Telecommunication Systems manuscript No.
(will be inserted by the editor)

A Game Theoretical Approach to Clustering of Ad-Hoc and Sensor Networks

Georgios Koltsidas · Fotini-Niovi Pavlidou

Abstract Game theory has been used for decades in fields of science such as economics and biology, but recently it was used to model routing and packet forwarding in wireless ad-hoc and sensor networks. However, the clustering problem, related to self-organization of nodes into large groups, has not been studied under this framework. In this work our objective is to provide a game theoretical modeling of clustering for ad-hoc and sensor networks. The analysis is based on a non-cooperative game approach where each sensor behaves selfishly in order to conserve its energy and thus maximize its lifespan. We prove the Nash Equilibria of the game for pure and mixed strategies, the expected payoffs and the price of anarchy corresponding to these equilibria. Then, we use this analysis to formulate a clustering mechanism (which we called Clustered Routing for Selfish Sensors - CROSS), that can be applied to sensor networks in practice. Comparing this mechanism to a popular clustering technique, we show via simulations that CROSS achieves a performance similar to that of a very popular clustering algorithm.

Keywords Game Theory · Clustering · Ad Hoc and Sensor Networks · Nash Equilibrium

1 Introduction

Wireless sensor networks (WSNs) are constantly gaining popularity the last few years in both the research community and commercial applications. The reason is their unique characteristics. A typical WSN is comprised of a large set of wireless nodes, with sensing, monitoring and processing capabilities, deployed in an ad hoc fashion that typically coordinate to perform a common task. These wireless nodes are usually small, low-cost, autonomous, battery-operated devices, with limited energy capacity and computational processing capability. Therefore, energy-aware mechanisms are required, so as to ensure a long-lasting operation without the need for battery replacement [2].

Since the number of nodes in such a network can extend to very large values, an efficient method to reduce the expenditure of the batteries' power is a grouping technique known as *clustering*. The essential operation in clustering consist in selecting a set of nodes to become *clusterheads* (CH), and group the rest of the nodes in clusters, around every clusterhead. The clusterhead is responsible for coordination among the nodes inside its cluster, and for forwarding the collected data to the sink node, usually after efficiently aggregating them. Clustering is particularly useful for applications that require scalability to hundreds or thousands of nodes. Scalability in this context implies the need for load balancing and adaptability to changes in network size, node density and topology. Moreover, self-organized clustering algorithms constitute natural solutions for large networks or networks lacking centralized control, such as ad-hoc and sensor networks.

On the other hand, game theory [12] has been developed and extensively used in the context of economy and biology. It is a very powerful mathematical tool for analyzing and predicting the behavior of rational and selfish entities. Due to its interesting and sometimes unexpected results, its popularity reached the field of communications and networking technology [3]. Recently, it has been used to model packet forwarding in wireless ad-hoc networks with energy constraints ([7], [6], [13]) or as a basis to propose cooperation enforcements mechanisms for these networks ([15], [10], [4]). In this work we use the terminology and theorems of game theory to form and analyze the problem of clustering in sensor networks.

The rest of the paper is organized as following: In section 2 an overview of the related works on ad-hoc and sen-

Georgios Koltsidas · Fotini-Niovi Pavlidou
Dept. Electrical and Computer Engineering, Aristotle University of Thessaloniki, Panepistimioupolis, 54124, Thessaloniki, Greece
E-mail: {fractgkb, niovi}@auth.gr

sor network clustering algorithms is provided. The following section is dedicated to describing and analyzing the clustering game, finding its equilibria, the expected payoffs and the price of anarchy. In section 4 a new clustering technique is described, based on the aforementioned analysis, and its evaluation is presented in section 5, followed by reports on the results. Finally, section 6 concludes the paper.

2 Related Work

Many sensor networks clustering algorithms have been proposed so far. The most well-known is the Low-Energy Adaptive Clustering Hierarchy (LEACH) algorithm [9]. According to this protocol, the role of the clusterhead is not permanently assigned to particular nodes. Instead, each sensor is randomly self-elected as clusterhead, but a mechanism ensures that all nodes will play this role within a predefined time interval. Consequently, the role of clusterhead is rotated between the nodes in a probabilistic way, so that the energy consuming operation is distributed among all the nodes of the network. Later, a centralized version of LEACH, called LEACH-C has been proposed in [8], where the decision on which nodes will play the role of clusterheads is not distributed, but it is decided by the base station node (alternatively called sink node or just sink). The benefit is that the base station has advanced computation capabilities and practically unlimited power. Besides, by collecting all the information, it maintains a global knowledge of the network and so it can make much better decisions and network planning. The results show that LEACH-C increases the network lifetime due to global knowledge and the ability to ensure that the optimum number of clusterheads is selected in each round.

Another centralized mechanism is described in [11], namely the Base-station Controlled Dynamic Clustering Protocol - (BCDCP). This technique forms balanced clusters, where each cluster has the same number of members to avoid cluster overloading. In addition to this, data from distant clusterheads is sent to the base station via other clusterheads. BCDCP outperforms LEACH both in terms of lifetime and of the number of delivered messages to base station. The Base-station Centralized Simple Protocol (BCSP), on the other hand, aims at extending the network lifetime by basing the clustering decision on the remaining energy of every node. The base-station redistributes the role of clusterhead among the nodes from time to time. BCSP does not require location knowledge, like other algorithms, but each node should send its energy level information along with the sensing information, increasing the overhead. The drawback of the two aforementioned techniques is that due to their centralized implementation, they are not so appropriate for sensor networks with a large number of nodes.

In [5] authors propose a clustering scheme that considers multiple parameters such as mobility, battery power and maximum number of cluster members. The proposed Weighted Clustering Algorithm (WCA) uses a weighted sum of the parameters in a combined metric that should be optimized. The weights permit the protocol's performance to vary among several operating points, depending on the desire of the operator and the particular needs of the application. However, the additive character of its metric means that not all its parameters could be optimized simultaneously, so a compromise among the them needs to be made, limiting its performance.

HEED [16] is another clustering mechanism for ad hoc and sensor networks that makes no assumption regarding the location knowledge and it is completely distributed. Clusterheads are randomly selected on the basis of their residual energy normalized by their initial energy. In addition, it can be combined with a metric that takes the node density into account. HEED is proved to perform better than a modified version of LEACH algorithm in case the nodes' initial energies are not identical.

To the best of our knowledge, very few works used game-theoretic terms to study clustering for sensor networks. In [17] the authors propose a centralized algorithm where the base station (or sink) decides on both the number of clusters and the nodes that will become clusterheads. This decision is based on the information of location and remaining energy of every node. The authors show that this way of clusterhead selection is more energy-efficient than a random one following the LEACH algorithm. However, no theoretical analysis is provided and the clusterhead selection is centralized, an operation that requires excessive overhead and thus consumes additional energy.

In [1], the authors formulate a cooperative game between the nodes of a sensor network. The payoffs for every node depend on three parameters: the reputation, the cooperation and the quality of security of every node. The first one refers to the received signal strength, the second one to the percentage of the received packets that a node has forwarded and the third one refers to the ratio of the exchanged messages between two nodes to the dropped messages between them. The strategy of the players corresponds to a probability of cooperation. The clusterheads of the clusters are initially selected. Nodes' mobility results in changes in cluster formation and new clusters may be formed or others may be deleted. Simulations revealed that although initially the number of clusters is large, cooperation between the nodes increases with time and thus the network converges to a constant value of clusterheads. Although the comparison of the proposed methodology seems to need less message exchanges than a technique based only on the distance metric, the authors do not provide any information regarding the energy consumption, which is crucial for sensor networks.

Moreover, the main objective of the algorithm is trust enforcement and trust maintenance between sensors. The authors do not mention what will happen in the case of static nodes, where the probability of some nodes being closer than others is different between the sensors.

In this paper we attempt to examine the problem of clustering from a new perspective: We assume that the nodes in the network behave selfishly, meaning that their primary goal is to maximize their own benefits while minimizing their own contribution for the benefit of other nodes. This context applies in energy-aware ad-hoc networks where the nodes are usually individual users acting without any predetermined agreement. Since the devices are mobile, energy is a precious resource that should be consumed wisely. Applying this model to sensor networks is also meaningful, when collocated sensors do not belong to the same authority. Finally, this effort provides another point of view to the problem of clustering specifying the limitations of selfish behavior in this context and possible a new methodology of designing clustering techniques.

3 The Clustering Game

3.1 Analysis and Equilibria

We begin with the definition of the *clustering game* (CG). The clustering game is the game played by the nodes of a network, when the purpose is to select a number of nodes as clusterheads. It actually corresponds to choosing at least one clusterhead from the population of the nodes. The responsibility of the clusterhead is to collect data from all other nodes (the cluster members) and forward them (after efficiently aggregating them) to another node, which is usually located far from the cluster. We formally define the game as $CG =< \mathbf{N}, \mathbf{S}, \mathbf{U} >$, where $\mathbf{N}$ is the set of players, $\mathbf{S} = \{S_i\}$ is the set of the available strategies and $\mathbf{U} = \{U_i\}$ is the set of utility functions of the nodes. The players are the sensors, the N nodes participating in the network. In pure strategies, the strategy space corresponds to two choices: a sensor decides to either declare itself as CH or not. Letting D be the strategy "declare myself as CH" and ND the strategy "do not declare myself as CH", the strategy space is $\mathbf{S} = \{\text{Declare}, \text{Not Declare}\} = \{D, ND\}$. Regarding payoffs, if a player chooses not to become a clusterhead, then if no other node becomes a clusterhead either, its payoff will be zero, as the player will be unable to sent its data towards the sink. If at least one other neighbor declares itself as CH, then its payoff will be v, i.e. the gain in successfully delivering the data to sink. Finally, if the player declares itself as CH, its payoff for successfully delivering the data v will be reduced by an amount equal to the cost c of becoming a clusterhead. So, in that case the final payoff will be $v - c$.

Table 1 The payoffs for the simple two player clustering game

	Declare	Not Declare
Declare	$(v-c, v-c)$	$(v-c, v)$
Not Declare	$(v, v-c)$	$(0,0)$

Let us now discuss the possible equilibria in the case of two players, whose payoffs are summarized in Table 1. It is clear that the game is symmetrical, since the payoff is only dependent on the strategies of the players and not on which player we consider for examination. The strategy (D, D) is not a Nash Equilibrium. This is because each player is better off to change its strategy to ND, as in this case its payoff will be $v > v - c$. For a similar reason the strategy (ND, ND) is not a Nash Equilibrium either, as any player would prefer to deviate and declare itself as a CH, since this leads to a positive payoff. In case the first player plays D and the second one plays ND, then none of them has any incentive to change its choice. Hence, the strategy set (D, ND) is a Nash Equilibrium. For the same reason, (ND, D) is a Nash Equilibrium too. Although these two strategies, (D, ND) and (ND, D), are Nash Equilibria, there is no symmetrical Nash Equilibrium in this game, since no common strategy for all players exists that results in an equilibrium. In the following we provide some useful propositions and theorems corresponding to the N player clustering game.

Extending the game to be played by N players, let $\mathbf{s} = \{s_1, s_2, ..., s_N\}$ be the vector profile of the strategies followed by the players. If no player "Declares", then all players' payoffs are zero. If at least one player, say k, plays "D", then the payoff of all other players except k will be v, while player k's initial gain v will be reduced by the cost of declaring itself. Hence, the utility function $U_i(\mathbf{s})$ of an arbitrary player i has the following form:

$$U_i(\mathbf{s}) = \begin{cases} 0 & \text{, if } s_j = ND, \forall j \in \mathbf{N} \\ v - c & \text{, if } s_i = D \\ v & \text{, if } s_i = ND \text{ and } \exists j \in \mathbf{N} \text{ s.t. } s_j = D \end{cases} \quad (1)$$

In the following we list some helpful propositions, whose proofs are omitted because they are straightforward.

Proposition 1 *For the symmetrical clustering game, the strategy $S_{allD} = \{D, D, D, ..., D\}$ is not a Nash Equilibrium.*

Proposition 2 *For the symmetrical clustering game, the strategy $S_{allND} = \{ND, ND, ND, ..., ND\}$ is not a Nash Equilibrium.*

Proposition 3 *For the symmetrical clustering game, the strategy where a single player plays D and all other players play ND is a Nash Equilibrium and there are N Nash Equilibria in the game.*

Proposition 4 *For the symmetrical clustering game, no symmetric pure strategies Nash Equilibria exist.*

In order to permit the game to have symmetrical Nash Equilibria, we need to allow the players to play mixed strategies. This means that the players choose their strategies randomly following a probability distribution. In other words, every player has now a probability of declaring itself as CH and a probability not doing so. Let us denote the probability of playing D as p and the probability of playing ND as $q = 1 - p$. A interesting theorem follows.

Theorem 1 *For the symmetrical clustering game, a symmetric mixed strategies Nash Equilibrium exists and the equilibrium probability p that a player declares itself as clusterhead is*

$$p = 1 - \left(\frac{c}{v}\right)^{1/(N-1)}.$$

Proof We are going to search for symmetrical Nash Equilibria in mixed strategies, that will correspond to a particular probability p of a node declaring itself as CH, using the methodology found in [14]. We first need to calculate the expected payoff for each choice available. The expected payoff when playing D is $U_D = v - c$, as the payoff is independent of the other players' strategies. The expected payoff when playing ND is

$$\begin{aligned} U_{ND} &= Pr\{\text{no one else declares}\} \cdot 0 \\ &\quad + Pr\{\text{at least someone else declares}\} \cdot v \\ &= v \cdot (1 - Pr\{\text{no one else declares}\}) \\ &= v \cdot \left(1 - q^{N-1}\right) \\ &= v \cdot \left(1 - (1-p)^{N-1}\right). \end{aligned} \tag{2}$$

At the equilibrium, these two payoffs are equal, so that no player has incentive to alter its strategy. Thus,

$$v - c = v \cdot \left(1 - (1-p)^{N-1}\right). \tag{3}$$

Solving the above expression, we can compute the probability p that corresponds to the equilibrium:

$$p = 1 - \left(\frac{c}{v}\right)^{1/(N-1)}. \tag{4}$$

Letting $\omega = c/v < 1$, the above equilibrium probability never exceeds 1 and can be also written as:

$$p = 1 - \omega^{1/(N-1)}. \tag{5}$$

3.2 Asymptotic Analysis

Let us have a more detailed view of the equilibrium probability we have computed. The parameter ω is positive but lower than 1. Thus, the probability we have computed remains within the interval $[0, 1]$. What is more, p falls as the number of players increases, which means that players become less cooperative as their number increases. Another interesting probability is that of at least one player declaring itself as clusterhead:

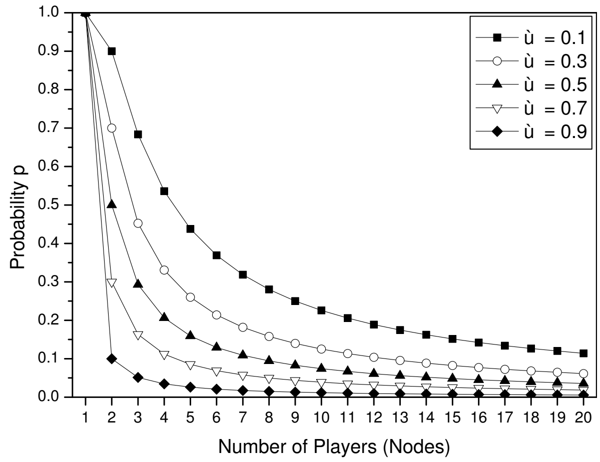

Fig. 1 Probability p of a player declaring itself as CH versus the total number of players (nodes).

$$\begin{aligned} P_A &= Pr\{\text{at least one node plays D}\} \\ &= 1 - Pr\{\text{no one plays D}\} \\ &= 1 - (1-p)^N \\ &= 1 - \omega^{N/(N-1)}. \end{aligned} \tag{6}$$

From the above expression we may observe that when there is only one player in the game, both the probability p of declaring itself as CH and the probability P_A of at least one node is self-elected as CH is equal to 1. For 2 players, $p = 1 - \omega$ and $P_A = 1 - \omega^2$. As N tends towards infinity,

$$\lim_{N \to \infty} p = 0 \tag{7}$$

and

$$\lim_{N \to \infty} P_A = 1 - \omega. \tag{8}$$

Thus, the higher the number of nodes, the less the probability that at least one node declares itself as CH. On the other hand, when N tends to 1 (meaning that there is only one node left), then P_A tends to 1, which means that if a player is alone, then it is always self-declared as CH. Figs. 1 and 2 depict the values of the two probabilities as the number of players increases for 5 different values of the parameter ω (0.1, 0.3. 0.5, 0.7 and 0.9).

These characteristics of the computed probability can be very useful in the designation of a clustering mechanism that will base the cooperation only on the rationality of selfish sensors and will not attempt to force any complex algorithm that each sensor should follow.

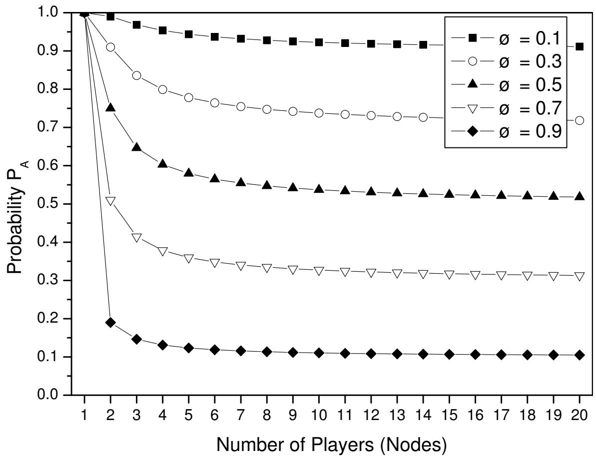

Fig. 2 Probability P_A of at least one player declares itself as CH versus the total number of players (nodes).

3.3 Costs and Payoffs

The above model can be modified to become more realistic if we consider the additional cost in the case of data exchange between the cluster member and the clusterhead, related to the energy expenditure. When a sensor needs to send a data packet of k bits to another sensor, it consumes energy in both ways. Firstly, the energy is spent at the transmitter's electronic circuitry and it is denoted as e_{elec}. Secondly, the energy is consumed by the transmitter's amplifier in order to achieve the required signal level at the receiver for correct decoding and it is denoted as e_{amp}. There are usually two variations of this parameter. The first one correspond to a square law distance attenuation (e_{amp2}) and the second one to an attenuation proportional to the forth power of the distance (e_{amp4}). The former is used for the communication between a cluster member and a clusterhead and the latter for the communication between a clusterhead and the sink, since the distance in this case is much greater. Typical values for these parameters can be found in Table 2 [16].

The energy spent when a sensor i transmits a packet of k bits to its clusterhead CH_i that is located at distance d_{i,CH_i} is calculated by the following equation:

$$E_{i,CH_i} = k \cdot \left(e_{elec} + e_{amp2} \cdot d_{i,CH_i}^2\right). \tag{9}$$

The receiver, on the other hand, consumes energy when receiving a data packet of k bits, equal to

$$E_{rx} = k \cdot e_{elec}. \tag{10}$$

The clusterhead is usually responsible for aggregating the data it receives and compressing them in order to minimize the required packet length that it will transmit to the sink. This is in accordance to the selfishness of the players in the CG described earlier, as it saves energy in this way. The energy spent by the clusterhead for aggregating N_u packets of the same length k is computed using the following expression

Table 2 Typical values for the energy expenditure parameters

e_{elec}	$50\ nJ/bit$
e_{amp2}	$10\ pJ/bit/m^2$
e_{amp4}	$0.0013\ pJ/bit/m^4$
e_{fuse}	$5\ nJ/bit/message$

$$E_{aggr} = N_u \cdot k \cdot e_{fuse}. \tag{11}$$

For the case of communication between a clusterhead and the sink, the energy expenditure of the clusterhead is derived from the following formula:

$$E_{CH,Sink} = k \cdot \left(e_{elec} + e_{amp4} \cdot d_{CH,Sink}^4\right). \tag{12}$$

Let us now discuss on these costs. When a sensor sends data to a clusterhead, the consumed energy E_{i,CH_i} depends on their distance $d_{i,CH}$, for a fixed packet length. We denote this energy consumption by δ_i to refer to this cost. Since the clusterheads may vary with time and due to the random distances between the sensor and the clusterhead, we can calculate an expected value for δ_i. As there is no reason to distinguish between the players (expecting that at least in a long-term run they all will have the same expected values), the parameter δ_i can be considered identical for all players and constant.

The other parameter that needs to be determined is the cost c for the CH due to aggregation of the collected data and transmitting them to the sink. There are a number of ways that sink can be reached: The CH can send the data directly to the sink, it could use intermediate nodes to reach it, or another clustering hierarchy could be used. Regardless of the method used, the parameter c will depend on the size of data, the number of packets received and the distance between the node and the node that will receive them, either this is the sink itself or another node in the network.

Nevertheless, as a first approximation, we may assume that the average degree (or average number of neighbors N_u) is the same for all nodes, for a given number of total nodes[1]. Hence, the expected number of packets received by a CH can be considered constant and thus the energy spent for reception E_{rx} and aggregation of packets E_{aggr} is the same for all clusterheads. Furthermore, we assume that the distance between the clusterhead and the sink is approximately the same for all clusterheads. This is true if the sink is located outside the region covered by the sensors and at a far distance or if the sink is mobile and it changes positions so that in long-term its average distance from every node is the same for all nodes. Consequently, the distance between an arbitrary clusterhead and the sink $d_{CH,Sink}$ is assumed constant and thus the energy expenditure $E_{CH,Sink}$ can be con-

[1] We will remove this assumption later in the paper

sidered constant too. This means that

$$c = N_u \cdot E_{rx} + E_{aggr} + E_{CH,Sink} > E_{i,CH_i} = \delta. \quad (13)$$

Based on the previous assumptions, we may recalculate the probability of a node declaring itself as clusterhead in the clustering game with N nodes, taking into account that the benefit when a node plays ND while at least one other plays D is $v-\delta$ instead of v:

$$p = 1 - \left(\frac{c-\delta}{v-\delta}\right)^{1/(N-1)} \Longrightarrow$$
$$p = 1 - \omega^{1/(N-1)}, \quad (14)$$

where we use the letter ω to represent the ratio $(c-\delta)/(v-\delta)$ and $0 < \omega < 1$. Thus, we have derived a probability that corresponds to a symmetrical mixed strategy Nash Equilibrium for the one-stage clustering game. So, for a set of nodes that can hear each other, there is a natural incentive to cooperate and form a cluster when every node becomes clusterhead with probability p. This result is very important, since it proves that there is no need to use a cooperation enforcement mechanism, since cooperation arises naturally from the rules of the game.

3.4 Expected Payoffs and Price of Anarchy

In this section we compare the expected payoffs corresponding to the Nash Equilibrium probability to the maximum possible expected payoff in the framework of the game and also against the global optimum. The latter will permit us calculate the Price of Anarchy.

The average payoff of an arbitrary node i is given by:

$$\overline{P} = (v-c)\cdot Pr\{s_i = D\} + v\cdot Pr\{s_i = ND \bigcap \exists j \text{ s.t. } s_j = D, j \neq i\} \quad (15)$$
$$= (v-c)\cdot Pr\{s_i = D\} + v\cdot Pr\{s_i = ND\}\cdot Pr\{\exists j \text{ s.t. } s_j = D, j \neq i\} \quad (16)$$
$$= (v-c)\cdot p + v\cdot(1-p)\cdot(1 - Pr\{s_j = ND, \forall j \in \mathbf{N}, j \neq i\}) \quad (17)$$
$$= (v-c)p + v(1-p)(1-(1-p)^{N-1}) \Longrightarrow$$
$$\overline{P} = v - cp - v(1-p)^N. \quad (18)$$

Substituting in Eq. (18) the equilibrium probability p^*, the average payoff of the equilibrium strategy $\overline{P_{NE}}$ is

$$\overline{P_{NE}} = v - cp^* - v(1-p^*)^N =$$
$$= v - c\left(1 - \left(\frac{c}{v}\right)^{1/(N-1)}\right) - v\left(1 - \left(1 - \left(\frac{c}{v}\right)^{1/(N-1)}\right)\right)^N = \quad (19)$$
$$= v - c + c\left(\frac{c}{v}\right)^{1/(N-1)} - v\left(\frac{c}{v}\right)^{N/(N-1)} \Longrightarrow$$
$$\overline{P_{NE}} = v - c. \quad (20)$$

However, this is not the maximum average payoff the users may gain. We can easily compute the probability p' that maximizes $\overline{P}$ by setting the derivative of $\overline{P}$ equal to zero. The corresponding probability p' is equal to:

$$p' = 1 - \left(\frac{c}{Nv}\right)^{1/(N-1)} \quad (21)$$

and the maximum average payoff $\overline{P_{max}}$ is then:

$$\overline{P_{max}} = v - cp' - v(1-p')^N =$$
$$= v - c\left(1 - \left(\frac{c}{Nv}\right)^{1/(N-1)}\right) - v\left(1 - \left(1 - \left(\frac{c}{Nv}\right)^{1/(N-1)}\right)\right)^N = \quad (22)$$
$$= v - c + c\left(\frac{c}{Nv}\right)^{1/(N-1)} - v\left(\frac{c}{Nv}\right)^{N/(N-1)} \Longrightarrow$$
$$\overline{P_{max}} = v - c + c\left(1 - \frac{1}{N}\right)\left(\frac{c}{Nv}\right)^{1/N-1}. \quad (23)$$

Nevertheless, the maximum average payoff we have computed is probably not the optimum solution. The latter corresponds to the case where each time just one player plays D and all the others play ND. Since we do not differentiate the players, the optimum solution (within a period of N rounds) corresponds to the case where each player plays D in just only one round, and no other player does so. In this optimum case the average payoff is

$$\overline{P_{opt}} = \frac{(N-1)v + (v-c)}{N} = \frac{Nv - c}{N} \Longrightarrow$$
$$\overline{P_{opt}} = v - \frac{c}{N}. \quad (24)$$

It is interesting to observe that $\overline{P_{opt}}$ is not equal to the $\overline{P_{max}}$, which means that probability p' that optimizes the expected payoff of a user playing the game is suboptimal with respect to the global optimum setting.

Once we have computed the $\overline{P_{NE}}$ and $\overline{P_{opt}}$, it is easy to compute the price of anarchy (PoA), which corresponds to the payoff loss due to the suboptimum choice of the equilibrium probability p^*:

$$PoA = \overline{P_{opt}} - \overline{P_{NE}} =$$
$$= v - \frac{c}{N} - \left[v - c + c\left(\frac{c}{v}\right)^{1/(N-1)} - v\left(\frac{c}{v}\right)^{N/(N-1)}\right] \Longrightarrow$$
$$PoA = c - \frac{c}{N} = \frac{N-1}{N}c. \quad (25)$$

The conclusion of this analysis is that the PoA depends only on the cost level of playing D and not on the gain v of managing to join a cluster. The minimum value of PoA corresponds to a two-player game ($N = 2$). On the contrary, when the number of nodes tends to infinity, an upper bound is reached:

$$PoA_{min} = c/2 \quad (26)$$
$$PoA_{upper\_bound} = c. \quad (27)$$

As a consequence, the PoA is upper and lower bounded and its values are always within the interval $[c/2, c)$ regardless the number of nodes.

Let us now investigate the case where the cost c is not constant but it depends on the total number of players N and the number of players that play D at the same round of the game N_{CH}. For simplicity we assume that if the number of CH in a round is greater than one, the rest of the nodes are distributed evenly between the formed clusters. In specific, we adopt the following form of the cost function:

$$c = c(N, N_{CH}) = \frac{N}{N_{CH}} c_1 + c_0. \tag{28}$$

The above expression incorporates the basic facts the cost function is supposed to have. When the total number of players N increases, the cost increases too. On the other hand, when the number of players that play D increases, the cost decreases. In particular, the cost decreases with the increasing number of N_{CH} and the decrease is faster for small values of N_{CH}. The rationale behind this is that the cost of being a clusterhead is distributed between the declared clusterheads. Thus, when the clusterheads are small in number, their increase has greater impact in the decrease of the cost that every clusterhead should bear than in the case where the number of clusterheads is high, so an additional clusterhead would only slightly reduce the cost of every single clusterhead.

Before evaluating the new equilibrium probability, we need to find the expected cost, as it is not constant like before. In order to do that, we need the expected number of CHs. The probability the number of self-elected clusterheads in a round is equal to n is given by the binomial distribution:

$$P\{N_{CH} = n\} = \binom{N}{n} p^n (1-p)^{N-n}, \tag{29}$$

and its expected value is Np. Hence, the expected cost will be a function of the probability p:

$$c = c(p) = \frac{N}{Np} c_1 + c_0 = \frac{c_1}{p} + c_0. \tag{30}$$

Now, based on Eq. (3) we may recompute the new equilibrium probability as:

$$\begin{aligned} v - c(p) &= v \cdot \left(1 - (1-p)^{N-1}\right) \Longrightarrow \\ c(p) &= v(1-p)^{N-1} \Longrightarrow \\ \frac{c_1}{p} + c_0 &= v(1-p)^{N-1} \Longrightarrow \\ \frac{c_1}{v} + \frac{c_0}{v} p &= p(1-p)^{N-1}. \end{aligned} \tag{31}$$

The above equation has no trivial solution and it is not guaranteed that there is a probability $p \in [0,1]$ that satisfies it. Obviously, for $c_1 = 0$ the above equation can be easily solved and the solution is the probability of Eq. (4). If we assume that $c_0 = 0$, then Eq. (31) takes the following form:

$$\frac{c_1}{v} = p(1-p)^{N-1}. \tag{32}$$

whose solution is not guaranteed to exist. The maximum value of the right part of the above equation is maximized for $N = 2$ and the maximum is 0.25. Hence, if $c_1/v < 1/4$ then the gain of delivering the data is so high that the node prefers to play D, no matter what all the other players play. This threshold decreases, as the number of players increases, because of the extra cost induced by the increased number of players.

Let us now recompute the expected payoff based on Eq. (18), taking into account the dependence of the cost on the equilibrium probability.

$$\begin{aligned} \overline{P} &= v - c(p)p - v(1-p)^N = \\ &= v - \left(\frac{c_1}{p} + c_0\right)p - v(1-p)^N \Longrightarrow \\ \overline{P} &= v - c_1 - c_0 p - v(1-p)^N. \end{aligned} \tag{33}$$

Thus, the expected payoff for the equilibrium probability p^* is now:

$$\begin{aligned} \overline{P_{NE}} &= v - c_1 - c_0 p^* - v(1-p^*)^N \Longrightarrow \\ \overline{P_{NE}} &= v - \left(\frac{c_1}{p^*} + c_0\right) = v - c(p^*). \end{aligned} \tag{34}$$

The maximum expected payoff is achieved for probability p', which is computed as:

$$p' = 1 - \left(\frac{c_0}{Nv}\right)^{1/(N-1)} \tag{35}$$

and the maximum average payoff $\overline{P_{max}}$ is then:

$$\overline{P_{max}} = v - c_1 - c_0 p' - v(1-p')^N \Longrightarrow \tag{36}$$

$$\overline{P_{max}} = v - c_1 - c_0 - c_0 \left(1 - \frac{1}{N}\right) \left(\frac{c_0}{Nv}\right)^{1/N-1}. \tag{37}$$

4 A new clustering mechanism

In this section we will try to exploit the results of the previous analysis to design a simple clustering algorithm that will be based on the natural incentive for cooperation as analyzed before. Let us summarize first the results obtained previously. We have resulted into an expression of the equilibrium probability p a sensor is self-declared as clusterhead. This means that no sensor has any incentive to deviate from this probability. Since the parameters v, c and δ are known to every node, so does the parameter ω. So, the remaining problem is only the calculation of the total number of players, that is the total number of nodes participating in the clustering game. Following the assumption made for the LEACH protocol [9], we will assume that every sensor may hear the transmission from every other sensor. This is of course not very realistic, however it will permit us evaluate this simple method with respect to LEACH, which is a very popular clustering mechanism for sensor networks. It could also be considered as the first level of a clustering procedure with many levels of hierarchy. Let us mention here that if we need to be very strict, we have to examine if a node has an incentive not to declare its existence to its neighbors. Since a node is interested in using another node to send its packets on behalf of it and thus save energy, if it does not declare its existence then it will be unable to pass its data to a clusterhead and so it will have to send the data to sink by himself, which is undesirable. Therefore, all nodes will make their existence known to all others and since we assumed that there is no sensor out of range of any other, all nodes will compute the same number of players participating in the clustering game.

Using Eq. (14), a node computes the probability of becoming a CH at the first round of the clustering game. Due to this random procedure, some of the nodes will declare themselves clusterheads and send beacons, so that every other node will be able to select the closest CH. Then, the CH will collect the data from the members of their clusters and then the data aggregation is performed at each CH. Finally, every CH sends the aggregated data to the sink. The question that we have to answer know is what should be the probability p for the nodes on the second round. In order for the energy expenditure of the clusterhead role to be evenly distributed among the nodes, it is a good idea to set $p = 0$ for those nodes that have been clusterheads in the previous round. If $N_{CH}(1)$ is the number of clusterheads at the first round, then the game at the beginning of the second round should be played among $N_{play}(2) = N - N_{CH}(1)$ nodes. In general, at round $j+1$, the number of players playing the clustering game should be $N_{play}(j+1) = N - \sum_{k=1}^{j} N_{CH}(k)$. To achieve this, we set an following rule:

Zero Probability Rule (ZPR): Every node that has served as clusterhead sets the probability p to zero ($p = 0$) until all its neighbors have also served as clusterheads. Then, it switches back to the normal way of computing probability p, according to Eq. (14).

ZPR has a very interesting property: The nodes that have served as clusterheads have no reason to deviate since any probability $p > 0$ would result in lower payoff, as there is a positive probability that it is self-declared as CH again and thus consume more energy. What is more, the nodes that have not served as CHs yet, have no reason to deviate from ZPR either, since they know that the number of players have been reduced to $N_{play}(j+1)$ and thus the equilibrium probability is given by Eq. (14). When all nodes will have served as CHs, a "reset" takes place and all nodes initialize the number of players to N. If a node's energy have been depleted, though, the total number of players is reduced to adapt to this change.

By playing the game in rounds we actually define a repeated clustering game. In order to analyze this repeated game, we should use an appropriate definition of the utilities of the nodes and a discount factor to model the patience of the nodes in receiving their payoffs. Due to the ZPR, which bounds the number of the stages of the repeated game, and assuming that the sensors are too impatient, the previous analysis is still valid.

To sum up, we have defined a clustering mechanism, that we call *Clustered Routing Of Selfish Sensors* (CROSS), whose critical characteristic is the random rotation of the role of clusterhead for energy balancing reasons, based on the rationality and selfishness of the sensor nodes participating in the network. There is no assumption that the nodes will cooperate with each other to follow the rules of a traditional clustering protocol. What is more, there is no guarantee that after a specific number of rounds all nodes will have served as CHs for exactly one time. However, we expect that the randomness of the choices and the selfishness of the nodes will finally result in the desired performance.

5 Performance Evaluation

In order to evaluate CROSS, we run a number of simulations for several values of parameter ω. In an area 50mx50m we randomly placed $N = 100$ nodes that remained stationary throughout the simulation time. The sink was placed at position $(25, 125)$, thus it was located at least 75m from the closest node of the network. The parameters for energy consumption were those of Table 2 and the initial energy of all nodes was set to E_{init} = 0.5J, while the packet size had a fixed length of k=2000bits. The path loss exponent for short range transmissions (between sensors) was 2, while for long distance transmissions it was considered equal to 4. The range of every node was set to be large enough, so that any node could reach the sink in one hop. We compared CROSS to LEACH, based on the methodology described in [9], as LEACH is a very well-known algorithm, it is distributed and most of the works in this area use it as reference. What is more, in [16] it is mentioned that when the nodes are assumed to be able to reach any other node, original LEACH achieves increased network lifetime compared to both HEED and a generalized and energy-aware version of LEACH. For LEACH, the average number of clusters per round is set to the proposed by the authors value of 5% (thus 5 nodes per round for 100 nodes it total) and after 20 rounds all nodes should have served as clusterheads only once. At the beginning of each round each sensor selects individually the probability of declaring itself as clusterhead according to the algorithm followed. Then, the rest of the nodes select the closest clusterhead and join its cluster. After collecting the data packets from all its members, every clusterhead fuses them into a single packet that is sent to the sink directly. The procedure continues, even if a node has exhausted all its energy. In this case, the rest of the nodes follow the process ignoring that node. We used 6 different values for ω, namely 0.05, 0.1, 0.3, 0.5, 0.7 and 0.9. All metrics were averaged over 100 independent simulation runs, which was found to be a number large enough to guarantee statistically confident values.

The most important metric that reveals the performance of any clustering technique is the *network lifetime*. Here, we use the most common definition (although alternative definitions exist), i.e. the network lifetime is the lifespan of the node that first among all the others depletes its energy. We assume that a node's energy is exhausted if 99.9% of its initial energy has been consumed. The results for several values of ω are presented in Fig. 3. The curve for LEACH is straight because it is independent of the parameter ω and we used the same value repeatedly for illustration purposes. It is clear from this graph that CROSS outperforms LEACH in almost all cases. Only when $\omega = 0.05$, the lifetime achieved by LEACH is slightly higher than that achieved by CROSS. Another interesting observation is that the network lifetime under CROSS seems to achieve a maximum value at $\omega = 0.5$. For smaller or larger values, the lifetime is decreased. However, the reduction is not greater than 200 rounds, which is around 10% of the maximum network lifetime. A general conclusion is that the selfishness of the sensors seems to lead to a performance at least as good as LEACH, which is a protocol that requires cooperation between nodes and expects from each one of them to follow it without deviations.

In order to show the performance of the protocols in more detail, we provide in Fig. 4 a graph showing the number of nodes that are still alive with respect to the number of rounds. Since there are initially 100 nodes, this graph shows actually the percentage of alive nodes as time passes. We plotted the results for both LEACH and CROSS. For the latter, we selected only two representative values of the parameter ω, 0.1 and 0.9, in order to compare two extreme cases and the graph is still easily readable. The results confirm the previous ones. Although for the small value of ω the first node "dies" sooner than for the large value of ω, the opposite is true for the last node that spends all of its energy. Another interesting observations is the time instance when the curves of LEACH and CROSS intersect. It is the instance when under both protocols, the same number of "alive" nodes. For ω=0.1, this happens when approximately 70% of the nodes have exhausted their energy, while for ω=0.9 this happens when only 60% of them are still alive. In any case, the curves corresponding to CROSS and LEACH intersect after the time instance when half of the nodes have lost all their energy. This means that, if we consider as a critical point in the network's operation the point in time when half of the nodes consume all their energy, CROSS would still be a better choice than LEACH. This is why we argue that CROSS has a superior performance when compared to LEACH. As a final comment, the curve corresponding to ω=0.9 is much steeper than the one corresponding to ω=0.1. This could be explained by the smoothness of the curves referring to the probabilities p and P_A, as they were presented in Figs. 1 and 2. The more smooth the curve for the two probabilities (and especially that of p) the more smooth the decrease of the number of alive nodes as the time passes, in Fig. 4.

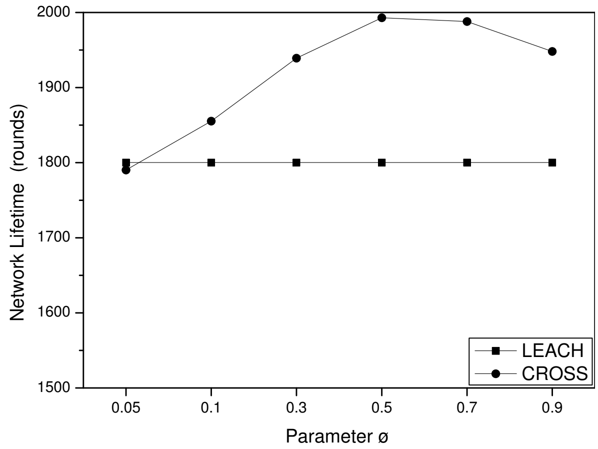

Fig. 3 Network Lifetime under CROSS for different values of parameter ω and comparison with LEACH.

Although the network lifetime metric is important, since it provides the knowledge of the instance where sensors begin to die due to loss of energy, the lifespan of the last node whose energy is depleted is of some importance too. We call this metric the *maximum node lifetime* and the results for both CROSS and LEACH are depicted in Fig. 5. Now, the performance of the two protocols has been inverted. LEACH achieves higher maximum node lifetime compared to CROSS, regardless of the value of ω used by the latter. This can be explained by the fact that, when the number of active nodes in the network decreases, the probability of a node declaring itself as clusterhead increases, as Eq. (14) shows. That being the case, as more and more nodes run out of energy, the remaining ones increase their probabilities and declare themselves as clusterheads more and more frequently. Hence, they begin to consume energy with an increasing rate, resulting in higher energy consumption rates. Fig. 1 indicates that the higher the ω the more smooth the curve is, thus we would expect that for higher values of ω, the maximum lifetime would be decreasing. This is in accordance with the results showed on Fig. 5, where the curve tends to decrease when ω increases.

Finally, we measured the average number of clusterheads per round in the case of both CROSS and LEACH. As expected, LEACH maintains this average at a value close to the targeted one, namely 5 clusterheads per round. The reason why the value is a bit lower than the targeted one is that we average the number of clusters over the maximum node lifetime (as described above) and not over the network lifetime. On the other hand, the number of clusterheads under CROSS strongly depends on the parameter ω. The higher the ω the less clusterheads per round. This behavior can be explained if we recall the dependence of the probability p

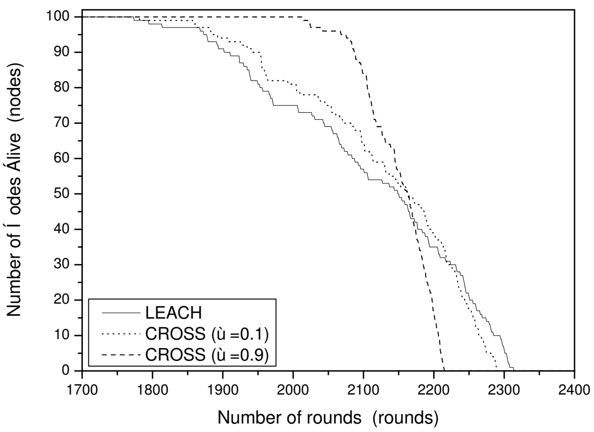

Fig. 4 Number of nodes with positive remaining energy under LEACH and CROSS for $\omega = 0.1$ and $\omega = 0.9$, versus the number of rounds.

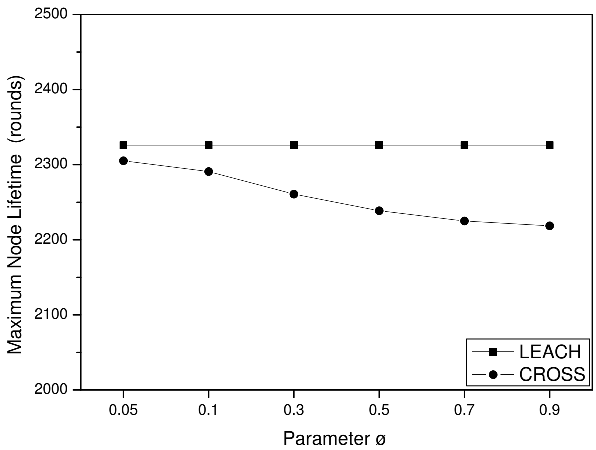

Fig. 5 Maximum Node Lifetime under CROSS for different values of parameter ω and comparison with LEACH.

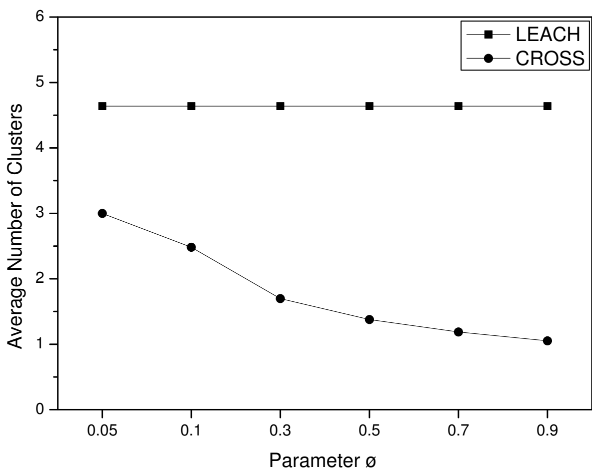

Fig. 6 Average Number of CHs per round under CROSS for different values of parameter ω and comparison with LEACH.

on the number of nodes and the value of ω. The clustering game we described admits an equilibrium when only a single node in the entire network declares itself as clusterhead and the calculated probabilities attempt to achieve exactly this objective. We can see that this objective is almost fulfilled for the majority of the values of the parameter ω. Only for small values of ω the average number of clusterheads per round in increased up to approximately 3 clusterheads per round. We remind here that the smaller the ω, the higher the probability of self-declaring, assuming the number of users constant. Hence, a higher number of clusterheads per round is expected. In game-theoretic terms, a low value for ω means that $v >> c$, in other words, the benefit of sending the data to sink is much greater in this case than the cost of being a clusterhead, and thus the node is prevented from risking not to send its data to the sink at all by not declaring. For this reason, the more the parameter ω tends to 1, the more a node risks by not declaring, resulting in reduced number of clusters per round. As a final comment, it seems that under this simulation settings, a total number of clusterheads for LEACH smaller that 5 would be more beneficial for the network lifetime. Indeed, simulations with a target value smaller that 5 showed an increase network lifetime. However, the performance achieved by LEACH in this case is still worse than that of CROSS for $\omega = 0.5$ and the nearby values. We need to mention here that the average number of clusterheads for CROSS changes from round to round, and strongly depends on the number of nodes that did not acted as CHs yet.

The purpose of the previous performance evaluation was to verify that a protocol as simple as CROSS, which uses the results of a theoretical analysis of selfish behavior can achieve a performance similar to algorithms specifically designed for a purpose (clustering a network). At the same time, the designed algorithm has one strong point with respect to the competition: it can be applied to networks where node behave selfishly, since it can ensure a robust network performance similar to that of networks where the entities are ”a priori” assumed to cooperate.

We were also interested in evaluating the performance of a CROSS algorithm following the equilibrium probability (referred as CROSS1) with respect to a CROSS algorithm that computes the probability that maximizes the expected payoff based on Eq. (21) (referred as CROSS2). Therefore, we run a number of simulations with the same parameters as above, except from the simulation area (100m x 100m), the sink position ((50,175)) and the initial energy (1J). The percentage of nodes declaring themselves as CHs under LEACH in every round was set to 5%, a value that results in a close to optimum behavior (unlike the previous case of the small network). Finally, for both CROSS1 and CROSS2 ω was fixed to 0.5, since CROSS was shown to achieve near optimal performance for this value.

First, we varied the number of nodes from 60 to 150, in order to observe the relation between LEACH, CROSS1 and CROSS2. The results for the network lifetime, which is the

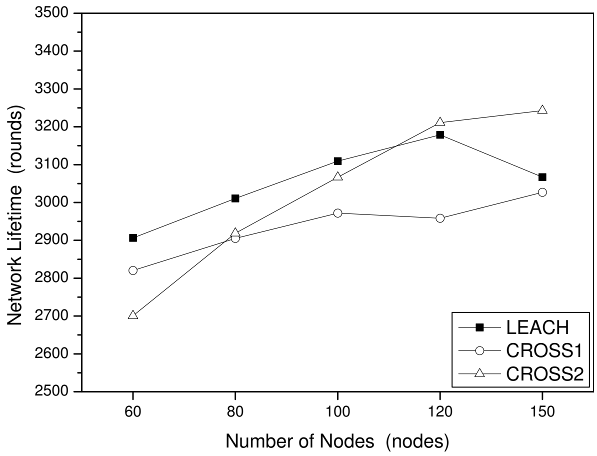

Fig. 7 Network Lifetime for LEACH, CROSS1 and CROSS2 for various number of nodes.

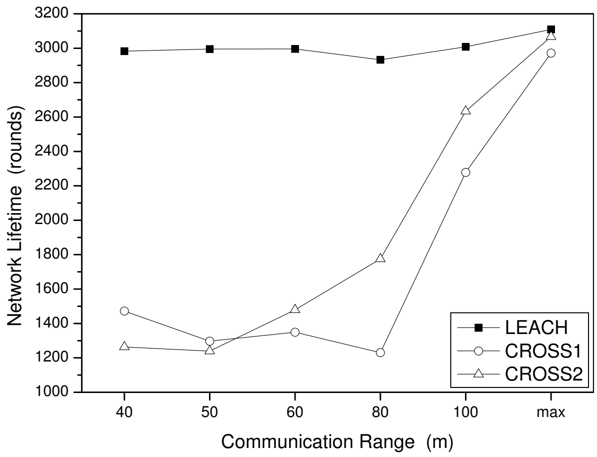

Fig. 8 Network Lifetime for LEACH, CROSS1 and CROSS2 for various radio ranges.

most critical metric in these networks, are shown in Fig. 7. As it was expected, CROSS2 achieves higher lifetime values than CROSS1, due to the more appropriate calculation of the probability of self-declaring. CROSS1 is also outperformed by LEACH. It is also interesting to observe that, for large number of nodes, CROSS2 seems to perform better than LEACH.

In order to investigate the performance of CROSS1 and CROSS2 for communication ranges less than the maximum, we simulated networks with lower ranges up to 40*m*. The results are quite interesting and are presented in Fig. 8. Neither CROSS1 nor CROSS2 are capable of maintaining the performance they achieve for maximum range. In particular, as the radius is reduced, their performance degrades, with CROSS2's curve being more smooth than CROSS1. The reason of this degradation is that the equilibrium probability is increased when the number of nodes decreases, which happens when the radius becomes smaller and the number of neighbors, resulting in higher energy expenditure. In general, when the number of nodes gets smaller, the probability of self-declaring becomes drastically higher. Thus, for small ranges or small number of nodes cross does not perform well, because the nodes' selfishness leads to high probability of self-declaration. When the number of nodes is high and the radius is large then this probability is reduced, leading to networks with small number of clusterheads and higher lifetimes.

6 Conclusions

In this work we provided a model appropriate to catch the basic rationality behind selfish nodes that form clusters in order to save energy. We have shown that the Nash Equilibrium of pure strategies corresponds to the case where only one node declares to be a clusterhead and all others restrain from declaring themselves. Although there is no symmetrical equilibrium in the pure strategies case, such an equilibrium exists for the mixed strategies case. We provide a formula for calculating the mixed strategy equilibrium probabilities taking into consideration some more particular characteristics of the clustering game, namely the energy consumption. The initial model was then extended so that it incorporates the dependency of the cost on the number of clusterheads and the efficiency of the Nash Equilibrium was evaluated by calculating the price of anarchy. We also showed that we may use the equilibrium probability in a real sensor network so as to evenly distribute the energy consumption among the sensors by randomly rotating the role of clusterhead among the sensor nodes. In comparison with LEACH, the proposed method achieves similar network lifetime values for most of the cases. In the future, we intend to investigate the iterative game, where the node maintains information about the history of the game and act according to the knowledge obtained, in case the nodes are not too impatient.

Acknowledgements This work was performed within the framework of the project PENED 2003 (Grant No. 636), co-financed by European Union-European Social Fund (75%) and the Greek Ministry of Development-General Secretariat for Research and Technology (25%).

References

1. Agah, A., Das, S.K., Basu, K.: A game theory based approach for security in wireless sensor networks. In: Proceedings of the IEEE International Conference on Performance, Computing and Communications (IPCCC 2004) (Phoenix, Arizona, April 15-17, 2004) (2004)
2. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: a survey. Computer Networks **38**(4), 393–422 (2002)
3. Altman, E., Boulogne, T., Azouzi, R.E., Jimenez, T., Wynter, L.: A survey on networking games in telecommunications. Computers and Operations Research **33**(2), 286–311 (2006)

4. Altman, E., Kherani, A.A., Michiardi, P., Molva, R.: Non-cooperative forwarding in ad-hoc networks. Lecture Notes on Computer Science **3462/2005**, 486–498 (2005)
5. Chatterjee, M., Das, S.K., Turgut, D.: WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks. Cluster Computing **5**(2), 193–204 (2002)
6. Felegyhazi, M., Hubaux, J.P., Buttyan, L.: Equilibrium analysis of packet forwarding strategies in wireless ad hoc networks - the dynamic case. Tech. rep. (2003)
7. Felegyhazi, M., Hubaux, J.P., Buttyan, L.: Nash equilibria of packet forwarding strategies in wireless ad hoc networks. "IEEE Trans. Mobile Comput. **5**(5), 463–476 (2006)
8. Heinzelman, W.B., Chandrakasan, A.P., Balakrishnan, H.: An Application-Specific Protocol Architecture for Wireless Microsensor Networks. IEEE Trans. on Wireless Commun. **1**(5), 660–670 (2002)
9. Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: Energy-efficient communication protocol for wireless microsensor networks. In: Proceedings of 33rd Annual Hawaii International Conference on System Sciences (HICSS) (Maui, Hawaii, USA, January 4-7, 2000) (2000)
10. Michiardi, P., Molva, R.: Analysis of cooperation strategies in mobile ad hoc networks with imperfect monitoring. Research Report RR-04-099, Institut Eurecom, Sophia-Antipolis (2004)
11. Muruganathan, S.D., Ma, D.C., Bhasin, R.I., Fapojuwo, A.O.: A Centralized Energy-Efficient Routing Protocol for Wireless Sensor Networks. IEEE Commun. Mag. **43**(3), S8–S13 (2005)
12. von Neumann, J., Morgenstern, O.: Theory of games and economic behavior. Princeton University Press (1944)
13. Nurmi, P.: Modelling routing in wireless ad hoc networks with dynamic bayesian games. In: in Proc. of 1st Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON 2004), pp. 63–70. Santa Clara, CA, USA (2004)
14. Osborne, M.J.: An Introduction to Game Theory. Oxford University Press (2004)
15. Srivastava, V., Neel, J., MacKenzie, A.B., Menon, R., DaSilva, L.A., Hicks, J.E., Reed, J.H., Gilles, R.P.: Using game theory to analyze wireless ad hoc networks. IEEE Commun. Surveys Tuts. **7**(4), 46–56 (2005)
16. Younis, O., Fahmy, S.: Distributed Clustering in Ad-hoc Sensor Networks: A Hybrid, Energy-Efficient Approach. IEEE Trans. on Mobile Comput. **3**(4), 366–379 (2004)
17. Zeng-wei, Z., Zhao-hui, W., Huai-zhong, L.: Clustering routing algorithm for game-theoretic techniques for wsns. In: Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS 2004) (Vancouver, Canada, May 23-26, 2004) (2004)