Hindawi Publishing Corporation
ISRN Sensor Networks
Volume 2013, Article ID 185719, 6 pages
http://dx.doi.org/10.1155/2013/185719

Research Article

QC^2: A QoS Control Scheme with Quick Convergence in Wireless Sensor Networks

Hao-Li Wang and Wei-Lun Hung

Department of Computer Science and Information Engineering, National Chiayi University, Chiayi 60004, Taiwan

Correspondence should be addressed to Hao-Li Wang; haoli@mail.ncyu.edu.tw

Received 5 December 2012; Accepted 10 February 2013

Academic Editors: I. Khalil and A. Olteanu

In wireless sensor networks, too many or too few power-on sensors may cause the waste of resources or poor sensing efficiency; thus, controlling the number of active sensors to meet the predicted target number is the purpose of this research. However, the total number of sensors may be unstable because of the increment and damage to the sensors. It is difficult to control the number of active sensors to meet the predicted target in this condition. Previous studies proposed the Gur Game algorithm to solve this problem. However, the convergence time of the Gur Game algorithm is too long, which causes sensors to consume excessive power and waste resources. Therefore, this paper proposed the QoS Control with Quick Convergence (QC^2). This method utilizes total virtual value to accelerate the convergence operation from the number of sensors to the target number. The experiment result shows that the QC^2 method can cause the number of sensors to converge rapidly with the target value and that QC^2 can be over a hundred times faster than the Gur Game algorithm with regard to convergence.

1. Introduction

With the recent technological developments of the wireless networks and multifunctional sensors with processing and communication capabilities, wireless sensor networks (WSNs) have been used in an increasing number of applications. WSNs can provide a more accurate or reliable monitoring service for different classes of applications. Quality of service can be an important mechanism to guarantee that the distinct requirements for different classes of applications are met.

A WSN consists of a large number of small sensors and a sink (base) station. Sensors are small devices with limited energy supply and low computational capability. They are used for covering and monitoring a sensing field to collect useful information. Sensor networks are useful in a variety of domains, such as environmental observation, health care, and military monitoring.

Sensors are usually placed randomly in a sensor field. The concept of redundancy is applied to WSNs to achieve a high degree of reliability. One or more sensors may cover the same region and gather similar data; thus, numerous redundant data are sent to the sink. Redundant data collection should be avoided to conserve energy in WSNs. Sensors are scheduled to be periodically active and idle. Only several sensors are active in a given period of time, resulting in high reliability and low data redundancy. The assumption that a sensor network has a fixed number of nodes is unreasonable because sensor nodes usually have limited lifetimes. Therefore, adding and deleting sensor nodes have to be taken into consideration.

New questions arise based on the addition and deletion of sensor nodes, such as the selection of active nodes in all sensors, which may be added or deleted randomly. This subject is known as quality of service (QoS) control in WSNs. QoS is defined as the number of active sensors that can send information at any given time.

Most studies on WSNs focused on medium access control, routing, and data aggregation. Only a few studies discussed QoS control. An earlier research introduced a QoS control approach based on the Gur Game algorithm. The Gur Game-based scheme maintains QoS without knowing the total number of sensors. However, the convergence time of the Gur Game algorithm is too long, which causes sensors to consume excessive power and waste resources. Therefore, this paper proposed the QoS Control with Quick Convergence (QC^2). This method utilizes total virtual value to accelerate the convergence operation from the number of sensors to the target number.

Our contributions are twofold. QC^2 enhances prior work by utilizing total virtual value. Furthers, QC^2 significantly improves convergence rate and can be over a hundred times faster than the Gur Game algorithm with regard to convergence.

The remainder of this paper is organized as follows. Section 2 presents several related studies. Section 3 describes the system model, problem definition, and the proposed solution. Section 4 displays the simulation results. Finally, Section 5 concludes the paper.

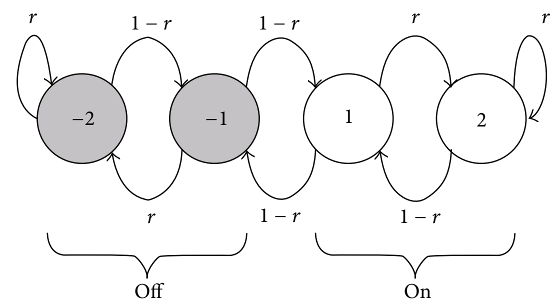

Figure 1: An example of the automaton with 4 states for the Gur Game-based QoS Control Scheme.

2. Related Work

2.1. Previous Literature on QoS Control in WSNs. WSNs have been attracting the attention of researchers' for the past years. A huge amount of general literature on WSNs exists. However, few studies focused on controlling the number of power-on sensors to a desired target number. This subject is also called QoS control. Although QoS control is not a hot issue in WSNs, previous studies on this topic still exist. Iyer and Kleinrock [1] defined the QoS control problem and proposed the first QoS control approach based on the Gur Game algorithm. Their study motivated our work in this paper. A brief introduction of the Gur Game-based scheme is provided later in this section.

Many researchers extend the study of Nayer and Ali in different ways [2–9]. Some studies are concerned with energy conservation in QoS control scheme [2–5], whereas others extend QoS scheme to cluster structures [6–9]. In addition, WSN lifetime is defined in [7–9] as the maintenance duration of the desired QoS.

Other related studies are briefly introduced as follows. A new WSN taxonomy with QoS is proposed in [10]; a reference model that enables the classification of WSNs is also established in this paper. A survey of QoS-aware routing techniques in WSNs is presented in [11]; middleware approaches and certain open issues for QoS support in WSNs are also explored. A traffic engineering model that relies on delay, reliability, and energy-constrained paths to achieve fast, reliable, and energy-efficient transmission of information routed by a WSN is proposed in [12]; this paper uses multipath routing to improve reliability and packet delivery in WSNs while maintaining low power-consumption levels. QoS requirement and the minimum number of active nodes are analyzed in [13] because the former is usually inversely proportional to energy consumption. A QoS protocol for WSNs that controls topology based on analytical results is proposed [13]. In addition, a dynamic clustering algorithm is presented to achieve the optimal assignment of active sensors while maximizing the number of regions covered by the sensors [2]. Ant algorithm and genetic algorithm are also taken into consideration in QoS control. A tradeoff between sensing coverage and network lifetime necessitates the use of a routing protocol, which was proposed in [14], to accommodate both energy balance and coverage preservation for sensor nodes in WSNs.

Although several aspects of QoS control in WSNs have been extensively investigated, however, slow convergence rate is relatively unexplored. To the best of our knowledge, the current work is the first attempt in solving the problem of slow convergence rate in QoS control.

2.2. A Gur Game-Based QoS Control Scheme. We introduce the use of the Gur Game algorithm in controlling QoS in this section. The principle of the Gur Game algorithm is based on biased random walks of finite-state automata. The automata describe a set of states with assigned meanings and a set of rules to determine switches from one state to another. Figure 1 is a simple example of a finite-state automaton with four states for the Gur Game algorithm. Each state has its own meaning. States −1 and −2 represent sleep modes, whereas states 1 and 2 represent active modes.

The key in the Gur Game scheme is the reward function. The reward function is responsible for measuring performance of the system. An example of the reward function is given as follows:

$$R^*(t) = 0.2 + 0.8 \exp\left(-0.002(K_t - n)^2\right), \quad (1)$$

where K_t is the number of active nodes and n is the desired QoS value. When K_t is close to n, the R value approaches the top value (1). Figure 2 shows an example of the reward function with $K_t = 35$.

In a WSN, the number of active sensors contained in the sink (base) is determined by the number of received data from the active sensors. The sink broadcasts the reward value R to all sensors using the information and reward functions. The sensors can then determine whether to be active or idle in the next iteration based on the received R value from the sink. The decision is made by each sensor based on the finite-state automaton, the current state, and the received R value. The Gur Game algorithm enables the number of active sensors to reach the target after a certain number of iterations.

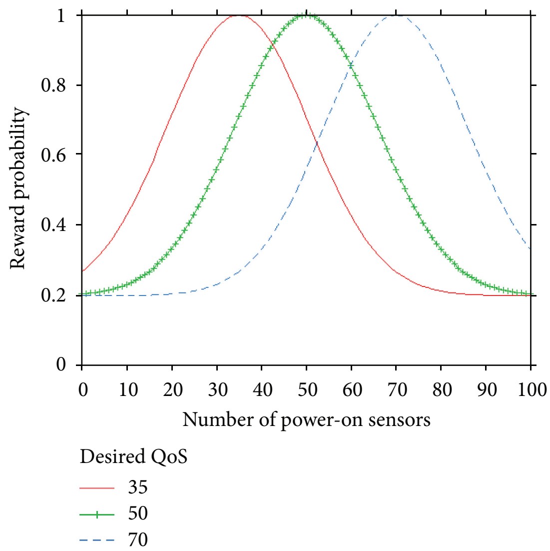

Figure 2: Examples of reward function with $K_t = 35, 50$, and 70.

3. QoS Control with Quick Convergence (QC^2)

The total number of sensors is not fixed, so it is difficult to control the number of sensors in the ON state. This research sets a total virtual number by using the number of sensors in the ON state to calculate the transition probability which will, in turn, adjust the number of sensors in the ON state to the target value. This concept is equivalent to the operation wherein the total number is known; thus, this method can cause the number of sensors in the ON state to reach the target rapidly.

3.1. Overview of the Methods. In the research method employed, data aggregation end determines a total virtual value by the number of sensors in the ON state. This total virtual value and the target value become the bases in determining the status switching probability. For example, when the number of sensors in the ON state is 70 and the target value is 35, data aggregation end will set 70 as the total virtual value and set the switching probability to 35/70, which is regarded as the switching probability of the sensor. Information relating to the switching probability and current network status (too many, too few, or equal to the target value) is then broadcasted to the other networks.

After receiving this information, the sensors will determine if status switching should be done on the basis of the current network state. If there are too many sensors in the ON state, the operating sensors would conduct status switching according to the switching probability. Conversely, if there are too few sensors in the ON state, the sleep sensors would conduct status switching. Provided that the target value is reached, all sensors would keep their original status. This operation method will be utilized to establish QoS Control.

3.2. Operation of Data Aggregation End. In the detailed operation of QC^2, data aggregation end determines the value of $f(k_t)$ according to the number of sensors in the ON state. The $f(k_t)$ function is used to reflect the current network state. When the number of sensors in the ON state is too big, $f(k_t)$ registers the value 1; $f(k_t)$ registers 0 when the number of sensors is too small. When the number of sensors reaches the target value, $f(k_t)$ registers –1. The corresponding relation of $f(k_t)$ is shown in the following:

$$f(k_t) = \begin{cases} 1, & k_t > n \\ 0, & k_t < n \\ -1, & k_t = n. \end{cases} \quad (2)$$

When the total number of sensors in the ON state is more than the target value, it means that there are too many sensors in the ON state. Switching probability is expected to reduce the number of sensors in the ON state. Therefore, the number of sensors in the ON state is set to the current virtual value to calculate the probability of status switching T_p. The calculation of T_p is shown as (2). In the equation, n refers to the target value and k_t refers to the total number of sensors in the ON state. The difference between k_t and n indicates if the number of sensors is greater than the target value, that is, too many sensors. Then, the difference is divided by the total number to obtain the switching probability, that is, the probability of switching to the sleep status:

$$T_p = \frac{|k_t - n|}{k_t}. \quad (3)$$

When the number of sensors in the ON state is less than the target value, the calculation of status switching probability is shown as (3). The sensors in the ON state are too few, so the sleep sensors must be reduced to increase the effectiveness of the sensors in the ON state. In this case, we set the lower limit of switching probability to 0.5 to prevent excessively lengthy adjustment time which is usually caused by a too low switching probability limit:

$$T_p = \mathrm{Max}\left(0.5, \frac{|k_t - n|}{k_t}\right). \quad (4)$$

The switching probability T_p and $f(k_t)$ can be obtained after calculating data aggregation end. The resulting information about T_p and $f(k_t)$ is broadcasted to all the sensors.

3.3. Operation of Sensors. In QC^2, sensors are equipped with additional flags referred to as cf. The flags aim to remove excessive sensors and keep these excessive sensors in the sleep status. Moreover, these flags can boost the operation of the sensors to reach the target value. When cf is 1, there is no status switching. When it is 0, there may be status switching. After receiving the information about T_p and $f(k_t)$ at data aggregation end, the sensors determine if status switching should be made based on $f(k_t)$ and cf. If status switching is necessary, the sensors will conduct status switching according to the T_p value. Otherwise, the sensors will remain in their status quo.

In sensors, the operation of the finite state machine can be described by the following three conditions.

(i) Provided that there are too many current sensors in the ON state, that is, $f(k_t)$ is equal to 1, the status of the sleep sensors remains unchanged. Moreover, cf marked as 1 also indicates that the succeeding regulation will not change its status. The sensors in the ON state confirm if switching to the sleep state should be done based on T_p.

(ii) Provided that there are too few current sensors in the ON state, that is, $f(k_t)$ is equal to 0, the status of the sensors in the ON state remains unchanged. The sleep sensors with cf unmarked as 1 confirm if switching to the operating state based on T_p would be done.

(iii) Provided that the current sensors in the ON state are equal to the target value, that is, $f(k_t)$ is equal to −1, the status of all the sensors remains unchanged.

4. Simulation Experiment

This experiment is divided into four parts. The first part establishes a comparison of convergence time and the Gur Game algorithm. The second part establishes a comparison of the rates at which the standard of QoS was reached. The third part analyzes the impact of the change in the total number of sensors in the experiment on convergence time. The final part determines the status of QC^2 in an environment of life and death. The experiment area is 100 m × 100 m. With regard to the third part of the experiment, the total number of sensors in the other experiments is 100 with the target value of 35.

4.1. Comparison of Convergence Time. The Gur Game method requires too much time; it can reach the target value with the average convergence of 4000 rounds to 5000 rounds (QoS). The QC^2 method proposed in this research can shorten the convergence time to 1 round to 10 rounds by means of the virtual target value, which is about 3.95 rounds on the average. It can be seen from Figure 3 that when the target value is 0–100 (availability of the Gur Game method is about 30%–70%), the convergence time of QC^2 is obviously shorter than that of the Gur Game method. The Gur Game method can facilitate convergence only when the target value is in the range of 30%–70%. Therefore, there is only data in 30–70.

Figure 4 refers to the convergence of QC^2 when the total number is 100 and the target value is 35. It can be seen from the figure that, in the convergence process, when the sensors in the ON state are too many, the sleep sensors will adjust their status without switching because of the cf flag. Therefore, in the later rounds, the number of sensors in the ON state is not more than the maximum. For example, in Figure 7, the number of sensors in the ON state in the initial environment is 45; we can see from the figure that the number of sensors in the ON state does not exceed 45 in the succeeding rounds.

4.2. Comparison of the Achievement Rate of QoS. This part of the study aims to compare the achievement rate of the Gur Game method with that of QC^2 after operating for 10,000 rounds. The achievement rate of QoS is calculated in the following:

$$\text{Achievement rate of QoS} = \frac{\text{Round for reaching the standard}}{\text{Total round}}. \quad (5)$$

Figure 5 refers to the number of rounds it took for QC^2 and Gur Game to reach the target value of 35. We can see from the figure that the achievement rate of QC^2 is 99% and the achievement rate of Gur Game is 66%.

QC^2 is superior in this aspect because of its rapid convergence time. The convergence time of Gur Game is too long, so it needs many rounds to make the adjustment. Therefore, QC^2 has a higher achievement rate of QoS than the Gur Game method.

4.3. Convergence Trend as Total Number is Changed. Figure 6 refers to the tendency chart that shows how the total number of sensors changes when the target value is changed to 35. In the figure, the average convergence time of the QC^2 method does not increase when the total number of sensors is increased. The impact of the total number of sensors on the convergence time of the QC^2 method is therefore not that great. This research finding can be applied to an environment with plenty of sensors.

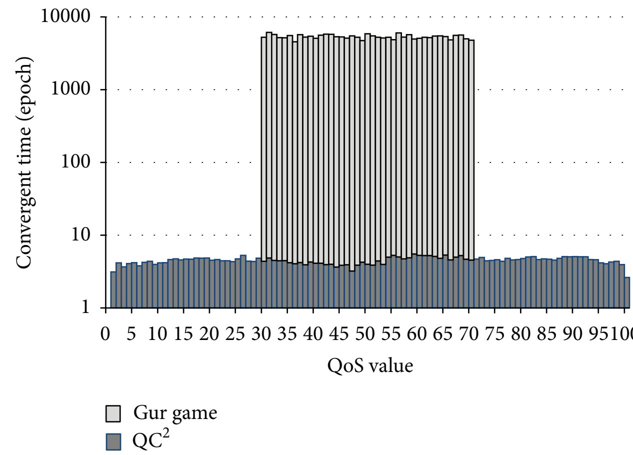

Figure 3: Comparison of convergence time of Gur Game versus QC^2.

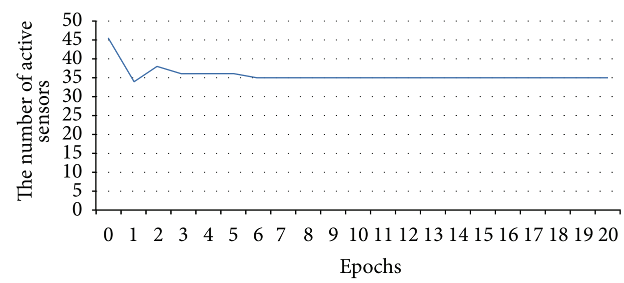

Figure 4: Convergence of QC^2 in 0 round to 20 rounds.

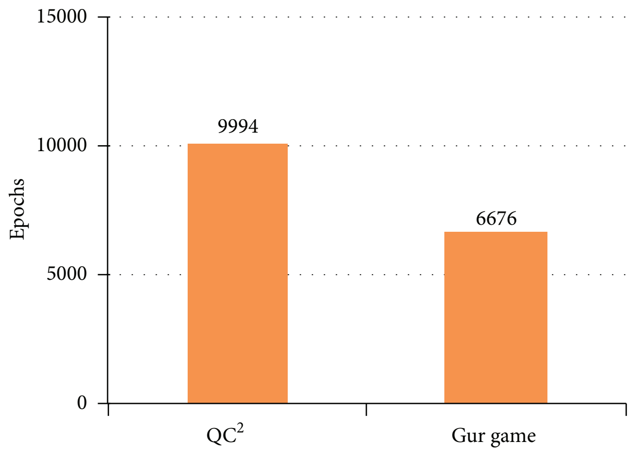

FIGURE 5: Rate of reaching the standard of QoS.

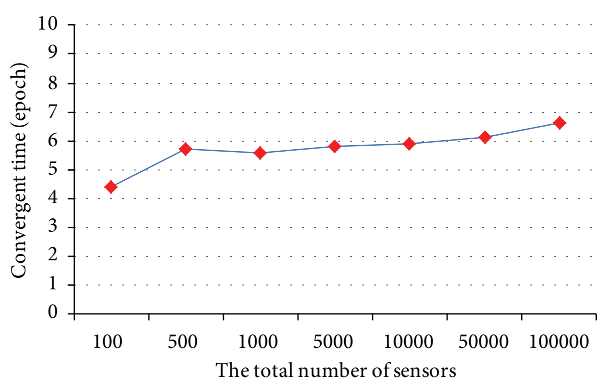

FIGURE 6: Convergence trend graph.

4.4. Impact of a Life-and-Death Environment. In this part of the study, we test the performance of QC^2 in an environment of life and death. The life-and-death environment model refers to the life-and-death environment in [1]. In this model, the sensors that are deployed initially are not damaged or dead. Additional sensors are added in the implementation of the test. The dates of birth of these additional sensors are generated from the distribution index with the average value of 100. The life span is generated from the distribution index with the average value of 101. The experiment is done for 20,000 rounds with the target value set at 35. Figure 7 refers to the operation situation of QC^2 in the life-and-death environment. Because the virtual target value method is used, the convergence time is accelerated. Therefore, when some sensors die, QC^2 converges rapidly to the target value. Its achievement rate of QoS can reach up to 96.7%.

5. Conclusion

This research proposed an enhanced QoS Control method with Quick Convergence, namely, QC^2. The idea of total virtual value is utilized in QC^2. This method allows data aggregation end to do the calculation in cases where the total number of sensors is known by means of setting the

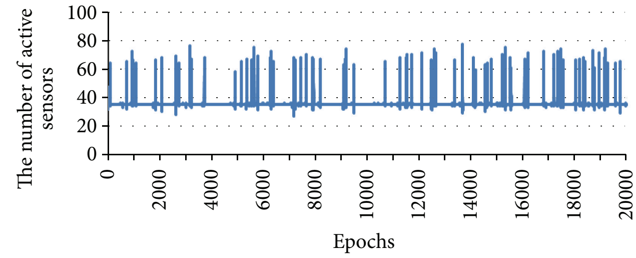

FIGURE 7: Changes in the life-and-death environment.

total virtual value. This method shortens the amount of time required for the number of sensors in the ON state and the target value to converge. The convergence time of QC^2 is shorter than that of the Gur Game method proposed by previous studies. QC^2 can complete its convergence in 20 rounds, on the average. This method has shorter convergence time, so the switching frequency of the sensors is lower. The experiment result shows that the QC^2 can cause the number of sensors to converge rapidly with the target value and that QC^2 can be over a hundred times faster than the Gur Game algorithm with regard to convergence.

References

[1] R. Iyer and L. Kleinrock, "QoS control for sensor networks," in *Proceedings of the IEEE International Conference on Communications (ICC '03)*, pp. 517–521, May 2003.

[2] S. I. Nayer and H. H. Ali, "A dynamic energy-aware algorithm for self-optimizing wireless sensor networks," *Lecture Notes in Computer Science*, vol. 5343, pp. 262–268, 2008.

[3] R. G. Tsai and H. L. Wang, "Shuffle: an enhanced QoS control by balancing energy consumption in wireless sensor networks," *Lecture Notes in Computer Science*, vol. 6104, pp. 603–611, 2010.

[4] R. G. Tsai and H. L. Wang, "A coverage-aware QoS control in wireless sensor networks," in *Proceedings of the International Conference on Communications and Mobile Computing (CMC '10)*, pp. 192–196, April 2010.

[5] L. Zhao, C. Xu, Y. Xu, and X. Li, "Energy-aware QoS control for wireless sensor network," in *Proceedings of the IEEE International Conference on Industrial Electronics and Applications (ICIEA '06)*, pp. 1–6, 2006.

[6] D. Chen and P. K. Varshney, "QoS support in wireless sensor networks: a survey," in *Proceedings of the International Conference on Wireless Networks (ICWN '04)*, pp. 227–233, June 2004.

[7] J. Kay and J. Frolik, "Quality of service analysis and control for wireless sensor networks," in *Proceedings of the IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS '04)*, pp. 359–368, October 2004.

[8] B. Liang, J. Frolik, and X. S. Wang, "Energy-efficient dynamic spatial resolution control for wireless sensor clusters," *International Journal of Distributed Sensor Networks*, vol. 5, no. 4, pp. 361–389, 2009.

[9] J. Frolik, "QoS control for random access wireless sensor networks," in *Proceedings of the IEEE International Conference on Wireless Communications and Networking Conference (WCNC '04)*, pp. 1522–1527, 2004.

[10] V. Pereira, "A taxonomy of wireless sensor networks with QoS," in *Proceedings of the IFIP International Conference on New Technologies, Mobility and Security (NTMS '11)*, pp. 1–4, 2011.

[11] B. Bhuyan, H. Sarma, N. Sarma, A. Kar, and R. Mall, "Quality of Service (QoS) provisioning in wireless sensor networks and related challenges," *Wireless Sensor Network*, vol. 2, no. 11, pp. 861–868, 2010.

[12] A. B. Bagula, "Modelling and implementation of QoS in wireless sensor networks: a multiconstrained traffic engineering model," *Eurasip Journal on Wireless Communications and Networking*, vol. 2010, Article ID 468737, 2010.

[13] D. Changmin, "Topology controlling for QoS in wireless sensor networks," in *Proceedings of the 5th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '09)*, pp. 1–4, September 2009.

[14] J. A. Jiang, T. S. Lin, C. L. Chuang et al., "A QoS-guaranteed coverage precedence routing algorithm for wireless sensor networks," *Sensors*, vol. 11, no. 4, pp. 3418–3438, 2011.