of K using the OH-Kmeans algorithm

In Figure 5, we can see that for different networks with 1000 nodes or 100 nodes, the number of iterations with OH-Kmeans which guarantees one hop exchanges converges rapidly. According to the previous section, an interesting K seems to be:

$$K = rounded(length/(CR*2))*rounded(width/(CR*2))$$

In this example, we obtain:

$$K = rounded(100/(15*2)) * rounded(100/(15*2)) = 9$$

According to Figure 5, for 100 proposed nodes and $K = 9$, the number of iterations, for the OH-Kmeans algorithm, is between 12 and 20 iterations but the simulated annealing used in LEACH, which allows to find the optimal number of clusters, converges in 200 to 500 iterations for a network of 100 nodes.

V. Conclusion

In conclusion, this paper introduces a new method to organize sensor networks, guaranteeing one hop exchange between cluster head and cluster nodes. This approach is based on an algorithm named OH-Kmeans. This algorithm gives an heuristic to find dynamically the number of clusters. OH-kmeans is inspired from the K-means algorithm and introduces a new manner to choose the number of clusters dynamically. Experimental results show that OH-Kmeans, for 100 nodes or 1000 nodes, converges more quickly than simulated annealing (used in LEACH). So, for any values of Communication Range (CR) or number of clusters K, with OH-kmeans, the number of nodes at more than one hop of its cluster head always equals zero. Future work concerns the integration of this new clustering method into the LEACH communication approach.

References

[ASSC02] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Comput. Netw.*, 38(4):393–422, 2002.

[GCSRS08] Antoine Gallais, Jean Carle, David Simplot-Ryl, and Ivan Stojmenović. Localized sensor area coverage with low communication overhead. *IEEE Transactions on Mobile Computing*, 7(5):661–672, 2008.

[HCB00] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *HICSS '00: Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8*, page 8020, Washington, DC, USA, 2000. IEEE Computer Society.

[Hei00] Wendi Beth Heinzelman. *Application-specific protocol architectures for wireless networks.* PhD thesis, 2000.

[HK05] A. Willig H. Karl. *Protocols and Architectures for Wireless Sensor Networks.* Wiley, May 2005.

[HNNB06] Ewa Hansen, Jonas Neander, Mikael Nolin, and Mats Björkman. Energy-efficient cluster formation for large sensor networks using a minimum separation distance. In *The Fifth Annual Mediterranean Ad Hoc Networking Workshop*, June 2006.

[KEW02] Bhaskar Krishnamachari, Deborah Estrin, and Stephen B. Wicker. The impact of data aggregation in wireless sensor networks. In *ICDCSW '02: Proceedings of the 22nd International Conference on Distributed Computing Systems*, pages 575–578, Washington, DC, USA, 2002. IEEE Computer Society.