International Journal of Ad hoc, Sensor & Ubiquitous Computing (IJASUC) Vol.4, No.2, April 2013

APPLICATION OF PERIODICAL SHUFFLE IN CONTROLLING QUALITY OF SERVICE IN WIRELESS SENSOR NETWORKS

Hao-Li Wang[1] and Rong-Guei Tsai[2]

[1]Department of Computer Science and Information Engineering,
National Chiayi University, Chiayi, Taiwan
haoliwang@gmail.com
[2] Department of Engineering Science,
National Cheng Kung University, Tainan, Taiwan
sig73ma@gmail.com

ABSTRACT

This paper addresses problems in controlling quality of service (QoS) in wireless sensor networks (WSNs). QoS is defined as the number of awakened sensors in a WSN. Sensor deaths (caused by battery failure) and sensor replenishments (caused by redeployment of new sensors) contribute to the difficulty of controlling QoS in WSNs. A previous research developed a QoS control scheme based on the Gur Game algorithm. However, this scheme does not consider the energy consumption of sensors, which shortens WSN lifetime. This paper proposes a novel QoS control scheme that periodically swaps active and sleeping sensors to balance power consumption. Our study also uses the distribution manner of the previous work. Our scheme significantly extends WSN lifetime and maintains desired QoS. Simulations that compared our scheme with previous schemes in various environments show that our scheme build a robust and long-lasting sensor network capable of dynamically adjusting active sensors.

KEYWORDS

Wireless Sensor Network, Quality of Service, Gur Game

1. INTRODUCTION

Rapid developments in wireless communication, distributed signal processing, and ubiquitous computing make wireless sensor networks (WSNs) popular in the last few years. 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 widely used 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.

DOI : 10.5121/ijasuc.2013.4202

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.

In WSN, QoS has different meanings in different applications. For applications of event detection and target tracking, QoS means the coverage of the WSN. For applications in sensing harsh environments, QoS means observation accuracy. For multimedia applications in WSN, QoS means information transportation related parameters. In this research, we consider the applications of event detection, and thus define QoS as the number of active sensors that can send information at any given time.

We have two goals regarding QoS control design. The first involves maximizing the lifespan of the sensor network. The second is concerned with having enough working sensors to send packets toward the sink. The lifespan of a sensor network is defined as the period of time until the first sensor in the network runs out of energy. Other researchers gave a different definition for network lifetime, which they described as the duration until the active nodes can no longer perform the required task. We chose the first definition in this paper because it is the most commonly used definition in WSNs.

Most studies on WSNs focused on medium access control, routing, data aggregation, and sensor deployment. Only a few studies discussed QoS control. A previous 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 Gur Game-based scheme does not consider power consumption and causes short sensor lifetimes. Many studies have been conducted on power-saving issues in WSNs. Some studies scheduled sensors to sleep longer, whereas others reduced transmission data. Moreover, most of these studies proposed centralized methods to achieve power consumption savings. However, centralized methods cannot be applied in the Gur Game-based scheme because they may destroy the potential distribution manner of the scheme. Therefore, we propose an enhanced QoS control scheme that balances power consumption and maintains the potential distribution manner of the Gur Game-based scheme.

Our contributions are threefold. (1) This paper recognizes hidden reasons for short sensor lifetime in the Gur Game-based scheme. Sensor networks are able to last longer after the cause of energy inefficiency is removed. (2) This paper enhances prior work by balancing power consumption. To keep the potential distribution manner of the Gur Game-based scheme, our method avoids centralized schemes, which were widely used in most prior studies on power saving. (3) Our method significantly improves sensor lifetime. Simulations that evaluate shuffle in various environments show that sensors exhibit great improvements in lifetime with our method. We clarify the proposed mechanism is the revised version of our previous research [3] to include extensive simulation results.

The remainder of this paper is organized as follows. Section II presents several related studies. Section III describes the system model, problem definition, and the proposed solution. Section IV displays the simulation results. Finally, Section V concludes the paper.

2. RELATED WORKS

This section presents a number of previous literatures on QoS control in WSNs, and then introduces a Gur Game-based QoS control scheme, which is the first and the most related QoS control scheme in this field.

2.1. Previous literature on QoS control in WSNs

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

Some researches extend the study of Iyer and Kleinrock in different ways [2–9]. Some studies discussed the energy conservation in QoS control scheme [2–5], whereas others extend QoS scheme to cluster structures [6–9]. Besides, WSN lifetime is defined in [7-9] as the maintenance duration of the desired QoS.

Other related works are briefly introduced as follows. A novel WSN taxonomy with QoS is proposed in [10], where a reference model that enables the classification of WSNs is also established. A survey of QoS-aware routing techniques in WSNs is proposed in [11]; a number of 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 reliable and energy-efficient transmission of information routed by a WSN is proposed in [12]. This paper adopts 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 explored 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 in [13]. Besides, 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 considered in the design of QoS control. A trade-off between sensing coverage and network lifetime necessitate the use of a routing protocol, which was proposed in [14], to accommodate both energy-balance and coverage-preservation for sensor nodes in WSNs. Both energy consumption for radio transmission and residual energy over the network are discussed. Although references [2–5] are concerned with energy conservation in the QoS control scheme, they do not focus on imbalances in power consumption.

Although several aspects of QoS control in WSNs have been extensively investigated, however, unbalanced power consumption is relatively unexplored. To the best of our knowledge, the current research is the first attempt in solving the problem of unbalanced power consumptions in QoS control.

2.2. A Gur Game-based QoS control scheme

The Gur Game algorithm in controlling QoS is presented in this section. In short, 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 are sleep modes, whereas states 1 and 2 are active modes.

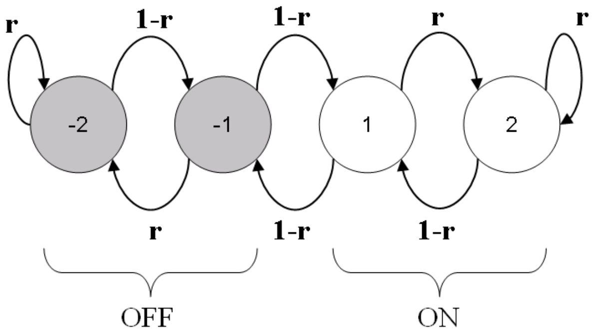


Figure 1. The automaton of Gur Game with 4 states

The reward function is the key in the Gur Game scheme and responsible for measuring performance of the system. Following equation is an example of the reward function.

$R^*(t) = 0.2 + 0.8exp\ (-0.002(K_t\text{-}n)^2)$

where K_t is the number of active nodes and n is the desired QoS value. As shown in Figure 2, when K_t is close to n, the R value approaches the top value (i.e. 1). Figure 2 presents an example of the reward function with K_t =35.

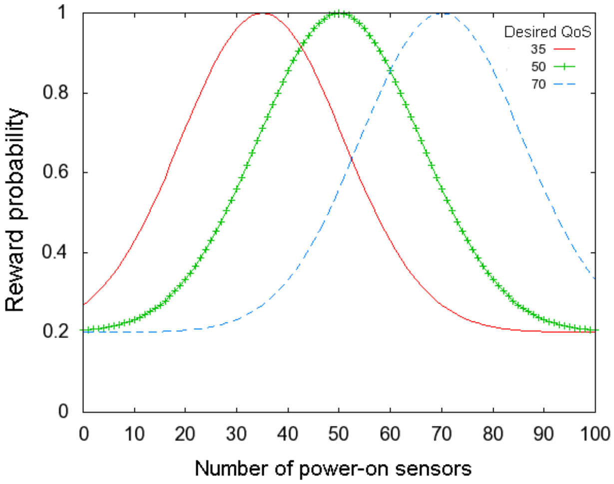


Figure 2. Examples of reward function with K_t =35, 50, and 70

The sink counts the number of received data packets from the active sensors and determines the number of active sensors. Then, the sink uses the information and reward functions to derive the reward value R, and broadcasts R to all sensors. The sensors can then decide whether to be active or idle in the next iteration based on the received R, its finite-state automaton, and the current state. Finally, the Gur Game algorithm can make the number of active sensors to reach the target after a certain number of iterations.

3. THE PROPOSED SCHEME: SHUFFLE

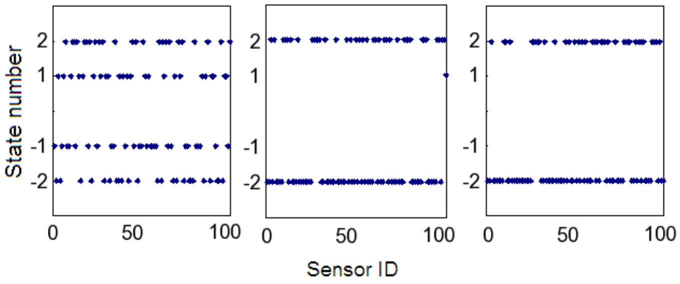
3.1. Problem description

The Gur Game-based scheme initially determines the state of each sensor node randomly. All sensors are uniformly distributed in all states with half of the sensors being active. The number of active sensors approaches desired QoS target after a certain number of runs. Finally, the number of active sensors equals that in the desired QoS, thus making the whole system stable. The probability of transition is equal to one when the desired QoS is achieved, thus keeping sensors in a steady state and maintaining the stability of the system.

Although the goal of QoS control is achieved, a potential problem still exists. All sensors are in a steady state, thus active sensors are always identical. These identical active sensors may expire soon because of energy depletion. By contrast, sleep sensors are always asleep in steady states. This imbalance in power consumption significantly reduces the lifetime of WSNs.

A periodical sleeping mechanism is adopted to solve this problem. However, periodical sleeping is not applicable to the Gur Game-based scheme. Moreover, a centralized scheduling scheme may control sensors quickly and effectively but may encounter scalability problems when the number of sensors increases. Therefore, maintaining the characteristics of the Gur Game scheme and avoiding unbalanced power consumption is our goal.

Figure 3 shows an example of the unbalanced power consumption in Gur Game-based scheme. Figure 3 shows an example of unbalanced power consumption in the Gur Game-based scheme. Figure 3a presents the initial states of all sensors, which are distributed uniformly in four states (-2, -1, +1, and +2). Figure 3b displays the node states at the 200th epoch. Ninety-seven percent of the sensors are in the edge states, that is, states -2 and +2. Figure 3c provides the node states at the 500th epoch, when almost all sensors are in the edge states.



(a) 1st epoch (b) 200th epoch (c) 500th epoch

Figure 3. States of sensors at different epoch

3.2. Proposed scheme: Shuffle

We first thought of exchanging active nodes for sleep nodes to avoid unbalanced power consumption. We considered a simple and easy method in the beginning. The base specifies several nodes to exchange states. However, this method is not suitable when fairness and scalability are concerned.

Moreover, exchanging sensor states may result in unstable systems. In particular, QoS approaches the target number and promotes system stability after a certain period of QoS vibrations. However, a long stable duration also implies unbalanced power consumption. Exchanging sensor states may help balance power consumption; however, this method may disrupt network stability.

We thought of using the Gur Game scheme once more to help the system return to stability after exchanging sensor states. The self-optimization characteristic of the Gur Game scheme enables QoS to return automatically to the desired value after a certain period of QoS vibrations.

Based on this idea, we propose an enhanced QoS control scheme called Shuffle, which periodically applies the Gur Game scheme to maintain network stability after the exchange of sensor states. Shuffle swaps the sensor states located in the two edge states (-2 and +2). In particular, all sensors in the two edge states are swapped. Subsequently, Shuffle uses the Gur Game scheme to help the system return to stability and achieve the desired QoS. Shuffle attempts to modify the Gur Game-based scheme as less as possible and to maintain its characteristic.

The time complexity of the Gur Game scheme is signified by O(one) because the number of nodes does not affect the computation in the reward function. The time complexity of Shuffle is denoted as O(*s*), where *s* is the number of times the sensors are shuffled.

The duration of the exchange of sensor states is an important issue in Shuffle. Stability will not be achieved if sensor states are exchanged too often. By contrast, long sensor state exchanges may lead to unbalanced power consumption. We observe QoS vibration to determine the period of shuffle. The system becomes stable after about 450 epochs of QoS vibration (Figure 4). Thus, we suggest a shuffle period larger than 500 epochs. The shuffle period in Figure 4 is 1000 epochs. The QoS clearly reverts to the desired value (35) after the second QoS vibration at the 1000th epoch. Moreover, Figure 4 shows that frequent shuffles result in system instability in a short period of time.

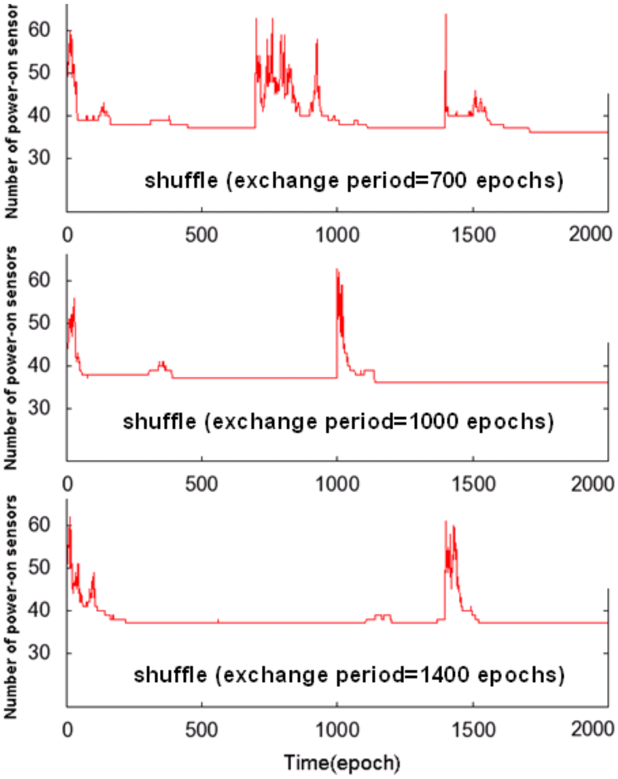


Figure 4. Number of awake sensors of modified scheme with period of 700, 1000, and 1400.

4. RESULTS AND DISCUSSION

The performance of the proposed scheme, Shuffle, in terms of residual energy and lifetime is compared with the Gur Game-based scheme.

4.1. Simulation environment

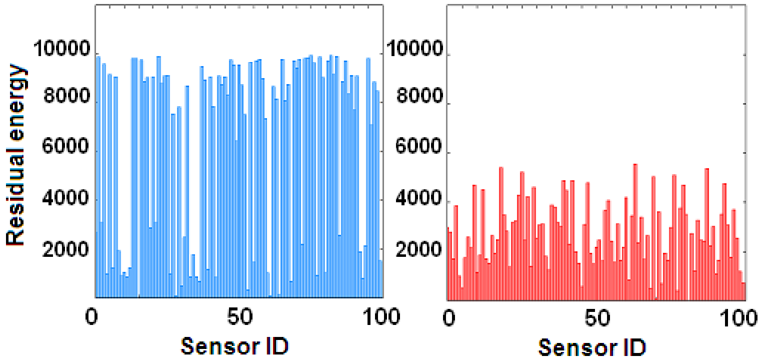
One hundred sensors are randomly deployed in a 100 m × 100 m area with a sink at the center upon simulation. All sensors can hear the broadcast message from the sink. Time is divided into discrete intervals, i.e. epochs. For each epoch, each active sensor can transmit one data to the sink though multi-hops. Sensors do not exchange messages among themselves. Specifically, sensors only transmit to and receive data from sinks.

Since this research does not focus on energy model, we assume a general energy model to evaluate the lifetime of proposed scheme in the simulation. A sensor has an initial 10000 units of battery power. For a power-on sensor, each round takes one unit of battery power. A power-off sensor does not use one unit of battery power in this round. A sensor is considered dead if its battery power is exhausted.

The Gur Game-based scheme is compared with the Shuffle scheme because the former is the most related to our proposed scheme. The simulation model is implemented with Java, and the two measured parameters are defined as follows:
Residual energy: The amount of energy for each sensor when the first sensor is dead.
Network lifetime: A period that begins from network initialization to the first instance of sensor death.

4.2. Simulation results

Simulated residual energy against various sensor IDs is shown in Figure 5, in which the left part is the Gur Game-based scheme, whereas the right part is the Shuffle. On average, the Gur Game-based scheme has more residual energy than the Shuffle (Figure 5). Difference in residual energy values implies that the Gur Game-based scheme cannot conserve energy efficiently. The unbalanced power consumption of the Gur Game-based scheme leaves a large amount of unused energy at the end of network lifetime. The values in the two sections of Figure 5 are all measured at the end of the lifetime. The lifetime of the Gur Game-based scheme is 10142 (when sensor #17 dies), whereas the lifetime of Shuffle is 16923 (when sensor #80 dies). By contrast, residual energy values of the sensors in Shuffle are more balanced and lower compared with the sensors in the Gur Game-based scheme.



(a) gurgame-based scheme (b) shuffle

Figure 5. Residual energy at the end of lifetime

Figure 6 plots network lifetime against total number of sensors. The desired number of power-on sensors in this experiment depends on the total number of sensors. In particular, the ratio of the desired number of power-on sensors to the total number of sensors is fixed. Therefore, if the number of sensors is 100 and 200, then the desired number is 35 and 70, respectively. The exchange period in the simulation is 100 epochs. Figure 6 shows that Shuffle can have longer network lifetime compared with the Gur Game-based scheme. Network lifetime increases with the increasing total number of sensors because more sensors are available to be powered on. Figure 6 also shows that higher frequencies state exchanges (that is, shorter exchange periods) lead to longer network lifetime because frequent exchanges result in more balanced power consumptions.

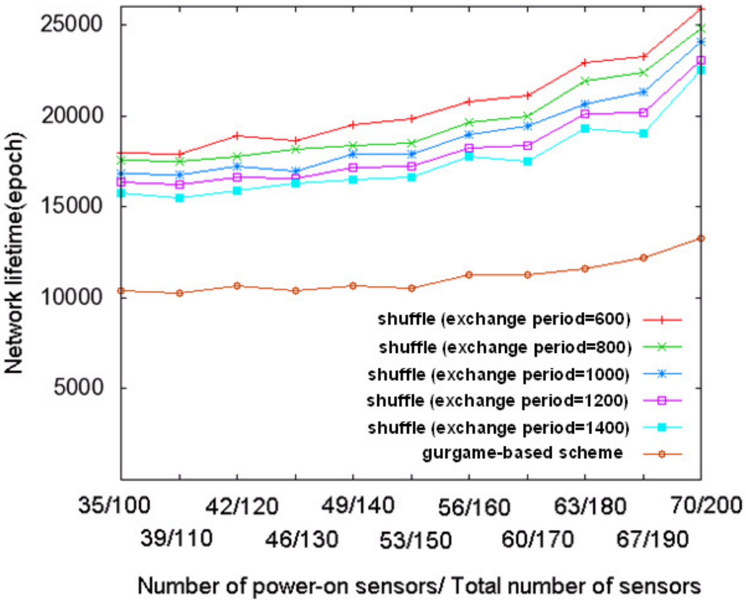


Figure 6. Comparison of lifetime in different number of sensors

Simulation results of network lifetime against the exchange period are shown in Figure 7. The network lifetime of the Gur Game-based scheme is independent from the exchange period; thus, the Gur Game-based network lifetime is close to a horizontal line. The network lifetime of Shuffle decreases with increasing exchange period because frequent exchanges result in more balanced power consumptions. The network lifetime of Shuffle with a very long exchange period is very close to that of the Gur Game-based scheme.

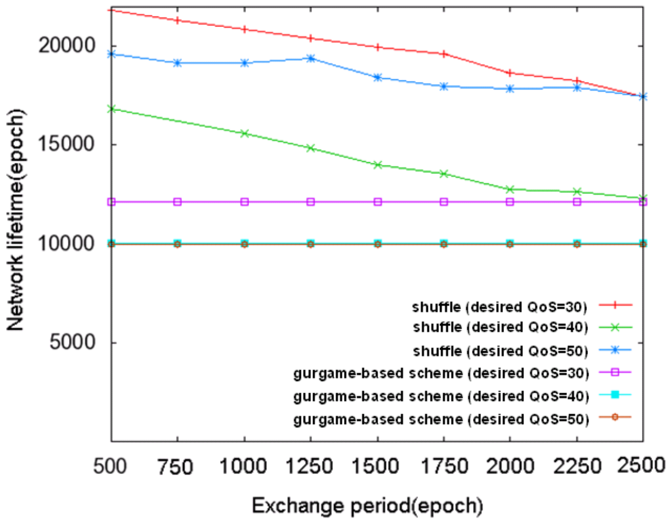


Figure 7. Comparison of lifetime in different period of shuffle

Simulation results shown in Figure 8 are those of network lifetime against desired QoS. Shuffle with different exchange periods has a longer lifetime than the Gur Game-based scheme (Figure 8). The network lifetime of Shuffle reaches its peak when the desired QoS is close to 50. This condition occurs because all sensors are easily devised into two disjoint groups that take turns in working. Moreover, Shuffle with a shorter exchange period has a longer network lifetime.

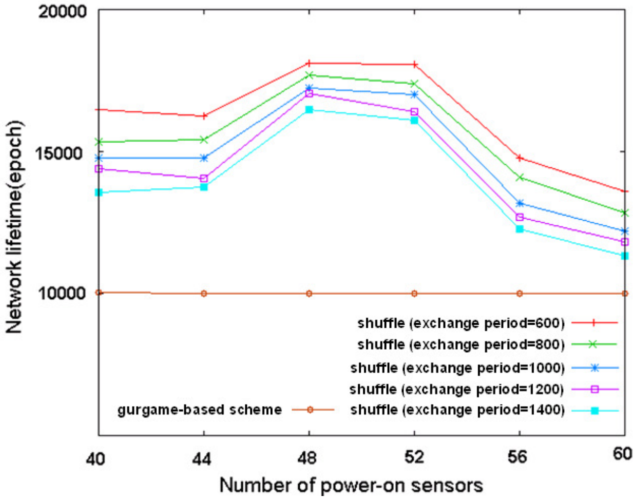


Figure 8. Comparison of lifetime in different desired QoS

All the aforementioned simulations are conducted under a static network, whereas, the following simulations are conducted under a dynamic network. No sensor failures and renewals are observed in the 100 sensors in the static network. By contrast, sensor failures, renewals, and transmission delays are experienced by the 100 sensors in the dynamic network. New sensors are added into the system with exponentially distributed times between births with mean 100 seconds for sensor failures and renewals. All sensors remain alive for an exponentially distributed time with mean 101 seconds. Packet delay for each sensor is uniformly distributed from zero to five epochs [1].

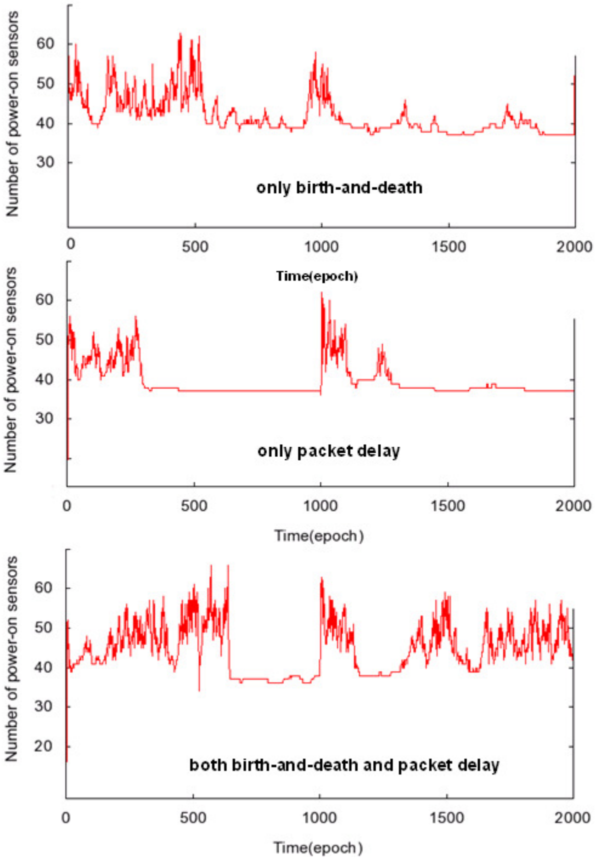


Figure 9. QoS performance against epoch for shuffle in a realistic environment (1)only sensor birth-and-death (2) only packet delay(3) both sensor birth and dead and packet delay

Figure 9 shows a trace of the number of active sensors versus the sample run time of 2000 epochs. Three figures are exhibited in Figure 9. The top figure only presents the results of sensor birth and death. Sensor birth and death leads to longer convergence time. Active sensors may still change after converging because of sensor birth and death. The middle figure shows the results of additional packet delay. Similar to sensor birth-and-death, packet delays lead to a long convergence time. However, active sensors lock once they converged. The bottom figure shows the results of adding sensor birth and death as well as packet delay, which makes the curve more unbalanced.

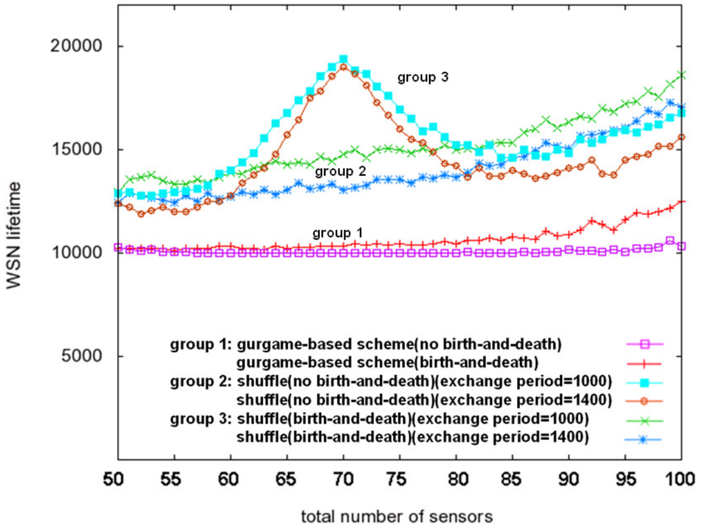


Figure 10. WSN lifetime against total number of sensors for a realistic environment considering sensor birth-and-death

Figure 10 plots WSN lifetime against different sensor numbers in the network with sensor birth and death. Shuffle is shown to have a longer lifetime than the Gur Game-based scheme (Figure 10). We can classify the six curves into three groups: (1) Gur Game-based scheme, (2) proposed scheme without sensor birth and death, and (3) proposed scheme with sensor birth and death.

In the first group, the lifetime of the Gur Game-based scheme with sensor birth and death is very close to that of the Gur Game-based scheme without sensor birth and death. The Gur Game-based scheme is not significantly affected by sensor birth and death, which only change active sensors slightly. The lifetime of the Gur Game-based scheme with sensor birth and death is slightly longer than that of the Gur Game-based scheme without sensor birth and death when the total number of sensors is large. The difference in lifetime is caused by the dynamics of sensor birth and death, which leads to disruptions and node state exchanges between active and sleep sensors.

The second group is Shuffle with different exchange periods (1000 and 1400) and without sensor birth and death. Lifetime reaches its peak at sensor number 70, which is double that of the desired QoS (35). This condition happens because of the easy behavior of the sensor in Shuffle when the desired QoS is half that of the sensor number. Two groups, each with half the number of sensors, take turns in waking up. This process wastes less energy during convergence and results in a longer lifetime. In addition, lifetime is longer with large sensor numbers (85) than with small sensor numbers (55). A longer lifetime is the result of more sensors taking turns in waking up. Furthermore, a smaller shuffle period causes sensors to change states more frequently.

The third group is Shuffle with different exchange periods (1000 and 1400) and with sensor birth and death. No lifetime peak is observed at sensor number 70 in this group because sensor birth and death break the balance of the two groups and cause active sensors to be inactive. Instead, the two curves increase with increasing sensor numbers. Similar to the second group, the third group exhibits larger sensor numbers or lower exchange periods with longer lifetime.

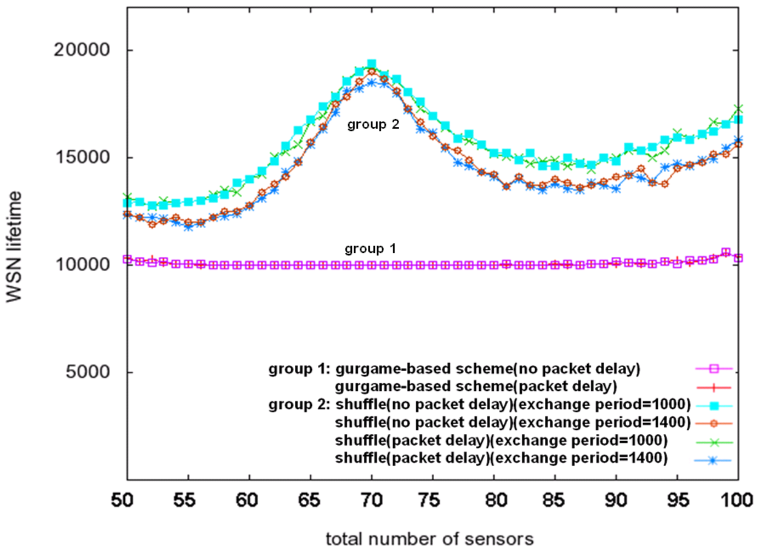


Figure 11. WSN lifetime against total number of sensors for a realistic environment considering sensor transmission delay

Figure 11 plots the WSN lifetime against different sensor numbers in the network with packet delay. Shuffle is observed to have a longer lifetime than Gur Game-based scheme (Figure 11). We can classify the six curves in Figure 11 into two groups: (1) Gur Game-based scheme and (2) Shuffle.

The lifetime of the Gur Game-based scheme is not affected by packet delay. The lifetime of Shuffle is not affected by packet delay either. Similar to the aforementioned results, Shuffle has the highest lifetime when the desired QoS is half of the sensor number. Furthermore, lifetime is longer when the sensor number is large because more sensors can take turns waking up. In addition, shorter exchange periods correspond to longer Shuffle lifetime because power consumption is more balanced.

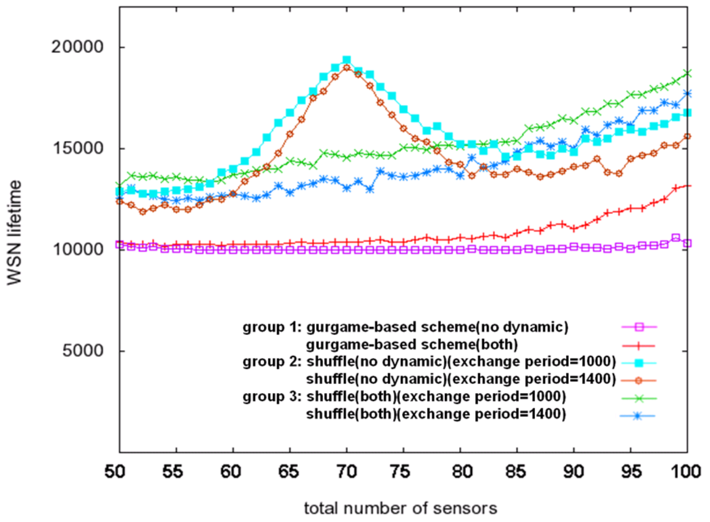


Figure 12. WSN lifetime against total number of sensors for a realistic environment considering both sensor birth-and-death and transmission delay

Simulation results of lifetime against total number of sensors in the network with packet delay, sensor failures, and sensor renewals are shown in Figure 12. Figure 12 shows results similar to Figure 10 except that larger fluctuations are caused by packet delay.

In conclusion, the results for the dynamic environment show that Shuffle lifetime is always larger than that of the Gur Game-based scheme regardless of transmission delay, sensor failure, and sensor renewal.

Based on all simulations, we conclude that Shuffle can effectively prolong lifetime by periodical shuffling regardless of the total number of sensors, desired QoS, and the period of Shuffle.

5. CONCLUSIONS

This paper focuses on the design of a QoS control scheme for WSNs. First, we recognize that sensors move to the edge state in the Gur Game automaton, resulting in a limited lifetime for a prior Gur Game-based scheme. Furthermore, this paper presents an enhanced QoS control scheme, called Shuffle, which balances power consumption and maintains the strength of the Gur Game-based scheme similar to self-optimization. The evaluation of Shuffle in various environments shows that Shuffle significantly improves network lifetime. Further simulation results show that the gains of Shuffle are dependent on the period of shuffles. A short shuffle period achieves a high degree of balance on power consumption, whereas frequent shuffles cause system instability in a short period.

REFERENCES

[1] Iyer, Ranjit, & Kleinrock, Leonard, (2003) "QoS control for sensor networks", IEEE International Conference on Communications(ICC '03).

[2] Nayer, Syed I., & Ali, Hesham H., (2008) "A Dynamic Energy-Aware Algorithm for Self-Optimizing Wireless Sensor Networks", International Workshop on Self-Organizing Systems, Vienna, Austria, pp262-268.

[3] Tsai, Rong-Guei, & Wang, Hao-Li, (2010), "Shuffle: An Enhanced QoS Control by Balancing Energy Consumption in Wireless Sensor Networks", Advances in Grid and Pervasive Computing, LNCS, No.6104, pp603-611.

[4] Tsai, Rong-Guei, & Wang, Hao-Li, (2010), "A Coverage-Aware QoS Control in Wireless Sensor Networks", International Conference on Communications and Mobile Computing (CMC), pp192-196.

[5] Lei, Zhao, Chaonong, Xu, Yongjun, Xu, & Xiaowei, Li, (2006), "Energy-Aware QoS Control for Wireless Sensor Network", IEEE Conference on Industrial Electronics and Applications, pp1-6.

[6] Chen, Dazh, & Varshney, Pramod K., (2004), "QoS Support in Wireless Sensor Networks: A Survey", International Conference on Wireless Networks (ICWN 2004), Las Vegas, Nevada, USA.

[7] Kay, James, & Frolik, Jeff, (2004), "Quality of service analysis and control for wireless sensor networks", IEEE International Conference on Mobile Ad-hoc and Sensor Systems, pp359-368.

[8] Liang, Biyu, Frolik, Jeff, & Wang, X. Sean, (2009), "Energy-Efficient Dynamic Spatial Resolution Control for Wireless Sensor Clusters", International Journal of Distributed Sensor Networks, Vol.5, No.4, pp361-389.

[9] Frolik, Jeff, (2004), "QoS control for random access wireless sensor networks", IEEE International Conference on Wireless Communications and Networking Conference pp1522-1527.

[10] Pereira, Vasco, Silva, Jorge Sa, Granjal, Jorge, Silva, Ricardo, Monteiro, Edmundo, & Pan, Qiang, (2011), "A Taxonomy of Wireless Sensor Networks with QoS", the IFIP International Conference on New Technologies, Mobility and Security, pp1-4.

[11] Bhuyan, Bhaskar, Sarma, Hiren Kumar Deva, Sarma, Nityananda, Kar, Avijit, & Mall, Rajib, (2010), "Quality of Service (QoS) Provisions in Wireless Sensor Networks and Related Challenges", Wireless Sensor Network, Vol.2, No.11, pp861-868.

[12] Bagula, Antoine B., (2010), "Modelling and implementation of QoS in wireless sensor networks: a multiconstrained traffic engineering model", EURASIP Journal Wireless Communication Network, pp1-14.

[13] Changmin, Duan, (2009), "Topology Controlling for QoS in Wireless Sensor Networks", International Conference on Wireless Communications, Networking and Mobile Computing.

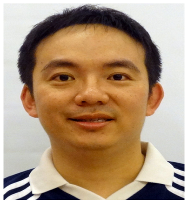
[14] Jiang, Joe-Air, Lin, Tzu-Shiang, Chuang, Cheng-Long, Chen, Chia-Pang, Sun, Chin-Hong, Juang, Jehn-Yih, . . . Liang, Wei-Wen, (2011), "A QoS-Guaranteed Coverage Precedence Routing Algorithm for Wireless Sensor Networks", Sensors, Vol.11, No.4, pp3418-3438.

[15] Nandha kumar.R , P.Varalakshmi, (2012), "An Efficient Data Collection Protocol in Wireless Networks", International Journal of Ad hoc, Sensor & Ubiquitous Computing (IJASUC) Vol.3, No.4.

[16] D.G.Anand, H.G. Chandrakanth, & M.N.Giriprasad, (2012)"An Energy Efficient Distributed Protocol for Ensuring Coverage and Connectivity (E^3C^2) of Wireless Sensor Network", International Journal of Ad hoc, Sensor & Ubiquitous Computing (IJASUC) Vol.3, No.1.

Authors

Hao-Li Wang received his BE and MS from National Cheng Kung University, Taiwan. He received his PhD from Iowa State University, United States. At present, he is a faculty in National Chiayi University, Taiwan. His research interests are wireless sensor networks and wireless LANs.



Rong-Guei Tsai received his MS from National Chiayi University, Taiwan. He is presently pursuing his PhD in National Cheng Kung University, Taiwan. His research interests are Quality of Service control and coverage in wireless sensor networks.