

Research Article

An Efficient Clustering Protocol for Wireless Sensor Networks Based on Localized Game Theoretical Approach

Dongfeng Xie, Qi Sun, Qianwei Zhou, Yunzhou Qiu, and Xiaobing Yuan

Wireless Sensor Network Laboratory, Shanghai Institute of Microsystem and Information Technology, 865 Chang Ning Road, Shanghai 200050, China

Correspondence should be addressed to Dongfeng Xie; dfhsie@gmail.com

Received 18 February 2013; Accepted 22 July 2013

Academic Editor: Rajgopal Kannan

Copyright © 2013 Dongfeng Xie et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Game theory has emerged as a brand new approach to model and analyse several problems of wireless sensor networks, such as routing, data collection, and topology control. Recently, a novel clustering mechanism called clustered routing for selfish sensors (CROSS) has been proposed based on game theory. The sensor nodes, which are modelled as players, join in a clustering game to campaign for cluster heads with an equilibrium probability. However, the CROSS algorithm needs the global information of how many nodes participate in the game at every round. Considering that this global way introduces much more packets exchange and energy consumption, we present a Localized game theoretical clustering algorithm (LGCA). In our protocol, each node selfishly plays a localized clustering game only with its neighbours within a communication radius R_c . Moreover, exactly one node can successfully bid for a position of the cluster head in one district, thus achieving an optimal payoff. Simulation results show that our method achieves a better result compared with CROSS and LEACH in terms of network lifetime.

1. Introduction

Recent developments in microelectromechanical system (MEMS) have led a fabrication of many low-cost, low-power, and multifunctional sensor nodes. Various kinds of these sensors are networked through wireless communication and deployed regularly or randomly in large numbers to a specific district. This kind of sensor network can be used to keep the battle field under surveillance, monitor ambient temperature, and so on [1]. Due to limited battery power, one of the challenges of wireless sensor network is to prolong the network lifetime. Thus, efficient energy management becomes one of the key consideration points for researchers.

Several protocols and algorithms have been investigated from different perspectives [2]. Among them, the clustering method that organizes sensor nodes into groups in an ad hoc way is the most traditional and influential in terms of energy dissipation. The entire network is divided into a few clusters, each consisting of a cluster head and several cluster members. The cluster head collects each member's data and transmits fused data to the base station. Since the base station is always farther than the head, a cluster member can save much more

energy on radio transmission. However, this is at the price of massive energy utilization of the cluster head because it consumes much more energy on receiving, fusing, and transmitting packets belonging to other members. Hence, it is better to regularly rotate the role of a cluster head to balance the energy load on every sensor node. Regarding clustering algorithms, the most important is how to effectively and fairly pick out the appropriate cluster heads while prolonging network lifetime as much as possible. This paper addresses how to choose cluster heads using a novel game theoretical approach.

Game theory is originally a mathematical method that describes the phenomenon of conflict and cooperation between intelligent rational decision makers. It is widely introduced and applied in wireless sensor networks in recent years [3]. The authors in [4] first utilize game theory to analyse the clustering problem and propose a clustering mechanism called CROSS. Each sensor node is modelled as a player, who can hear all other players' messages and know how many players there are. According to this number of players, every player calculates an equilibrium probability, which decides whether a player declares to be a cluster head.

Obviously, the global view in CROSS is not realistic and effective for two reasons: first, sensors usually have limited communication ability; second, a longer communication distance brings in much more interference. Hence, every sensor node, regarding to play a localized clustering game, can only hear messages from neighbours within its communication radius in our algorithm. Based on the number of players in a node's neighbourhood, it computes a probability and determines whether to compete for a cluster head. Compared with CROSS, our protocol is more practical and feasible. Simulation results also prove that our algorithm can effectively prolong the network lifetime.

The rest of the paper is organized as follows. Section 2 summarizes the related works on clustering problem. Section 3 gives an overview of CROSS, including a brief introduction and some interesting and important observations on CROSS. Section 4 describes our proposed algorithm in detail. In Section 5, simulation results between our algorithm and the other two algorithms are carefully compared and analysed. Finally, we conclude this paper in Section 6.

2. Related Works

Many clustering algorithms for wireless sensor networks have been proposed in recent years. Probably the most famous one is the low-energy adaptive clustering hierarchy (LEACH) algorithm [5]. The running time of LEACH is divided into numerous rounds. In every round, each node randomly produces a number between 0 and 1 and decides to be a cluster head (CH) if this number is less than a threshold. This threshold is elaborately designed so that a node can just be elected once in a certain time interval. Hence, all nodes nearly take turns to serve as CH, which achieves a uniform distribution of energy consumption among all sensors. Later, numerous algorithms are proposed to improve LEACH. Reference [6] designs LEACH-C in which the CH is no longer elected by node itself, but assigned by the base station which is capable of making the best decision via the global knowledge of network. The authors in [7] bring two improvements to LEACH. First, a novel energy-LEACH protocol gives more chances on nodes with more residual energy, thus preventing the whole network to die too quickly. Second, a multihop-LEACH protocol adopts the strategy of multihop communication among cluster heads instead of direct transmission between a CH and the base station (BS) because the latter wastes more energy on long-distance message delivering. However, these modified algorithms all depend on global information of the whole network such as node position (in LEACH-C) and residual energy (in energy-LAECH). Yet, in contrast, the expellant self-organization (ESO) algorithm is designed to replace the clustering formation procedure of LEACH [8]. ESO uses neighbour nodes' residual energy and connectivity as a metric to determine candidates, who further bid for a formal CH. Nevertheless, the information collection of localized neighbours is operated at the beginning of every round, which brings in a large amount of overheads.

In [9], the authors take several node attributes into consideration, such as node degree, transmission power,

mobility, and battery power. These parameters are weighed correspondingly and summarized to get a combined weight. The node with the smallest weight is chosen to be a CH. This weighted clustering algorithm (WCA) is completely distributed and dynamic, so that it quite adapts to the ever-changing topology. Similarly, [10] uses the weight of a node's k -density and residual energy as a metric to elect a CH, which is in the range of its 2-hop neighbourhood. Simulation results in this efficient cluster-based self-organization algorithm (ECSA) indicate a more evenly distribution of energy compared with LEACH and LEACH-C. Another algorithm, namely, DEC [11], also uses residual energy to elect CHs in a deterministic way, but outperforming than the probabilistic-based protocol LEACH.

Hybrid, energy-efficient distributed clustering approach (HEED) [12] is another well-known algorithm, in which CHs are selected by means of node residual energy and node degree. Moreover, at the beginning of clustering procedure, every node takes a number of iterations to decide whether to be a final CH by exchanging neighbouring messages in each repetitive step. This iterative scheme can effectively guarantee that the nodes with the least cost are chosen as final CHs.

In [13], an unequal cluster-based routing protocol (UCR) is put forward. Based on the observation that a CH close to BS consumes more energy on intracluster data forwarding, the authors group the nodes in proximity to BS into clusters of a rather small size, so that the corresponding CHs spend less energy on intercluster communication. In this way, no matter far or close, the power usage of an arbitrary CH is balanced. Consequently, the network lifetime is improved.

Besides, there are some other algorithms which concentrate on the clustering problem of heterogeneous wireless sensor networks. For example, (stable election protocol) SEP [14] uses a similar method of LEACH to elect CHs, with the effort of adding a few parameters to model different node types and initial energy. However, SEP solely takes two types of nodes into consideration while (energy-efficient cluster head election) EECHE [15] solves the clustering problem with three types of nodes.

In recent years, game theory is emerging as a powerful tool in the area of network formation. The authors in [16] first model the creation of internet-like networks as a game participated by selfish node agents. Without central control or coordination, these selfish nodes pay for their establishing links and benefit from short paths to destinations. This paper has shown some very interesting and heuristic results on distributed network designing. Afterwards, many people use the game theory to research the network formation problem [17–20]. Then in [21], the authors use game theory to construct a backbone in a wireless environment. By using the model of "Volunteer's Timing Dilemma", every node selfishly takes the strategy of "when to volunteer". In order to obtain a maximum expected utility, each node has a dominant strategy to volunteer. Implementations on several laptops, connected by 802.11g wireless cards, show that a wireless backbone can quickly be formed. However, previous works have assumed that every selfish node in a network creation game can communicate with all others, while in [22], the authors restrict every peer to a certain number

of neighbours, thus providing a selfish neighbour selection (SNS) game. Nevertheless, different from those of Internet-like wired or wireless networks, in wireless sensor networks, clustering mechanism is an effective method to construct a network topology. Therefore, CROSS [4] introduces a game theoretical approach on clustering. A detailed description is shown in the next section. In addition, literature [23] proposes a repeated game theoretical approach for clustering in mobile ad hoc networks. Each node is assumed to have a weight value composed of residual energy, average moving speed, and connectivity with neighbours. The node with the lowest weight value is chosen to be the CH. In order to prevent a node from playing the “dishonest” strategy and reporting a deceitful weight value, the authors introduce a repeated game model with limited punishment mechanism. By doing this, all nodes will act honestly and further guarantee the correctness and fairness of CH election. Besides, [24] investigates hedonic clustering game from two aspects of fixed clustering and correlation clustering. The most important contribution of this paper is the comprehensive theoretical analysis of clustering game using a mathematical method.

This paper also deals with the clustering problem by means of a game theoretical approach. What is more, most of our work is based on CROSS, and most of our simulation results are compared with it. So next, an overview of CROSS is necessarily presented.

3. Overview of CROSS

3.1. A Brief Introduction to CROSS. This subsection reviews the mechanism of CROSS [4]. In CROSS, the cluster head selection problem is modelled as a clustering game defined by $CG = \langle \mathbf{N}, \mathbf{S}, \mathbf{U} \rangle$, where \mathbf{N} is the set of players, $\mathbf{S} = \{S_i\}$ is the available strategies, and $\mathbf{U} = \{U_i\}$ is the set of utility functions of sensor nodes. Defining D as the strategy “declare myself as CH” and ND as the strategy “do not declare myself as CH”, then the strategy space is $\mathbf{S} = \{D, ND\}$. Considering payoffs, if none of the sensors becomes a CH, the payoff of every node is zero as packets are unable to transmit to the BS. If a node’s packets can be delivered by at least one other CH, its payoff is v . If a node declares itself as CH, then a cost c will be reduced from the gain v as more energy is consumed on directly transmitting packets to the BS. Assuming that only two sensors A and B join the game, then Table 1 gives the payoffs for all strategies. From this table, it can be seen that the best case is that only one player declares as CH and the other transmits packets by way of this CH.

Considering N players participate in the clustering game, according to Table 1, the utility function of player i can be described as:

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

Let us denote the probability of a player to choose strategy D as p and the probability to choose ND as $q = 1 - p$. According to Theorem 1 in [4], there exists an equilibrium

TABLE 1: The payoffs for two players clustering game.

	Player A	
	Declare	Not declare
Player B		
Declare	$(v - c, v - c)$	$(v - c, v)$
Not Declare	$(v, v - c)$	$(0, 0)$

probability p based on a symmetric mixed strategies Nash Equilibrium. Formula (2) shows the expression of p .

$$p = 1 - \left(\frac{c}{v}\right)^{1/(N-1)}. \quad (2)$$

Denoting $w = c/v < 1$, then

$$p = 1 - w^{1/(N-1)}. \quad (3)$$

The expected average payoff of the equilibrium strategy for an arbitrary node i is given by

$$\begin{aligned} \overline{P}_{NE} &= (v - c) \cdot \Pr\{s_i = D\} \\ &\quad + v \cdot \Pr\{s_i = ND \cap \exists j \text{ s.t. } s_j = D, j \neq i\} \\ &= v - cp - v(1 - p)^N \\ &= v - c. \end{aligned} \quad (4)$$

However, this average payoff is not the optimum value. The optimal case corresponds to the situation that only one player plays D and all the others play ND . In this case, the average payoff is

$$\overline{P}_{opt} = \frac{(N - 1)v + (v - c)}{N} = v - \frac{c}{N}. \quad (5)$$

By using the above game theoretical analysis, the authors in [4] propose the new clustering mechanism CROSS. They define a repeated clustering game in terms of rounds as the same in LEACH [5]. In the first round, N sensor nodes representing N players participate in the clustering game, and each of them has a probability calculated by formula (3) to compete for a CH. As a result, some of the nodes successfully declare themselves as CHs and form their corresponding clusters. In order to evenly distribute the energy consumption among all nodes, the players having served as CHs in the initial round better not join the game in the second round. Thus, the probability of these nodes is set to zero (zero probability rule). Now the number of players in the second round is $N_{PLAY} = N - N_{CH}$ (1), and the equilibrium probability p has to be recalculated by replacing N with N_{PLAY} in function (2). In general, nodes that are already to be CHs are excluded from the following rounds. Hence, CROSS requires that all nodes have the ability to hear transmissions from all others, so that every player knows which players have been CHs and how many players are left to take part in the next round game. Once all nodes have served as CHs, they reset to the initialization state, and the number of players returns to N .

3.2. *Our Observation on CROSS.* CROSS has shown a good paradigm of using game theory in clustering problem. However, it lacks deliberate considerations in many aspects. First, CROSS assumes that every sensor can hear the transmissions from all others so that it is aware of the nodes who are serving as CHs and further confirms the number of players in the next round. This global way is obviously not realistic, for every sensor node has a certain transmission distance due to limited power level. Moreover, a larger communication range wastes much more energy on radio transmission and brings in more interference. Hence, we will regard the clustering game in a localized way by introducing a parameter R_c . R_c represents the largest communication distance that a sensor can reach when it serves as a normal node. Generally, a node only communicates with other nodes within this radius. Therefore, to every node, the clustering game participants are itself and its neighbours. Once a node successfully competes for a CH, it immediately adjusts its power level until it is able to communicate with the BS.

The second problem comes from the equilibrium probability p . Seeing from formula (3), p is an exponential decaying function of N . However, CROSS supposes all nodes are simultaneously playing the game, which makes N too large. As a result, a sensor node has an extremely small probability p to become a CH. In fact, it is most likely that no one declares to be CH. This is an important case that CROSS does not take into consideration.

The probability that all N nodes choose not to be CHs is

$$P_{ND}(N) = q^N = (1 - p)^N = w^{N/(N-1)}. \quad (6)$$

Figure 1 gives a graphic illustration of function (6). As the number of players N increases, the probability P_{ND} increases accordingly. Moreover, P_{ND} is closely related to the parameter w . Figure 2 gives a more concrete statistical result on the number of cluster heads for different w when we run CROSS for 1000 rounds per w . Here, the total number of nodes is 100. This plot exposes at least two important problems which are thoroughly ignored by CROSS: first, the cluster number is rather uneven, ranging from 0 to 9; second, there exists a considerable portion of rounds that no one declares to be a CH. As to the former, [6] points out that the optimum number of clusters is 3 to 5 for a whole number of 100 nodes in a 100×100 topology. The latter case is particularly disadvantageous to energy efficiency since every node has to send packets directly to the BS rather than CHs. Even if these packets can be postponed until the next round or next several rounds, it still causes a long delay. In our protocol, we will address how to solve this troublesome problem.

The third problem exists in the observation that the equilibrium average payoff (4) is less than the optimal average payoff (5). Despite the difficulty that it is probably not able to achieve the optimal payoff by calculating an optimal probability p_{opt} instead of the equilibrium probability $p_{NE} = p$, it is still capable of hunting for the optimum by making use of the special characteristics of communication media, such as contention access and collision avoidance. A detailed description is presented in the next section for this issue.

Without doubt, CROSS has successfully applied game theory to the clustering problem of wireless sensor networks.

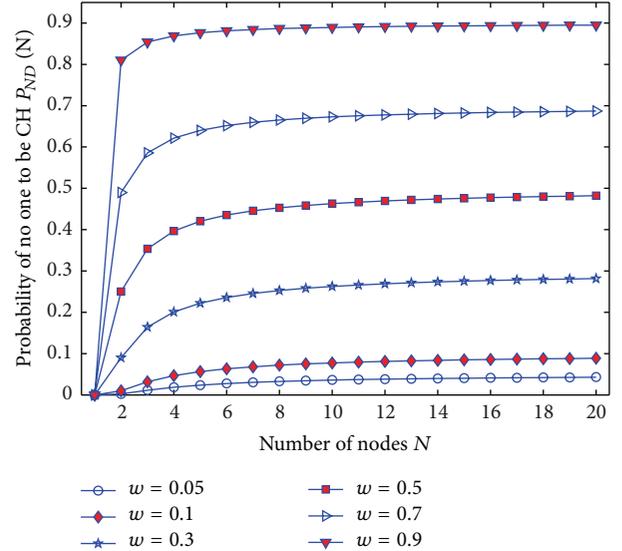


FIGURE 1: Probability P_{ND} of no sensors declare to be CHs versus the number of nodes for different w .

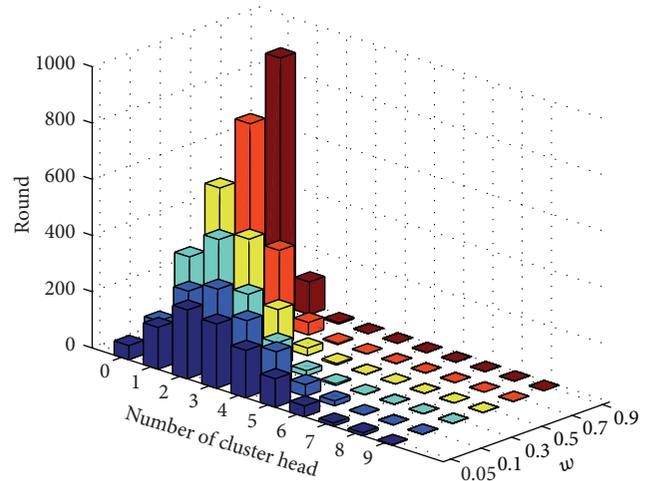


FIGURE 2: Round statistics on the number of CHs for different w . The total experimental round is 1000.

By calculating an equilibrium probability, each node has a certain chance to serve as a CH. Once the CHs are picked out, the other nodes will join in the CH with the closest proximity. By reselecting CHs in every round, CROSS has explored a more uniform energy distribution among all nodes and thus obviously extended the network lifespan compared to LEACH. Nevertheless, CROSS still leaves many unnoticed aspects to be carefully examined, which are the primary motivations for developing our protocol.

4. Our Proposed Protocol

Our proposed algorithm consists of an initialization phase to collect localized information and many repeated rounds, which is the same concept as in LEACH and CROSS. Each

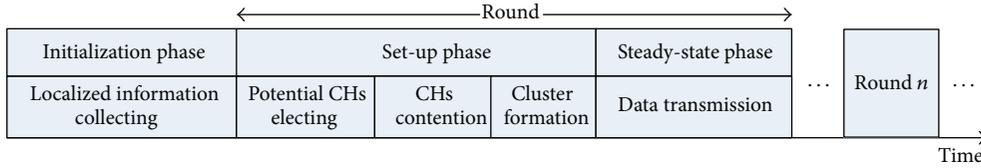


FIGURE 3: Time line showing the procedures of our proposed protocol.

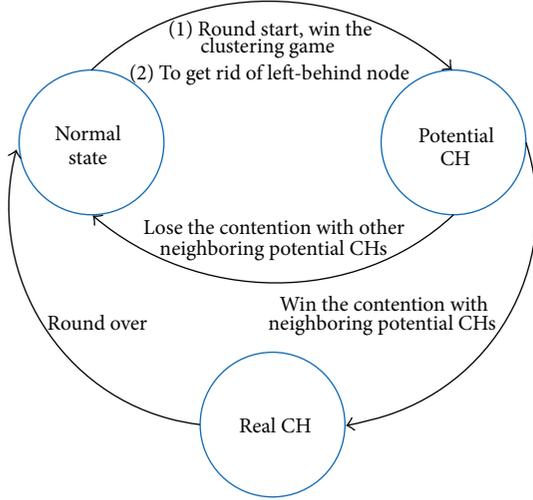


FIGURE 4: Node state transition diagram.

round is further divided into a set-up phase and a steady-state phase. The former is composed of three steps: potential CHs electing, real CHs contention, and cluster formation while the latter is responsible for data transmission among cluster member, CHs, and the BS. Figure 3 explicitly presents how our algorithm operates as time goes on.

Besides, in our model, every node has three states: normal state, potential CH state, and real CH state. Figure 4 is the diagram of node state transition. Initially, all nodes are in the normal state. A node who wins the clustering game temporarily switches to the potential CH state while a node finally serves as a real CH stays in the real CH state.

4.1. Initialization Phase. We suppose that the maximum power level is large enough for every node to transfer packets to the BS. In addition, it is presumed that every node can adjust its power level to adapt to a certain communication distance. However, for the sake of energy conservation, it is not necessary to hold every node at such a high power level like CROSS. For example, a node in the head state has to hold its power level at the highest level while in the normal state it can only keep its power level at a particular level P_c , which is just enough for a communication radius of R_c as shown in Figure 5. Paper [13] has a similar parameter of R_c , namely, competition range R_i .

The initialization phase is used to collect localized neighbour’s information for every node. When sensor nodes are deployed to a specific area, each of them is then broadcasting

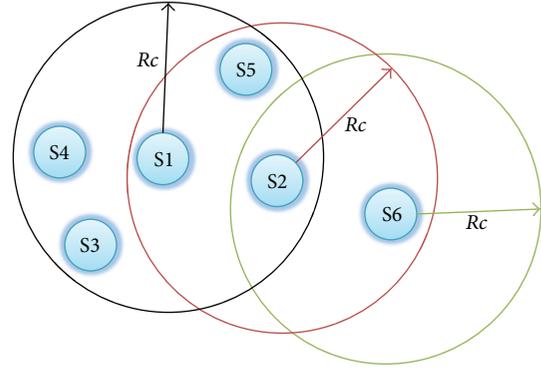


FIGURE 5: Sensor nodes only communicate with neighbours within radius R_c .

a “Hello” message to its neighbours. Meanwhile, it also knows how many neighbours are around by receiving “Hello” messages. We denote this value as $Nn(i)$ for node i . For example, in Figure 5, $Nn(1) = 4$ for node S1, $Nn(2) = 3$ for node S2 and $Nn(6) = 1$ for node S6. A “Hello” message probably not only indicates a node’s existence but also contains some other information such as position, sensor type and total energy. However, in our algorithm, we just need to be aware of neighbour’s existence so as to identify $Nn(i)$, which is a vital parameter for every next round. Once a new node joins, a “Hello” message is broadcasted, and every node reconfirms its Nn .

4.2. Set-Up Phase

4.2.1. Potential CHs Electing. This is the first procedure that a new round starts. As we have addressed, our algorithm runs in a localized way. Hence, the clustering game is also played locally. Since node i only knows that $Nn(i)$ neighbours are around, including itself, it selfishly believes that $Nn(i) + 1$ players are participating in the game. Thus, by replacing N to be $Nn(i) + 1$ in (3), the probability that node i declares to be CH is calculated by

$$p(i) = 1 - w^{1/Nn(i)}. \tag{7}$$

In the first round, each node egotistically determines whether to be a CH according to (7). If a node decides to be a CH, it temporarily stays in the potential CH state. A potential CH has to further compete for a real CH. In addition, those failed potential CHs must give up the opportunity to bid for real CHs and return to the normal state. If a node successfully competes to be an actual CH, a localized zero probability

rule (ZPR) [4] is applied. Its probability p is kept zero until all its neighbours have been real CHs. For a node that has not yet served as a real CH, it maintains a table that marks the already served neighbours. Suppose that the number of served neighbours for node i is $N_{\text{CH}}(i)$, then the number of qualified neighbours to play the next round clustering game is $N_{\text{cur}}(i) = Nn(i) - N_{\text{CH}}(i)$. Accordingly, node i has to update its potential CH electing probability to be

$$p(i) = 1 - w^{1/N_{\text{cur}}(i)}. \quad (8)$$

4.2.2. Real CHs Contention. As what we can see from the last paragraph, every node selfishly decides to be a CH, so it is most probable that more than two nodes in close proximity happen to be CHs. Actually this situation is quite common in CROSS, but we are unwilling to see it: for one reason, the CHs are likely to be unevenly distributed; for another, it is not the optimum payoff that we seek for. As what we have described in Section 3, the optimal case is that just one node declares as CH, and all the others choose to be cluster members. Thus, in order to achieve the optimum payoff in our algorithm, all potential CHs have to further contend for real CHs to ensure that only one real CH in a certain region.

In our scheme, the contention procedure is naturally carried out by utilizing CSMA/CA mechanism without any extra communication overhead. A potential CH who first fortunately contends for the physical media will immediately announce itself to be a real CH. Once hearing the announcement, the other potential CHs will abandon the declaration to be real CHs. Moreover, they all return to the normal state. As a result, there is only one real CH in a neighbourhood though several potential CHs have emerged. In addition, the real CHs are rather uniformly scattered in the whole district because every real CH is at least R_c meters away from the other. Figure 6(a) demonstrates how real CHs are geographically positioned on the whole topology. In a word, by adding the contention progress, the annoying phenomenon described in last paragraph is successfully eliminated from CROSS in our new protocol. Besides, in the neighbourhood of every real CH, it and its neighbours are best benefited in terms of the optimum payoff.

As we know, LEACH, SEP, and CROSS are all presuming that there are no restrictions on the communication distance of sensor nodes. This means that every node can reach another no matter how far it is away. However, as we restrict the communication radius of every normal node to be R_c , a node cannot be covered by another if their distance is more than R_c . Therefore, the “left-behind node” problem is posed to our protocol. A “left-behind node” is the node that always stays in the normal state and cannot join any clusters after the real CHs contention procedure. There are two reasons that cause this phenomenon. First, the competition probability p is still quite small though the number of players for every node is reduced. Therefore, it is quite possible that some nodes fail to become potential CHs. Second, these nodes are out of the range of other real CHs due to limited communication radius R_c . In reality, if a normal node waits such a long predefined time that no announcements are heard from real CHs, it is assumed

to become a left-behind node. As illustrated in Figure 6(b), outside of the two circles, three green sensor nodes are isolated from the other two clusters and become left-behind nodes. Regarding this problem, we propose two solutions to avoid the left-behind nodes directly communicating with the BS. First, all left-behind nodes immediately become potential CHs and bid for real CHs via MAC contention. Figure 6(c) is a presentation for this scheme. Second, all left-behind nodes raise their power level to communicate with the nearest CH and join the corresponding cluster as shown in Figure 6(d). The former solution is preferred as the new CH has no influence on existing CHs while the latter requires related CHs to improve the power level for a longer distance than the preset value R_c , therefore, bringing in more interference and energy expenditure.

4.2.3. Cluster Formation. When all real CHs have been determined, these heads will broadcast messages to announce their election. The remaining normal nodes pick out the nearest CH to join, and the clusters are then formed. Subsequently, every node waits for its corresponding CH to allocate a time slot. Once the slot is assigned, every normal node is needless to contend for media access.

4.3. Steady-State Phase. This phase is almost the same as described in [5, 6]. Every normal node collects its sensing data and transmits it to the CH in the designated time slot. After all intracluster packets have been received, the CH performs a data aggregation and sends a compressed packet to the BS. In such a way, sensor data streams from nodes to CHs and eventually to the BS again and again. After a predefined time, the current round comes to an end and the next round starts, accompanied by new CHs election and clusters formation.

As a summary, we name our protocol as (localized game theoretical clustering algorithm) LGCA. The most critical characteristic of LGCA is that sensor nodes are assumed to play a localized clustering game with their neighbours. According to the number of participating neighbours, every node gets an equilibrium probability to become a CH and form a cluster. Although some nodes may be left and fail to join any cluster, two solutions are proposed. First, these left-behind nodes are given another chance to bid for real CHs and form new clusters. We call this algorithm LGCA1. Second, the left-behind nodes increase their power level and join proper existing clusters. We call this algorithm LGCA2. Due to the solutions of “left-behind nodes”, the convergence of both LGCA1 and LGCA2 can always be guaranteed to make sure that a node either to be a real CH or to join a cluster. Finally, the flowchart of our LGCA family of algorithms is presented in Figure 7. Every node is treated equally and carries out the same algorithm. Despite the simplicity of this protocol, it is indeed effective according to our simulation results described in the next section.

5. Performance Evaluation

Our proposed algorithm is validated by a number of MATLAB simulations compared with CROSS and LEACH. Table 2 is a list of simulation parameters.

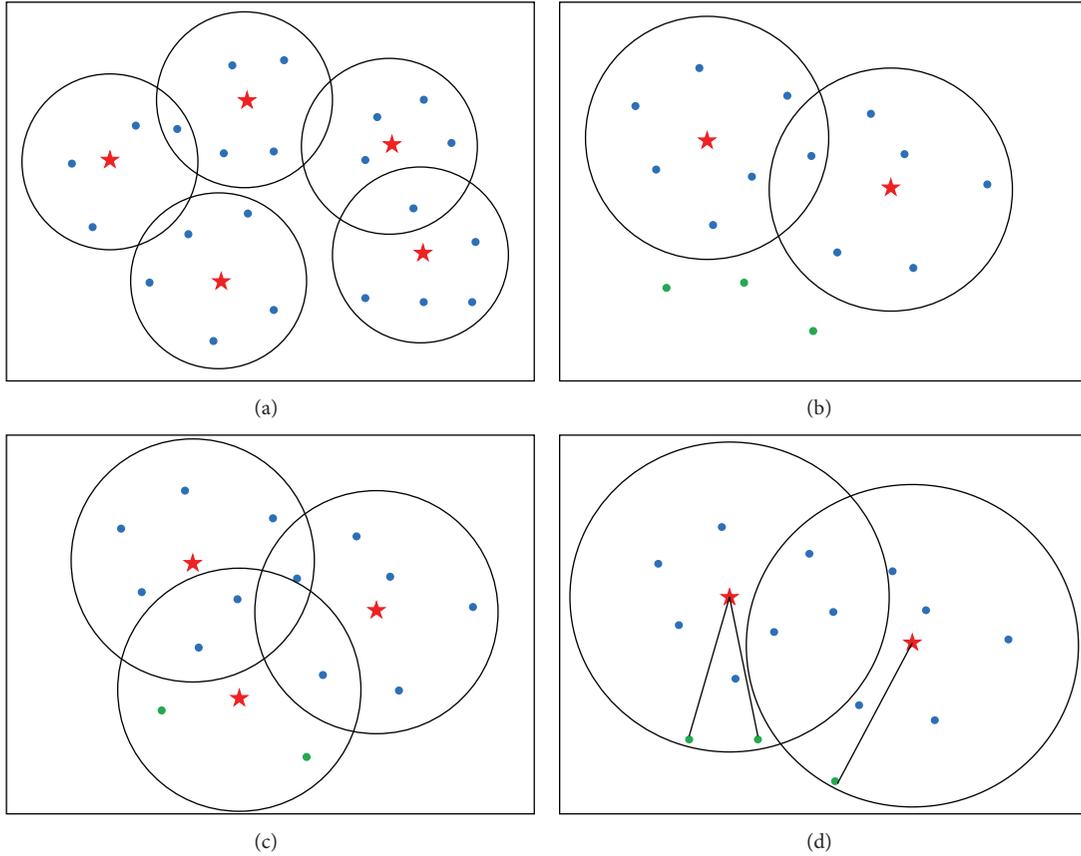


FIGURE 6: Real CHs election. A five-pointed star represents a real CH, and a filled circle represents a normal node. (a) Real CHs are regularly distributed. (b) The “left-behind node” problem. (c) and (d) Two solutions of the “left-behind node” problem.

TABLE 2: Simulation parameters.

Parameter	Value
Network size	$50 \times 50 \text{ m}^2$
Position of BS	(25, 125)
Number of nodes	100
Initial energy	0.5 joule
Packet size k	2000 bits
Predefined parameter w	Ranging from 0 to 1
Predefined parameter Rc	Ranging from 0 to 70 m

The parameters we used are almost the same as in CROSS except a new parameter Rc which is unique in our algorithm. Although w and Rc are theoretically ranging from zero to an upper limit continuously, we only select some representative values in our tests. For example, w is set to be 0.02, 0.05, 0.1, 0.3, 0.5, 0.7, and 0.9 while Rc is 5, 10, 20, 30, 40, 50, 60, and 70.

The radio energy dissipation model is assumed to be the same as in [6]. In transmitting state, the energy is supposed to be spent on radio electronics and power amplifier. Moreover, a threshold distance d_0 is introduced to determine when free space model and multipath model are used, respectively. Formulas (9) and (10) present how to calculate the energy

loss of transmitting. As to receiving, only radio electronics is required as shown in (11):

$$E_{TX}(k, d) = \begin{cases} kE_{\text{elec}} + k\epsilon_{\text{fs}}d^2 & d < d_0, \\ kE_{\text{elec}} + k\epsilon_{\text{mp}}d^4 & d \geq d_0, \end{cases} \quad (9)$$

$$d_0 = \sqrt{\frac{\epsilon_{\text{fs}}}{\epsilon_{\text{mp}}}}, \quad (10)$$

$$E_{RX}(k) = kE_{\text{elec}}. \quad (11)$$

The radio energy dissipation parameters are set as follows: $E_{\text{elec}} = 50 \text{ nJ/bit}$, and $\epsilon_{\text{fs}} = 10 \text{ pJ/bit/m}^2$, $\epsilon_{\text{mp}} = 0.0013 \text{ pJ/bit/m}^4$. The data aggregation energy for CHs is $E_{\text{DA}} = 5 \text{ nJ/bit/packet}$.

In order to eliminate accidental errors, every simulation for a specific parameter has been carried out 100 independent times, thus obtaining a reliable average result.

Network lifetime is a common evaluation criterion for clustering problem. Usually the network lifetime is defined as the lifespan of the node who first runs out of energy. Figure 8 is a network lifetime comparison among LEACH, CROSS, and LGCA families for seven different values of w , which are mentioned before. By the way, here $Rc = 50$. Table 3 gives the 95% confidence intervals of every result in Figure 8. Although there is some fluctuation of network lifetime as shown in

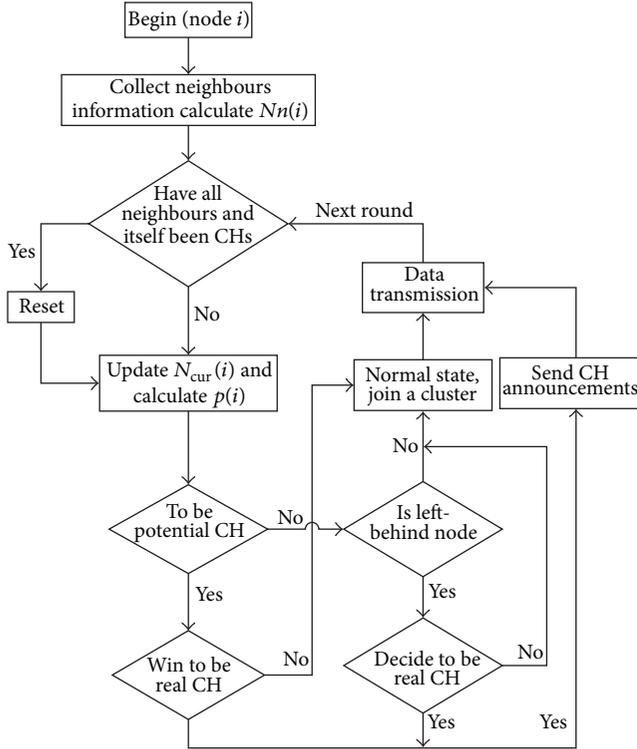


FIGURE 7: Flowchart of LGCA protocol.

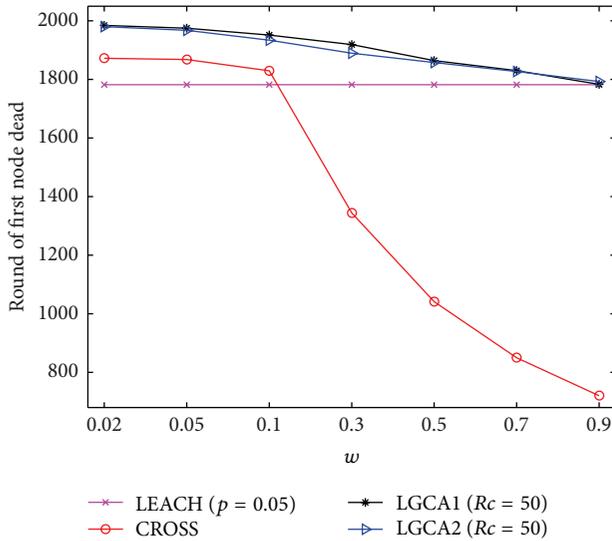
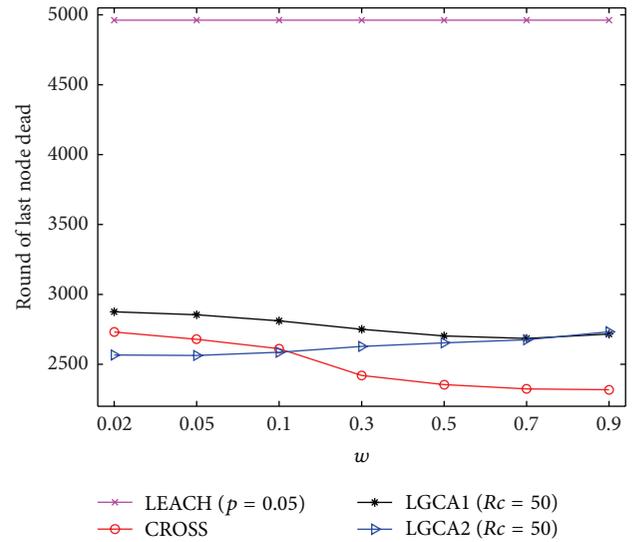
FIGURE 8: Network lifetime comparison among LEACH, CROSS, and LGCA families for different w .

Table 3, it is clear to see that both LGCA1 and LGCA2 outperform LEACH and CROSS to some extent in terms of average network lifetime. In addition, except LEACH which has no relationship with the parameter w , the other three curves all have a downward trend when w becomes larger. This is in accordance with previous expectations shown in Figure 1. The larger w becomes, the less probability nodes declare to be CH. When no nodes are willing to be CHs,

TABLE 3: Confidence intervals of corresponding network lifetime shown in Figure 8 for different w among 4 different algorithms.

w	LEACH	CROSS	LGCA1	LGCA2
0.02	(1671, 1893)	(1787, 1957)	(1870, 2099)	(1813, 2148)
0.05	(1671, 1893)	(1782, 1954)	(1863, 2086)	(1797, 2138)
0.1	(1671, 1893)	(1754, 1904)	(1836, 2067)	(1788, 2080)
0.3	(1671, 1893)	(1274, 1414)	(1789, 2048)	(1753, 2026)
0.5	(1671, 1893)	(987, 1096)	(1710, 2019)	(1715, 1999)
0.7	(1671, 1893)	(804, 895)	(1652, 2008)	(1664, 1991)
0.9	(1671, 1893)	(684, 755)	(1611, 1955)	(1628, 1957)

FIGURE 9: Maximum node lifetime of LEACH, CROSS, and LGCA families for different values of w .

sensors have to communicate directly with the BS, thus reducing the total lifetime. However, the curves of LGCA1 and LGCA2 are less sloping than CROSS. This is because the localized strategy in LGCA family reduces the number of players; besides, the mechanisms to deal with “left-behind nodes” avoid the risk of none declaration of CHs.

Figure 9 is a comparison of maximum node lifetime among LEACH, CROSS, LGCA1, and LGCA2 for different values of w . Figure 10 gives the corresponding number of nodes alive at every round. As we can see, LEACH achieves the best maximum node lifetime. In contrast with the round of first node dead in Figure 8, the energy consumption rate of LEACH is rather uneven. Figure 10 also verifies this. Obviously, LGCA family has the most uniform rate of energy expenditure. In addition, our LGCA protocol is more energy saving due to the fact that the maximum node lifespan of both LGCA1 and LGCA2 is longer than that of CROSS except LGCA2 when $w < 0.1$. This phenomenon can be explained as follows: when w is less than 0.1, there are very fewer clusters in LGCA2 compared with CROSS as shown in Figure 11. Thus, the left-behind nodes in LGCA2 have to communicate with rather distant CHs, resulting in a severe waste of energy. As w increases, the cluster number of LGCA2 keeps almost

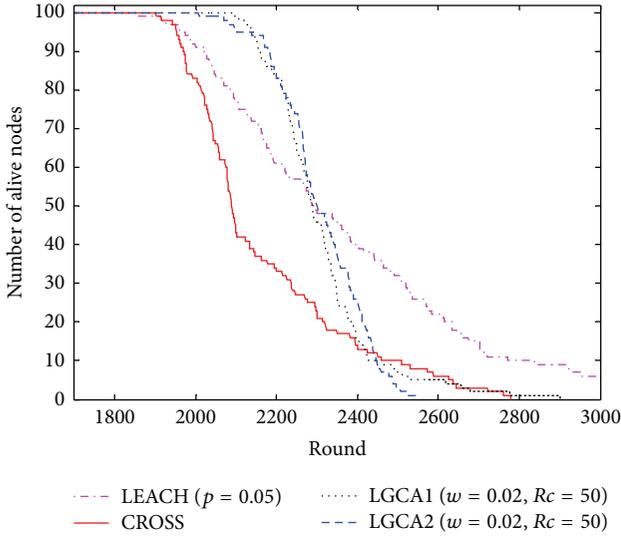


FIGURE 10: Number of alive nodes for LEACH, CROSS, and LGCA families.

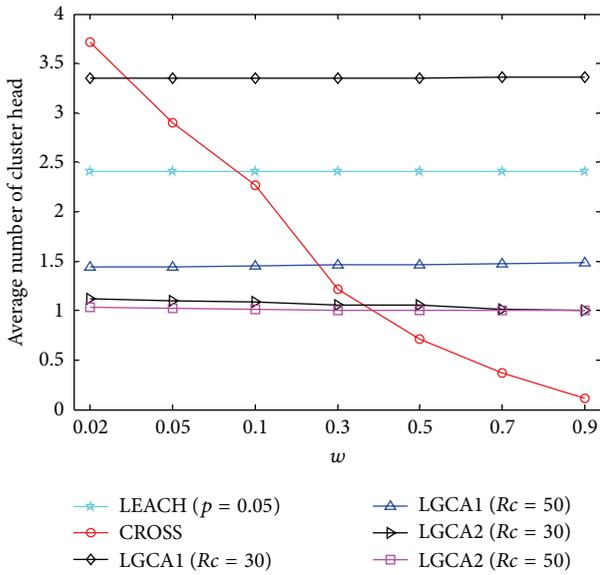


FIGURE 11: Average number of CHs among LEACH, CROSS, and LGCA families.

unchanged while CROSS decreases sharply. Hence, LGCA2 outperforms CROSS when $w \geq 0.3$ in terms of maximum node lifetime.

Figure 11 illustrates the differences of average number of CHs among LEACH, CROSS, and LGCA families. In accordance with previous analysis in Section 3, the cluster number, of CROSS sharply decreases as w grows. However, LEACH keeps a constant cluster number and LGCA family achieves a relatively consistent value with slight variations. This indicates that w has little influence on the cluster number of LGCA family. In addition, node coverage is extended following the increase of R_c , so the cluster number is naturally reduced. This is verified by the observation that the curves

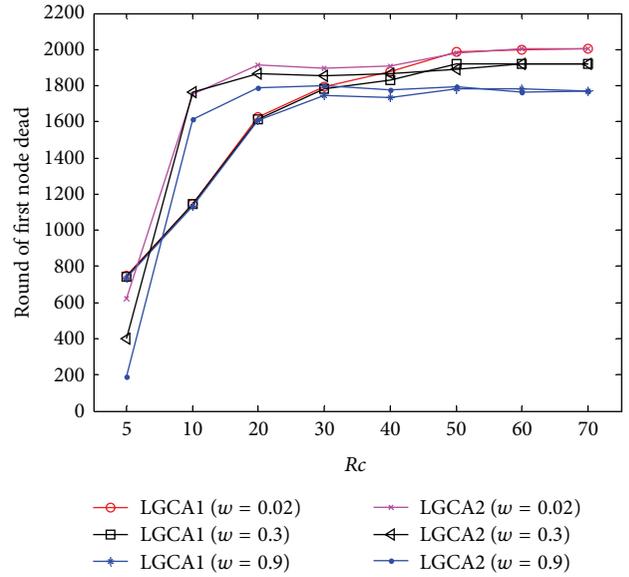


FIGURE 12: Network lifetime of LGCA1 and LGCA2 for three different values of w .

of $R_c = 30$ are on top of $R_c = 50$. However, compared with LGCA1, LGCA2 has less relationship with R_c , mainly because the left-behind nodes in LGCA2 have to join existing clusters rather than form new clusters.

R_c is the most important parameter in LGCA family. In order to evaluate its influence on the network lifetime, we provide Figure 12 to show how network lifetime changes with respect to different values of R_c . Basically, when R_c is small, the cluster size is small too. Thus, the clustering mechanism is not so effective on the energy conservation because most communication cost concentrates on the BS and CHs while intracluster interaction only makes up a small proportion. Therefore, the network lifetime is relatively short when R_c is less than 50 for both LGCA1 and LGCA2. However, when R_c reaches a certain value, due to limited topology size, the network lifetime remains almost unchanged. Figure 13 is the maximum node lifetime for LGCA1 and LGCA2. It is interesting to discover that when R_c is less than 20, LGCA2 performs better than LGCA1 on the maximum node lifetime. On the contrary, from Figure 12, LGCA2 performs poorly on the issue of node lifetime. In a word, both LGCA1 and LGCA2 are greatly related to R_c , thus a proper value of R_c is to be a priori selected according to the topology size.

The simulation results shown perviously demonstrate how effectively our designed protocol LGCA performs compared with CROSS and LEACH. Moreover, the dependency relationship to parameters R_c and w thoroughly evaluated. As long as they are optimally selected, LGCA can harvest an outstanding clustering result in terms of network lifetime.

6. Conclusion

On the basis of CROSS, we present a localized game theoretical approach to the clustering problem of wireless sensor networks. Every node and its neighbours are simulated to

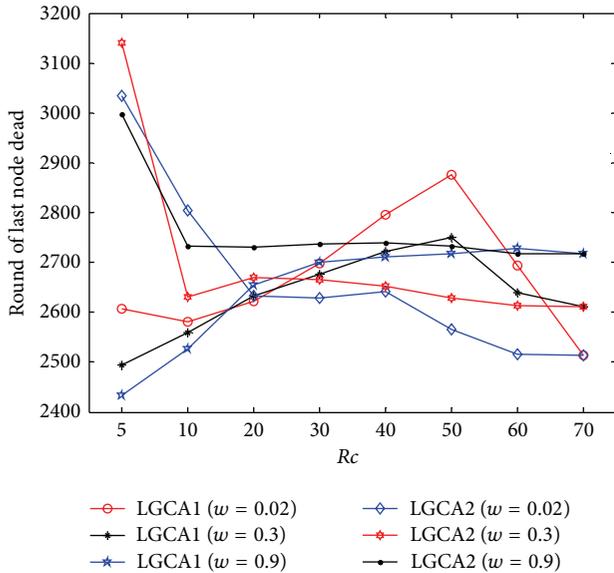


FIGURE 13: Maximum node lifetime for LGCA1 and LGCA2 for three different values of w .

play a clustering game with the purpose of selecting a cluster head. Though all nodes selfishly decide whether to be a potential CH, a real CH is restricted in a specific district by MAC contention. What is more, two solutions are suggested to handle the “left-behind node” problem. Simulation results reveal that LGCA outperforms CROSS and LEACH by properly selecting parameters. In the future, more node parameters such as residual energy and distance to the BS will be taken into consideration to ensure that the worthiest node wins the cluster game. In addition, the clustering problem of heterogeneous sensor networks is also concerned in our future plan.

References

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.
- [2] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, “Energy conservation in wireless sensor networks: a survey,” *Ad Hoc Networks*, vol. 7, no. 3, pp. 537–568, 2009.
- [3] H. Y. Shi, W. L. Wang, N. M. Kwok, and S. Y. Chen, “Game theory for wireless sensor networks: a survey,” *Sensors*, vol. 12, no. 7, pp. 9055–9097, 2012.
- [4] G. Koltsidas and F. N. Pavlidou, “A game theoretical approach to clustering of ad-hoc and sensor networks,” *Telecommunication Systems*, vol. 47, no. 1-2, pp. 81–93, 2011.
- [5] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences (HICSS-33)*, p. 223, January 2000.
- [6] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, “An application-specific protocol architecture for wireless microsensor networks,” *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.
- [7] F. Xiangning and S. Yulin, “Improvement on LEACH protocol of wireless sensor network,” in *Proceedings of the International Conference on Sensor Technologies and Applications (SensorComm ’07)*, pp. 260–264, Valencia, Spain, October 2007.
- [8] L. Zhao, X. Hong, and Q. Liang, “Energy-efficient self-organization for wireless sensor networks: a fully distributed approach,” in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM ’04)*, vol. 5, pp. 2728–2732, December 2004.
- [9] M. Chatterjee, S. K. Das, and D. Turgut, “WCA: a weighted clustering algorithm for mobile ad hoc networks,” *Cluster Computing*, vol. 5, no. 2, pp. 193–204, 2002.
- [10] M. Lehsaini, H. Guyennet, and M. Feham, “An efficient cluster-based self-organisation algorithm for wireless sensor networks,” *International Journal of Sensor Networks*, vol. 7, no. 1, pp. 85–94, 2010.
- [11] F. A. Aderohunmu, J. D. Deng, and M. K. Purvis, “A deterministic energy-efficient clustering protocol for wireless sensor networks,” in *Proceedings of the 7th International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP ’11)*, pp. 341–346, Adelaide, Australia, December 2011.
- [12] O. Younis and S. Fahmy, “HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks,” *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004.
- [13] G. Chen, C. Li, M. Ye, and J. Wu, “An unequal cluster-based routing protocol in wireless sensor networks,” *Wireless Networks*, vol. 15, no. 2, pp. 193–207, 2009.
- [14] G. Smaragdakis, I. Matta, and A. Bestavros, *SEP: A Stable Election Protocol for Clustered Heterogeneous Wireless Sensor Networks*, Boston University, Computer Science Department, 2004.
- [15] D. Kumar, T. C. Aseri, and R. B. Patel, “EECHE: energy-efficient cluster head election protocol for heterogeneous wireless sensor networks,” in *Proceedings of the International Conference on Advances in Computing, Communication and Control (ICAC3 ’09)*, pp. 75–80, January 2009.
- [16] A. Fabrikant, A. Luthra, E. Maneva, C. H. Papadimitriou, and S. Shenker, “On a network creation game,” in *Proceedings of the 22nd Annual ACM Symposium on Principles of Distributed Computing (PODC ’03)*, pp. 347–351, Boston, Mass, USA, July 2003.
- [17] E. Anshelevich, A. Dasgupta, J. Kleinberg, E. Tardos, T. Wexler, and T. Roughgarden, “The price of stability for network design with fair cost allocation,” in *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS ’04)*, pp. 295–304, October 2004.
- [18] J. Corbo and D. Parkes, “The price of selfish behavior in bilateral network formation,” in *Proceedings of the 24th Annual ACM Symposium on Principles of Distributed Computing (PODC ’05)*, pp. 98–107, Las Vegas, Nev, USA, July 2005.
- [19] S. Albers, S. Eilts, E. Even-Dar, Y. Mansour, and L. Roditty, “On Nash equilibria for a network creation game,” in *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 89–98, Miami, Fla, USA, January 2006.
- [20] Y. Halevi and Y. Mansour, “A network creation game with nonuniform interests,” in *Internet and Network Economics*, X. Deng and F. Graham, Eds., vol. 4858 of *Lecture Notes in Computer Science*, pp. 287–292, Springer, Berlin, Germany, 2007.
- [21] S. Lee, D. Levin, V. Gopalakrishnan, and B. Bhattacharjee, “Backbone construction in selfish wireless networks,” *ACM*

SIGMETRICS Performance Evaluation Review, vol. 35, no. 1, pp. 121–132, 2007.

- [22] G. Smaragdakis, N. Laoutaris, V. Lekakis, A. Bestavros, J. W. Byers, and M. Roussopoulos, “Selfish overlay network creation and maintenance,” *IEEE/ACM Transactions on Networking*, vol. 19, no. 6, pp. 1624–1637, 2011.
- [23] L. Sun, H. Sun, B. Yang, and G. Xu, “A repeated game theoretical approach for clustering in mobile ad hoc networks,” in *Proceedings of the IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC '11)*, pp. 1–6, Xi’an, China, September 2011.
- [24] M. Feldman, L. Lewin-Eytan, and J. Naor, “Hedonic clustering games,” in *Proceedings of the 24th ACM symposium on Parallelism in Algorithms and Architectures*, pp. 267–276, Pittsburgh, Pa, USA, 2012.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

