Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 32 (2014) 1168 – 1173

Procedia Computer Science

International Workshop on Wireless Networks and Energy Saving Techniques (WNTEST-2014)

Modified gur game for WSNs QoS control

Eman M. Elshahed[a,*], Rabie A. Ramadan[b], Shahinaz M. Al-tabbakh[a], H. El-zahed[a]

[a] *Computer Applications- Physics Dept., Faculty of women for Art, Science and Education, Ain Shams University, Cairo, Egypt*
[b] *Computer Engineering Dept., Faculty of Engineering, Cairo University, Egypt*

Abstract

Wireless Sensor Networks (WSNs) have been utilized by many critical applications such as military and health care monitoring. Such applications require special Quality of Service (QoS) treatment. However, the components of sensor nodes suffer from different limitations including the scarce energy source, limited processing capabilities, and limited storage space. At the same time, nodes are supposed to live for long time. This could occur when the energy is balanced on the active nodes in the network. This paper handles the problem of QoS in Wireless Sensor Networks (WSNs). The paper considers the limitation of the nodes in handling many messages at the same time especially the sink node. Therefore, nodes can coordinate 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 QoS control enhances the operation of the network and positively affect the nodes' consumed energy. This work applies one of the game theory schemes which is Gur game. The paper proposes two QoS control techniques with Quick Convergence called Periodic Gur Game (PGur) and Adaptive Periodic Gur Game (APGur). These two approaches are evaluated using different sets of experiments.

Keywords: Wireless sensor networks; periodic gur game; adabtive periodic gur game; quality of service; network lifetime

1. 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

* Corresponding author. Tel.: +201141020134
E-mail address: eman.elshahedsc86@gmail.com

doi:10.1016/j.procs.2014.05.549

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, 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[4] and clustering[5]. 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. The Gur game algorithm is included in different studies such as in [6] to control the amount of messages that reach the sink node.

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 = 2 of Gur Game algorithm. Each state has its own meaning. States -1 and -2 represent sleep states, whereas states 1 and 2 represent active states. Only one state is allowed for a player to be in and can transit to only its adjacent state.. 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 two modifications to the Gur game to fit WSNs requirements. The first modification considers the Periodic Gur game and the second modification considers the Adaptive Periodic Gur game. The paper is organized as follows: Section two presents the problem statement; section three proposes solution approaches; section four displays the simulation results with the evaluation for all techniques; finally, the paper concludes in section five.

2. 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. 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.

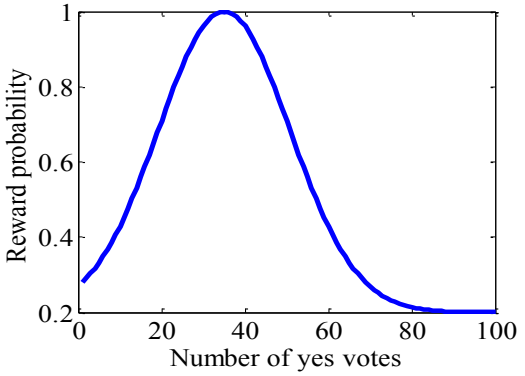

Fig. 1. Typical Gur Reward Function **[6]**

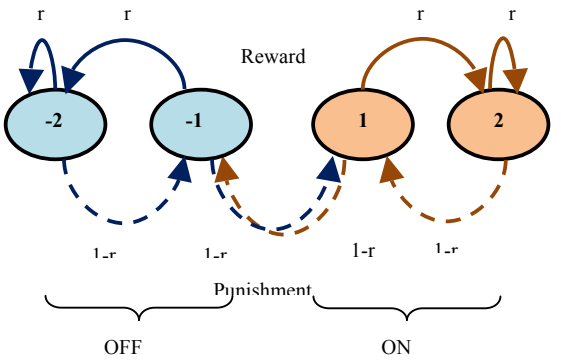

Fig. 2. An example on the automaton with 4 states for the Gur game

3. Solution Approaches

In this section, the revision of Gur game algorithm to fit WSNs is presented then the proposed techniques are presented starting by the Periodic Gur game. In addition, an Adaptive Periodic Gur game is presented for enhancing the QoS control.

3.1. Gur Game

For the Gur game to fit QoS control of WSNs operation, each sensor node is considered as a player and the base station (sink node) as the referee. Assume that there is a collection of m nodes from S1 to Sm 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. Therefore, the mathematical paradigm of the Gur game is used to adjust the optimum number of active sensors dynamically.

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 (r) used by the base station could be given as follows:

$$r = 0.2 + 0.8e^{v} \quad (1)$$

$$v = -0.002(k_t - n)^2 \quad (2)$$

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 $1\text{-}r(k_t)$.

3.2. 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 [7]; 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).

3.3. Adaptive Periodic Gur Game (APGur)

Another problem that has been discovered during running the Periodic Gur game is the long period of time, iterations, are required for the game to be stabilized. 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 time of the Gur Game algorithm takes too long time. To enhance the convergence time of the Gur Game, it was suggested to use unambiguous reward/punishment mechanism [8]. 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 receives 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. From the previous discussion, it seems the merging the adaptive concept along with the periodic technique, sensor network operations might be enhanced. By adding the adaptation using unambiguous reward/punishment, the game convergence time is expected to be enhanced. At the same time, after reaching the stability, the system could be left for specific period of time then the game will be initialized again for energy balancing.

4. 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 desires a rate of 35 packets received at each time t. The reward function used by the sink node is considered as given 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 [9] 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- Convergence: it is the number of trails for the WSN reaches stability.

2- Life time : it the time it takes the network for the first node to die.

Based on these criteria, the proposed methods are compared to the network without game and the designed experiments include the effect of changing the desired number of sensors on the WSN lifetime and the QoS ratio. The results of the following experiments are the average over different network topologies, and settings.

4.1. Periodic Gur Game Convergence

One of the important criteria of measuring the performance of utilizing Periodic Gur game in WSNs QoS control is the convergence. As can be seen in Fig. 3, the relation between the time and the received messages (packets) at the sink node is illustrated. Fig. 3 shows the convergence at both M=1 and M=3. From the Fig., it has been noticed that the network takes too long time to reach stability. Stability in this case occurs when the sink node receives only the desired number of messages regardless the number of its neighbors. When M=1 , the number of messages fluctuates up and down for 271 iterations out of 2000 iterations for the network to reach stability until the time of restarting the Gur game again (1000 runs); then the game starts again fluctuating up and down for 132 iterations. However, when M= 3, the system doesn't reach to stability but in average it is close. Therefore, we can conclude that the minimum number of required iterations for the network to be stabilized is 132 iterations. Nevertheless, the time that we need to restart the Gur game after it can be estimated based on the network properties and requirements.

4.2. Adaptive Periodic Gur Game Convergence

As mentioned, the purpose of APGur is to speed up the convergence of the PGur game in the network. In this section, another set of experiments applying the concept of adaptive Periodic Gur game. As can be seen in Fig. 4 , the number of received messages by the sink node is monitored along the time with M=1 and M=3. Based on the results shown in the Fig. 4 compared to the Periodic Gur game results, stated in the previous section, the network seems to reach stability faster than before.

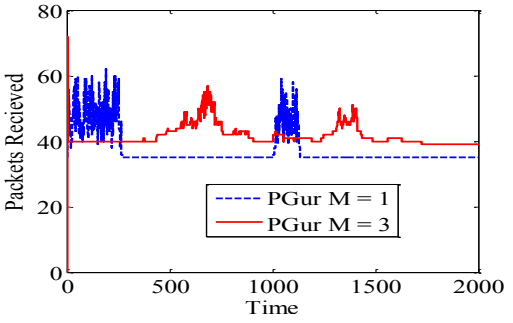

Fig. 3. Number of active sensors vs. time for M = 1 and M = 3 .

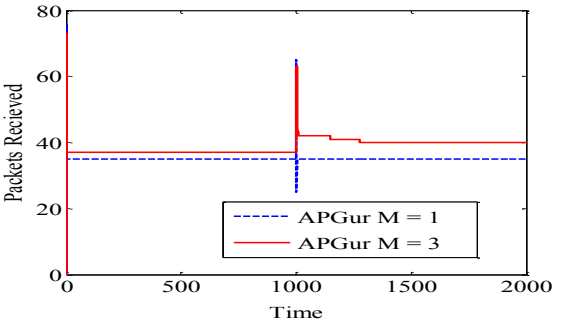

Fig. 4. Number of active sensors vs. time for M = 1 and M = 3.

In fact, the network needs only few trails to reach stability; for instance when M=1 only 5 trails, cannot be clearly shown on the graph, are needed while when M=3 only 6 trials are required for network stability. Therefore, the adaptation of Gur game seems to be effective.

4.3. Network Lifetime Evaluation

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. The number of active sensors are increased from 35 to 60 sensors. In addition, the experiments have been conducted for two proposed algorithms

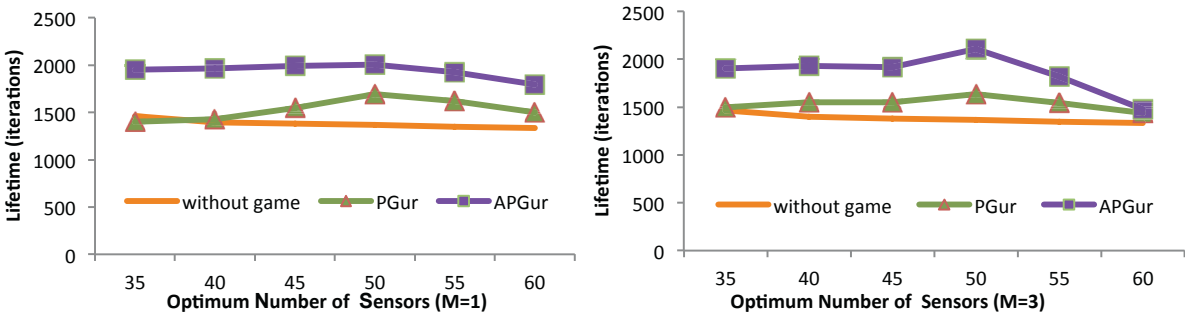

Fig. 5. Comparison of network lifetime for two algorithms (a) when M = 1 (b) when M = 3.

and network without applying the game for comparison. In addition, these experiments are repeated with M = 1 and M =3 as shown in Fig. 5a and 5b. As can be revealed from the Fig. 5, when M =1, the lifetime accomplished by APGur is better than the lifetime of PGur and the lifetime of network without applying the game in all cases. For instance, when the optimum number is 45 sensors, the lifetime for APGur, PGur and without game is 1991, 1546 and 1381 iterations, respectively. Clearly APGur has 29 % increase in lifetime than PGur game and PGur has 12 % increase in lifetime than without game. However, when M=3, the lifetime of APGur, as expected, is much better than PGur and without game. For instance, when the optimum number is 45 sensors, the lifetime for APGur, PGur and without game is 1919, 1552 and 1381 iterations, respectively. Clearly APGur has 24 % increase in lifetime than PGur game and PGur has 12.4 % increase in lifetime than without game.

5. Conclusions

In this paper, we introduced two QoS control algorithms which are PGur and APGur. The proposed is APGur seeks to decrease the amount of time required to reach to stability and sustain the desired number of active sensors. With different sets of experiments, both showed a reasonable enhancement in the network performance. The two proposed technique are tested against the network without game where, each round, the sink node receives the desired number of messages based on its capability and reject the remainder. The APGur has higher lifetime at M = 1 and M = 3 than PGur and network without game. Our future work relies on using PGur and APGur in distributed network with certain clustering.

References

[1] Stojmenovic, *Handbook of Sensor Networks: Algorithms and Architectures*, John Wiley & Sons Inc., Canada, 2005.
[2] C. Morais Cordeiro and D. P. Agrawal, *Adhoc & Sensor Networks Theory and Applications*, World Scientific Publishing Co., London, 2006.
[3] B. Yang, *Reliable Data Delivery In WSN*, M.Sc. Thesis, College of Graduate Studies and Research, Saskatchewan University, Canada, 2010.
[4] B. P. S. Sahoo, S. Rath, D. Puthal, Energy Efficient Protocols for WSN, *I. J. of Computer Applications* 44, 42-48, 2012.
[5] R. Krishnan, D. Starobinski, Efficient Clustering Algorithms for Self-Organizing WSN, *J. Ad-Hoc Networks*, 1-39, 2004.
[6] R. Iyer, L. Kleinrock, QoS Control for Sensor Networks, *IEEE Trans. on Communications, ICC. 1*, 517-521, 2003.
[7] R.G. Tsai, H.L. Wang, Shuffle: An Enhanced QoS Control by Balancing Energy Consumption in Wireless Sensor Networks, *Advances in Grid and Pervasive Computing*, 1-9, 2010.
[8] M. Ayers, Y. Liang, Gureen Game: An Energy-Efficient QoS Control Scheme for Wireless Sensor Networks, Green *Computing Conference and Workshops (IGCC)*, Orlando, 1-8, 2011.
[9] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy Efficient Communication Protocol for Wireless Microsensor Networks, *Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS)*, IEEE, 1-10, 2000.