Energy Management of Wireless Sensor Network Based on Modeling by Game Theory Approaches

Shahinaz M. Al-Tabbakh  
Email: drshainaz@gmail.com  
Computer Sciences and Applications Group,  
Physics Dept.,  
Faculty of girls  for Arts, Sciences and Education, Ain Shams University, Cairo, Egypt

Eman M. Elshahed  
Email: Eman.elshahed86@gmail.com  
Computer Sciences and Applications Group-  
Physics Dept.,  
Faculty of Girls  for Arts, Sciences and Education, Ain Shams University, Cairo, Egypt

Rabie A. Ramadan  
Email: rabie@rabieramadan.org  
Computer Engineering Dept.  
Faculty of Engineering.  
Cairo University, Giza, Egypt, on leave at Hail University, Hail, KSA.

Hayam.M. Elzahed  
Email: helzahed2000@yahoo.com  
Physics Dept.,  
Faculty of Girls  for Arts, Sciences and Education, Ain Shams University, Cairo, Egypt

Abstract—Energy constrained (WSNs) have been deployed widely for monitoring and surveillance purposes, so energy efficient protocols must be employed to prolong the network lifetime. Sensor node expends maximum energy in data communication. Minimizing the number of communications (exchanged messages) by eliminating redundant sensed data saves much amount of energy and extends the lifetime of overall sensor networks. The problem manipulated in this paper is that the sink capability is limited and can’t receive all messages sent from all sensors. Therefore, nodes can cooperate with each other to minimize the number of dropped message. Defining some nodes to send to the sink node at certain time allows other nodes to go to sleep. Certainly, this operation positively reduce the nodes’ consumed energy and increase the life time of the network under consideration. This work compares some of the game theory schemes which is Gur game and its three enhanced versions (AGur, PGur and APGur) with a novel algorithm based on Market Entry Game (MEG). This game is adapted for WSNs to decrease the rejected number of messages accordingly produces energy saving in the overall network energy.

Keywords- WSNs; Game Theory; Gur; AGur; PGur; APGur; MEG.

I. INTRODUCTION AND OVERVIEW

WSN is one of the networks that is used in many of real life applications. Some of these applications are critical such as military and health monitoring. The requirements of the WSNs change with the used application; hence, it is called application-specific [1][2]. However, such networks suffer from different limitations including limited memory and reduced capabilities. In addition, a sensor node has limited buffering spaces and it is battery operated as well. At the same time, a node has to be autonomous in terms of knowing its neighbors in ad hoc manner. Nevertheless, nodes might measure different environmental features such as temperature, sound, chemicals, intensity, vibration, pressure, motion, and pollutants, or the presence (absence) of certain objects [1]. One of the main concerns of WSNs is the energy consumption due to messages’ exchange through the network and the internal node processing. In fact, the communication is considered the main source of energy consumption compared to the energy required for processing [2][3]. Therefore, focusing on minimization to the number of exchanged messages through the network was the major concern of many research areas such as routing, clustering, security, and data fusion. For instance, the authors of [4][5][6][7] were designed some algorithms to handle this problem. Such as [5] present a novel approach for message-efficient clustering called Rapid, in which nodes allocate local growth budgets to neighbors. This approach significantly reduces the number of messages exchanged and allows the cluster to grow based on local decisions. In [7] the authors proposes a new optimization based data collection scheme called Cuckoo Based Particle Approach (CBPA), this technique helps in energy efficient data collection or fusion, by eliminating data redundancy and minimization of energy consumption.

Abd, Mehmmood [19] study the problem of unbalanced energy consumption in WSNs by designing load balancing geographical routing protocols. Two decentralized, scalable and stable routing protocols were proposed in this thesis: Game Theoretic Energy Balanced (GTEB) routing protocol for WSNs and three dimensional Game Theoretic Energy Balance (3D-GTEB) routing protocol for WSNs. Both protocols are built based on balancing energy consumption into region level and node level using different game theory in each level. In the first level, evolutionary game theory was used to balance the energy consumption in various packet forwarding sub-regions, while in the second level classical game theory was used to balance the energy
consumption in forwarding sub-region nodes. 3D-GTEB benefits from utilizing the third coordinate of nodes' locations to achieve better and accurate routing decision with low network overhead.

In [18], game theory based real time fault-tolerant routing protocol called GTRF has been proposed. GTRF consists of two parts: a VA model and a jumping transmission model. In the VA model, designing the auction process and some rules for nodes to obey in this work. The work also prove that GTRF can meet real-time requirements. The simulation results show that GTRF can not only efficiently balance the energy cost of the network, prolonging the lifetime, but also utilize fewer control packets in achieving real time transmission.

Although the previous work tries to minimize the consumed energy, few have considered the buffering limitations as well as the message handling capabilities of the node itself. Such limitations force the node to drop some of the incoming messages which greatly affect the number of exchanged messages through the network due to retransmission. Our focus, in this paper, is on the efficient adjustment to the transmission and receiving of each node based on its capabilities. The proposed algorithms stated utilize one of the game theories techniques used in Gur game. In addition, a new algorithm was proposed based on of a famous game named Market Entry Game (MEG) [16] inspired from electronic marketing commerce. This game is adapted to wireless sensor networks (WSNs) to decrease the rejected number of messages.

The Gur game algorithm is included in different studies such as in [8] and [8] to control the amount of messages that reach the sink node [10]. In addition, in [11][12], [13] the authors try to solve the problems of power balance and energy consumption issues using the original Gur game. However, the proposed work suffer from many problems including the long time the game takes to reach stability, once the algorithm stabilized, some nodes will continue sending and others stay in sleep mode which leads to energy unbalanced network, the performance of the game depends on the number of state machine uses which is not considered in the previous work. This papers tries to solve such problems by proposing three solutions which are Adaptive Gur, Periodic and Adaptive Periodic techniques. In addition, this paper is considered as a motivation to other researchers to look at the concepts of game theory to be utilized in the field of WSNs in which we are convinced that these theories could be beneficial in solving many of the WSNs problems.

A. Gur Game

Gur game is one of the simple games and it is based on unlimited number of players and a referee. None of players is aware of the others; of course, the referee can watch all of the players. Therefore, the Gur game is a centralized game which leads to a centralized algorithm. The referee frequently asks the players to vote yes or no and count their answers. A reward probability \( r = r(k) \) is produced by the referee as a function of the number \( k \) of players who voted yes in the last run. \( r(k) \) is assumed to be between 0 and 1 \((0 \leq r(k) \leq 1)\). A typical function is shown in Fig. 1 where the reward probability function has its maximum value at the desirable number of players who voted yes (i.e. \( k = 35 \)). Each player is then independently rewarded with probability \( r(k) \) or penalized with probability \( 1-r(k) \) based on its answer.

The idea behind Gur Game algorithm is based on biased random walks of finite-state automata. A set of states have been described by the automata with assigned meanings and a set of rules to determine switches from one state to another. Fig. 2 is a simple example of a finite-state automaton with memory size \( M = 3 \) of Gur Game algorithm. Each state has its own meaning. States -1, -2 and -3 represent sleep states, whereas states 1, 2 and 3 represent active states. Only one state is allowed for a player to be in and can transit to only its adjacent state. If the player in a positive state and voted yes, his transition will be to the right after receiving the referee reward. On the other hand, the player will transit to the left if he is punished by the referee. The other case is the player is in a negative state and voted no, his transition will be to the left if rewarded by the referee and to the right if he got punished. Therefore, the reward pushes the player to the edge states while the punishment pushes him to the center states. However, and unfortunately, based on our experiments, there is serious problem noticed when using Gur game in WSNs which is nodes’ energy unbalance where the game may reach to stability and only some nodes continue to send their data and others are neglected. Thus, we propose three modifications to the Gur game to fit WSNs requirements. The first modification considers the Adaptive Gur game, the second considers the Periodic Gur game and the third considers the Adaptive Periodic Gur game.
B. Market Entry Game (MEG)

The market entry game is a game with \( N \) players who must decide simultaneously and independently whether to enter a market or to stay out. If they choose to enter the market, then, their payoff depends on the number of players who also choose this option, but communication between players is strictly forbidden. The game begins with a public announcement of a positive integer \( C \), which is the capability of the market, where \( 1 \leq C \leq N \). In each round, each player \( i \) must decide whether to enter the market \( (s_i = 1) \) or stay out of it \( (s_i = 0) \), where each player enters with probability \( p \) and stays out with probability \( 1-p \), i.e. \( p = (C-1)/(N-1) \). Individual payoffs are determined according to the payoff function \( P \)[16].

\[
P_{pi} = \begin{cases} 
0, & \text{if } s_i = 0 \\
\frac{V}{K}, & \text{if } s_i = 1 \text{ and } K < C \\
\frac{V}{C} - \frac{K-C}{K}, & \text{if } s_i = 1 \text{ and } K \geq C \\
\end{cases}
\]

where \( V \) and \( F \) are constants and indicated as industry profit and a fixed loss respectively and \( K \) is the number of players enter the market.

The payoff function includes three cases: 1) in the first case, when the player decision is to stay out \( (s_i = 0) \) the resultant payoff is zero. 2) in the second case, when the player decision is to enter \( (s_i = 1) \) and the number of players entered \( (K) \) is less than the capability of the market \( (C) \), a constant industry profit \( V \) is split equally among the \( K \) players entered and the payoff to each of the players stay out \( (N-K) \) becomes zero. 3) when \( K \) is equal to or greater than \( C \), which is the third case, a random number \( r \) is generated for each player; then, if \( r \) less than a threshold value, the payoff is as in the second case, otherwise, the players entered suffer a loss of \( F \) and are rejected. The threshold value is set as the same as the probability \( p \) value. The MEG can be considered as one member of the large class of coordination games, characterized by having large numbers of Nash equilibria. However, unlike games of pure coordination, where players have an incentive for all to take the same action, here successful coordination involves different players taking different actions: some entered and some stayed out [17].

The paper is organized as follows: Section three presents the problem statement; section four proposes solution approaches; section five displays the simulation results with the evaluation for all techniques; finally, the paper concluded in section six.

II. PROBLEM STATEMENT

The problem of this paper is to maximize the lifetime of the network by allowing more sensors to be in the sleep mode while others are doing their job. At the same time, the number of sensors that need to be powered on have enough data to report to the sink node. Therefore, the QoS of the WSNs is defined as the optimum number of sensors sending information toward the sink node. It is assumed a centralized WSN with a single sink node. The sink node is supposed to reach every other node through broadcasting. In addition, the capability of the sink node is limited to a certain number of connections (messages) at the same time. In fact, this issue is ignored in most of the current research in WSNs. It is usually assumed to be highly capable node. Sensor nodes are deployed randomly in the field and they report their data to the sink node. Some of these nodes may have a common coverage areas. Therefore, such sensors are considered redundant and only one node could be enough to be waked up (active) nodes and other nodes could go to sleep. The problem is how to dynamically adjust the number of active nodes according to the limitation of the sink node. At the same time, using the optimum number of nodes to be active maximizes the data to be received by the sink node and reduces the number of retransmitted messages due to the dropped ones. Certainly, this saves nodes energy and increases the network lifetime. To solve such problem, Gur game and its modified versions are utilized as explained in the next sections.

III. SOLUTION APPROACHES

In this section, the revision of Gur game algorithm to fit WSNs is presented then the proposed techniques are explored starting by the adaptation of Gur game to the WSN. In addition, an adaptive Gur game is presented as well as a periodic algorithm is introduced for enhancing the QoS control. Nevertheless, a combination between the adaptive Gur game and periodic algorithm is investigated. Finally, applying the MEG to WSN for solving the same problem is proposed and investigated.

A. Gur Game

For the Gur game to adapt WSNs operation[8], each sensor node is considered as a player and the sink node play as the referee. Assume that there is a collection of \( m \) nodes from \( S_1 \) to \( S_m \) and one base station BS/Sink. The BS is able to communicate directly or through a multi-hop with all other nodes where nodes are not aware of each other. At the same time, the sink node is assumed limited in terms of number of messages that can be received concurrently. At each second, if a sensor node is in a positive numbered state it will be powered-up and it will send a data packet
containing its data to the base station; however, if it is in a negative numbered state it will be powered-down and it is simply “sleeps” but still receives messages including reward probability by the sink node after each turn. The sink node desires optimal QoS from the sensor network at each time t, but it does not know the total number of live sensors at time t. The sink node, as mentioned, wants the information to be uniformly distributed from all the sensors. Therefore, the definition of optimal QoS could be receiving an optimal number of packets at time t as well as a fair messages receiving from each node. A typical reward function used by the base station could be given as follows:

\[ r = 0.2 + 0.8e^r \]  
\[ v = -0.002(k_t - n)^2 \]

where \( k_t \) is the number of packets received at time \( t \) and \( n \) is the known optimum number of packets that the base station wants. At each time \( t \), the base station counts the number of packets \( k_t \) it has been received from the nodes. It, then, calculates the Gur reward probability \( r(k_t) \). Finally, it broadcasts this probability to all the nodes. Each node, in turn, independently rewards itself with probability \( r(k_t) \) and punishes itself with probability \( 1-r(k_t) \).

\[ \text{C. Periodic Gur Game (PGur)} \]

Unfortunately, based on our experiments, there is a problem noticed when using Gur game in WSNs which is nodes’ energy unbalance. This energy unbalance is due to when reaching the optimum level of the required QoS, it causes the active sensors to stay active till their energy depleted and other sensors will stay in their standby mode. Such case accelerates the ending of the network lifetime as well as leaving uncovered areas in the monitored field. One of the proposed solution to solve this problem in Gur game was inspired from shuffling idea proposed by the authors in [11]; the authors propose to periodically shuffle the active and sleep nodes. However, in our case, exchanging sensor states may cause the system to be unstable. Therefore, we believe that it will be more effective to reapply the Gur game periodically. In the later sections, this period time is computed through experiments and turned to be 1000 runs (turns).

\[ \text{D. Adaptive Periodic Gur Game (APGur)} \]

It has been noticed that the convergence time of the Gur Game algorithm takes too long time as it will be shown in the results section. Certainly, this long time causes sensor nodes to consume excessive power and wastes resources. To handle such situation and reduce the convergence time, the problem is investigated based on two cases as follows:

1. Assume that the number of packets \( k_t \) arrived to sink node is less than the optimum number \( n \). In other words, the number of active sensors is less than the number of standby sensors. The expected transition for sensors is from the sensor standby states to active states (with probability \( 1-r \)) to reach the desired number of packets quickly, instead of the opposite transition from active states to standby states (with probability \( 1-r \)).

2. In the second case, assume that the number of packets \( k_t \) arrived at the sink node is greater than the optimum \( n \). In other words, the number of active sensors is greater than the number of standby sensors. The expected transition for sensors is from the sensor active states to standby states (with probability \( 1-r \)), instead of the opposite transition from standby states to active states (with probability \( 1-r \)). In both cases, in Gur Game, when each sensor node receives a reward probability \( r \) from the sink node, it cannot differentiate between two cases due to the lack of information, and thus makes transition blindly. Consequently the convergence of the Gur Game algorithm takes too long time.

To enhance the convergence time of the Gur Game, the idea is to use unambiguous reward/punishment mechanism inspired from [14]. The mechanism is based on sending a bit besides the reward probability by the sink to the sensor nodes. This bit of information can denote to as the “sign” of the reward probability. If the sign of the reward probability is negative it indicates the first case; otherwise it is the second case, as mentioned before. When sensor nodes receive a negative probability, all standby nodes move to active state with probability \( 1-r \); while all active nodes will stay in active. Similarly, when receiving a positive probability, all active nodes move to standby state with probability \( 1-r \), while all standby nodes will stay in standby. When receiving reward probability \( r = 1 \), all nodes will reward themselves and stay in their current state, and the system will remain stable.

\[ \text{E. Market Entry Game(MEG)} \]

When the MEG is applied to a WSN, each sensor node is considered as a player and the base station (sink node) as the
store or the market that each player wants to enter. Assume that there is a collection of $N$ nodes from $S_i$ to $S_N$ and one base station/sink. The BS is able to communicate directly with all nodes where there are no connections between them. At the same time, the sink node is assumed limited in terms of number of messages that can be received concurrently. At each round, if a sensor node is entered (sent a message) ($s_i = 1$) and the number of sensors already in the market (already sending to the sink node) ($k$) is less than the desired number of messages or sink node capability ($C$), it will be active and it will send a data packet containing its data to the base station. If the number of sensors entered ($k$) is greater than or equal to the desired number of messages ($C$), the base station will receive messages from sensors that have a priority to send and reject the others. However, if the sensor stays out ($s_i = 0$), it will sleep and its payoff function became zero. With many rounds, it is expected that the network can be stabilized and only number of sensors equivalent to the sink node capability will be active at the same time. It is also expected that, nodes can efficiently schedule themselves to go sleep during their inactive periods. Moreover, nodes with redundant data can cooperate to let only one node to report according to the sink node capability; however, the problem of redundant data reporting is out of the scope of this research article and it is considered as possible extension.

IV. SIMULATION RESULTS

In this section, different sets of simulation experiments are conducted. The simulation environment assumes a monitoring area of 100m X 100m and 100 sensors are deployed randomly in this field. In addition, it is assumed that the base station capability changes from 35 to 60 sensors (messages at the same time). The reward function used for Gur game and its modifications by the sink node is considered as given in Equation (2). While the payoff function used for MEG is the same one presented in Equation (1). At the beginning, all sensor nodes are in power-down state (-1). Simple Radio Energy Dissipation Model is used and it is adopted from [15] with nodes initial energy = 0.5 j. The results of the following experiments are the average over 5000 periods of time, different network topologies, and settings. Moreover, sensors’ memory ($M$) is considered during the different sets of experiments; in our case, the memory size is given one of two values which are $M=1$ and $M=3$.

The performance metrics are:

1)Life time: it is the time taken by the network for the first node to die. The time of one trial = time of BS broadcasting to all nodes + time of sending packets to BS and is calculated as time of one trial = $2d/v$, where $d$ is the average distance between the BS and the sensors and $v$ is the speed of light (i.e. $v = 3*10^8$ m/s).

For instance the time of one trial in centralized network = 0.65 microsecond (μs).

2)QoS ratio: it is used to effectively evaluate the quality of control on the number of active sensor nodes in the network, and it is defined as:

$$QoS = \frac{T_{QoS}}{T_f}$$

where $T_f$ is the lifetime when the first node die; $T_{QoS}$ is the total number of trials until the first node die, in which the number of received packets $k$ is limited to between $n\eta$ and $n(2-\eta)$ where $n$ is the desired optimum number, i.e., $[n\eta] \leq k \leq [n(2-\eta)]$. For example, if $n=35$, and $\eta=90\%$, then we have $32 \leq k \leq 38$. That is, at each second, the QoS level of the network is satisfied if the number of active sensor nodes is controlled in the range from 32 to 38 [14].

3)Average residual energy: it is used to show the average energy saved per node in the network after the first node dies.

A. Gur Game and its Modifications Comparison with respect to lifetime:

This section contains a set of experiments to measure the lifetime of the WSN with the number of sensors. The lifetime, in this context, means the number of trails (iterations) until the first node dies. We tend to increase the number of active sensors along with the total number of sensors. The number of active sensors are increased from 35 to 60 sensors. In addition, the experiments have been conducted for all proposed algorithms and network without applying game for comparison. In addition, these experiments are repeated with $M=1$ and $M=3$ as shown in Fig. 3(a) and 3(b).

As can be revealed from the Fig. 3 the MEG has higher lifetime than the other algorithms and the network without applying the Gur game has lowest lifetime at $M=1$ and $M=3$. For instance, at $M=1$, when the optimum number is 45 sensors, the lifetime for MEG, APGur, PGet, AGur, Gur and without game is 2771, 1990, 1546, 1404, 1339 and 1078 iterations, respectively. Clearly MEG has 39% increase in lifetime than APGur game and APGur has 78% increase in lifetime than the other algorithms and the network without game. However, when $M=3$, for instance, when the optimum number is 45 sensors, lifetime for MEG, APGur, PGet, AGur, Gur and without game is 2771, 1919, 1552, 1439, 1323 and 1078 iterations, respectively. Clearly MEG has 44% increase in lifetime than APGur game and APGur has 78% increase in lifetime than without the game.

B. Gur Game and its Modifications Comparison with Respect to QoS Ratio:

Again with increasing the number of optimum nodes from 35 to 60, the QoS ratio is monitored in this section. It is noted that the AGur and APGur have the best QoS level
than the other algorithms at M=1 as in Fig. 4(a). For instance, at M = 1 as in Fig. 4(a), when the optimum number is 45 sensors, the QoS ratio for AGur, APGur, Gur, PGur, MEG, and without game is 99.7%, 99.6%, 99.5%, 98.6%, 94.9% and 4.7% ratios, respectively. Clearly, the AGur and APGur have 1% increase in QoS ratio than Gur and PGur while Gur and PGur have 94% increase in QoS ratio than without the game.

Note that in network without game the average residual energy per node is relatively small. Because, at each round, the sensors energy is depleted when trying to retransmission their messages to the sink node in case of rejection (where the rejection occurs when the number of messages sent is larger than the required messages). But when the number of sent messages is less than or equal to the required messages, the sink node will broadcast message to all the sensors at each new message received. Therefore, their energy is depleted quickly.

The QoS ratio of network without the game is not included in the Fig. because it is not good because of, in this network, at each round a random number is generated which indicates to the number of sensors that will send their messages to the sink node. Then the sink node will receive these messages if the random number is less than the required number of messages otherwise, it will reject them. In case of rejection the sensors will try to retransmission their messages again and consequently this affects on its QoS level and its energy.

However, at M = 3 as in Fig. 4(b), the QoS ratio for AGur and MEG have the best QoS level than the others. For instance, when the optimum number is 45 sensors, the QoS ratio for AGur, APGur, Gur, PGur, MEG, and without game is 99.8%, 84%, 88.3%, 54%, 94.9%, and 4.7% ratios, respectively.

C. Gur Game and its Modifications Comparison with respect to average residual energy:

Fig. (5) show the average residual energy after the first sensor die for all algorithms and network without applying the game when M = 1and M = 3 with increasing the number of optimum sensors.

As shown in Fig. (5) the MEG has less residual energy than the other algorithms at M = 1 and M = 3. For instance, at M = 1 as in Fig. 5(a), when the optimum number is 45 sensors, the average residual energy for MEG, APGur, PGur, AGur, and Gur is 0.27, 0.32, 0.36, 0.37 and 0.39 joule, respectively. If we compare Fig. 5 with Fig. 3 we can find that as the lifetime increases the average residual energy decreases and the network with higher lifetime, it has less residual energy.

V. CONCLUSIONS

In this paper, we introduced three Game algorithms which are APGur, PGur, and AGur are compared with simple network without applying the game. The proposed techniques are APGur and AGur seek to decrease the amount of time required to reach to stability and sustain the desired number of active sensors. In addition, the paper presented a novel technique based on the game theory which is named MEG. The MEG is adapted for WSNs. The new technique performance is tested against the network without

![Graph](image1.png)

Fig. 3. Comparison of network lifetime for all game algorithms (a) when M = 1 and (b) when M = 3.

![Graph](image2.png)

Fig. 4. Comparison of QoS ratios for all game algorithms (a) when M = 1 and (b) when M = 3.

![Graph](image3.png)

Fig. 5. Comparison of average residual energy after the first sensor die for all algorithms and network without applying the game when M = 1 and M = 3.
applying the game. The simulation results showed that the MEG technique has higher lifetime and less residual energy than other algorithms. Our future work relies on using all techniques in distributed network and with assuming certain clustering structure.

Fig. 5. Comparison of average residual energy for all game algorithms (a) when M = 1 (b) when M = 3.

REFERENCES