

**Research Article**

**Stochastic Polling Interval Adaptation in Duty-Cycled Wireless Sensor Networks**

_Sungryoul Lee_

_The Attached Institute of Electronics and Telecommunications Research Institute (ETRI), P.O. Box 1, Yuseong, Daejeon 305-600, Republic of Korea_

_Correspondence should be addressed to Sungryoul Lee; srlee0525@ensec.re.kr_

_Received 29 October 2014; Accepted 16 January 2015_

_Academic Editor: Andrei Gurtov_

_Copyright © 2015 Sungryoul Lee. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited._

In past decades, to achieve energy-efficient communication, many MAC protocols have been proposed for wireless sensor networks (WSNs). Particularly, asynchronous MAC protocol based on _low power listening_ (LPL) scheme is very attractive in duty-cycled WSNs: it reduces the energy wasted by idle listening. In LPL scheme, a sensor node wakes up at every polling interval to sample the channel. If the channel is busy, the sensor node will stay in wake-up mode for receiving the data packet. Otherwise, it goes to sleep and saves power. However, wrong choice of polling interval in LPL scheme causes unexpected energy dissipation. This paper focuses on the polling interval adaptation strategy in LPL scheme with the aim of maximizing energy efficiency, defined as the number of packets delivered per energy unit. We propose a novel polling interval adaptation algorithm based on stochastic learning automata, where a sensor node dynamically adjusts its polling interval. Furthermore, our simulation results demonstrate that the polling interval asymptotically converges to the optimal value.

1. **Introduction**

In recent years, wireless sensor networks (WSNs) have been developed and implemented to realize automated home system, combined with consumer electronics technologies [1]. The design principle of such a system is very simple: the low-cost sensor nodes with low-power processing, narrow communication range, and small battery size are scattered in a sensing field. Therefore, it is unreasonable to replace or recharge the batteries of sensor nodes. For example, a real-time intrusion detection system detects the humidity and transmits an alert message to a remote base station, that is, a sink, immediately [2]. In this application, it is desirable to retain for long operational time of sensor node, but the network lifetime is substantially constrained due to the limited capacity of the battery.

Recently, a number of research efforts have been undertaken to save power of sensor node in communication activities. The main direction of these researches is duty-cycling which reduces power wasted through idle listening, that is, the time spent in wake-up mode with receiving any radio packets. Note that idle listening is a dominant reason that drains the energy of sensor node [3–5]. In duty-cycled WSNs, the sensor node alternates between sleep mode and wake-up mode to reduce energy consumption caused by idle listening. The sensor node wakes up in small portion of operating time and it goes to sleep in most of time. The transition between sleep mode and wake-up mode is decided by a predefined schedule. In sleep mode, the sensor node turns off its radio to save power. In wake-up mode, the sensor node turns on its radio to communicate with its neighbors. If the sensor node has a packet destined to its neighbors or it is an intended receiver, it transmits or receives a data packet in wake-up mode.

_Low power listening_ (LPL) scheme has motivated recent advances in asynchronous MAC protocol design for duty-cycled WSNs. In [6, 7], the authors introduced the concept of LPL scheme in a perspective of duty-cycled WSNs to reduce the idle listening, where a sender transmits a _C_-sized preamble to a receiver which is supposed to wake up at every polling interval _C_ for preamble detection. In B-MAC [6], a sender transmits a long preamble to make its neighbors wake up before transmitting a data packet. Upon detecting the preamble, all neighbors keep wake-up mode until the transmission
of data packet has been finished. However, B-MAC inherently suffers from the excessive long preamble accompanied with the transmission of data packet and the overhearing of nonintended receivers. X-MAC [7] has been proposed as an enhanced LPL scheme fixing the drawbacks of B-MAC. X-MAC adopts a series of short preambles to avoid overhearing. In X-MAC, a sender transmits a short preamble and waits for a response from a receiver for a short time as shown in Figure 1. And the sensor node samples the channel when it wakes up. If it is the intended receiver, upon receiving the short preamble, it replies with ACK to trigger the transmission of data packet. Otherwise, it goes to sleep, to avoid overhearing. As compared with synchronous MAC protocols such as S-MAC [3] and T-MAC [4], these asynchronous MAC protocols have the advantage of not requiring any time synchronization among sensor nodes.

In LPL scheme, the best performance can be obtained by appropriately selecting the polling interval \( C \) which is a key parameter in determining energy used to communicate between sensor nodes. The optimal polling interval heavily depends on traffic loads of the sensor node. In low traffic loads, a large polling interval takes advantage of saving energy of sensor node, while a relatively small polling interval consumes less energy spent for communication between sensor nodes in high traffic loads. Also, the polling interval must be adapted during runtime since traffic condition may not be known a priori [8]. If a priori assumption about traffic loads is not accurate, energy of sensor node will be wasted and the intended goal of application cannot be achieved. For instance, electronic security system which is installed in home detects suspicious objects. To achieve this requirement, sensor nodes should be in sleep mode for most of the time when there are no events of interest.

Recently, there have been a few studies on polling interval adaptation in duty-cycled WSNs. The representative technique employed by existing polling interval adaptation schemes such as [9, 10] guesses the traffic condition based on previous results of channel sampling. Note that this approach is referred to as Dynamic LPL (DLPL) scheme. In DLPL scheme, the sensor node obtains information about the number of consecutive busy (idle) sampling and dynamically adjusts the polling interval by adopting the following. The polling interval is increased after \( U \) idle polling. Similarly, \( D \) consecutive busy pollings induce a decrease of polling interval. This approach is simple and intuitive but has some problems. Most importantly, it can make a scheme either too aggressive or too conservative depending on \( U \) and \( D \). For example, BoostMAC [9] which employs \( U = 1 \) and \( D = 1 \) reacts well to traffic changes, as it immediately increases/decreases the polling interval. But it introduces unnecessary polling interval fluctuations in stable environments, which causes deterioration of performance. Additionally, it cannot be expected to converge to optimal polling interval.

In this paper, we propose a stochastic polling interval adaptation algorithm based on learning automata technique, where a sensor node dynamically adjusts its polling interval according to the probability distribution of selecting the polling intervals and selects the optimal polling interval by learning. In the proposed algorithm, we adopt energy efficiency, which is defined as the number of packets delivered per energy unit, to update the probability distribution associated with the polling intervals. Additionally, our evaluations verify that the proposed algorithm makes the polling interval converge to the optimal value.

Our main contributions are summarized as follows.

(i) We consider the problem of runtime adaptation of polling interval in duty-cycled WSNs. Fixed polling interval may degrade the performance of LPL scheme when traffic loads are not known a priori.

(ii) We investigate the problem of adapting the polling interval via stochastic learning automata. To the best of our knowledge, our work is the first attempt to provide a learning automation based solution for adaptation of polling interval in duty-cycled WSNs.

(iii) We propose a novel polling interval adaptation algorithm which dynamically adjusts the polling interval, and the polling interval asymptotically converges to the optimal value. Our simulation results show that the proposed algorithm converges toward the optimal value.
The reminder of this paper is organized as follows: In Section 2, we present a review of some previous related work. Section 3 describes the proposed polling interval adaptation algorithm based on stochastic learning automata. Section 4 discusses the simulation results, and we conclude this paper in Section 5.

2. Related Work

In this section, we overview current researches on MAC protocol design in duty-cycled WSNs.

Basic LPL scheme, that is, B-MAC [6] which uses a long preamble to establish rendezvous between a sender and its receiver, has overhearing problem. In B-MAC, since the sender does not indicate the intended receiver of the packet, all neighbor nodes must wait for finishing the long preamble. To solve this problem, several works [7, 11, 12] adopt an aggressive short preamble to replace the long preamble, which divides into a series of short packets. In place of the long preamble, SpeckMAC-B [11] uses the wake-up packet, and B-MAC+ [12] sends the chunks which contain information about remaining chunks before transmitting the data packet. Note that the authors of [12] also presented an extension of B-MAC+ that adapts the polling interval of the transceiver to the traffic loads experienced by different sensors [13]. Therefore, the sensor node which receives the early preamble can sleep during waiting for the data packet. Other approaches [7, 14] reply with an early ACK for stopping the transmission of excessive short preamble. The sender transmits a series of short preambles including ID of intended receiver. ACK sent in response to the short preamble triggers the transmission of data packet.

In duty-cycled WSNs, static approach which adopts a fixed polling interval cannot be adapted to various network conditions. To overcome this problem, several works [9, 13, 15–20] allow dynamically changing sensor’s polling interval. The role of these polling interval adaptation mechanism is to select the optimal polling interval according to traffic loads of sensor node. PMAC [15] adopts an adaptive duty-cycled scheme, instead of having a fixed duty cycle as in S-MAC, to improve energy efficiency. PMAC allows the sensor node to adaptively determine the sleep-wake up schedules based on its own traffic and the traffic patterns of its neighbors. Dynamic LPL (DLPL) scheme [9, 16] is a widely adopted and well-known polling interval adaptation algorithm. This works as follows: if \( U \) consecutive samplings are idle, the sensor node increases its polling interval; if \( D \) consecutive busy samplings are busy, the polling interval is increased. In BoostMAC [9], changes of polling interval in sensor node are accomplished by using AIMD (Additive Increase/Multiplicative Decrease) mechanism in response to the results of channel sampling. Also, in [16], the authors presented a Markov model which evaluates the performance of DLPL scheme in terms of energy consumption. The analytical model enables us to investigate the effect of the up/down threshold (\( U \) and \( D \))—these parameters determine how long a sensor node should stay at a certain polling interval before it concludes the traffic condition has changed—on the performance of DLPL scheme. Obviously, DLPL scheme is easy to deploy, but it may produce unnecessary fluctuations or cannot react quickly to traffic changes according to up/down threshold. Meanwhile, the queue state is a useful information for guessing the network condition implicitly. In [17, 18], the authors proposed an adaptive control mechanism based on the queue management where the controller changes the polling interval dynamically by constraining the queue length. Similarly, TA-MAC [19] adjusts sleep interval adaptively according to state of sending/receiving buffer, traffic loads, and battery lifetime. In [20], the authors presented a cross-layer design approach for joint optimization at the MAC and routing layers. To address this challenge, they proposed an adaptation of listening modes according to local state of each sensor, and it enables the sensor node to learn listening mode of its neighbors in order to ensure correct data delivery.

Learning automata have been applied to study a wide range of solving optimization problems in wireless networks. In [21], the authors adopted a stochastic learning automata model to find the optimal channel selection for secondary users in cognitive radio networks. Since primary user’s traffic patterns are unknown and unpredictable, they claim that the secondary users must select the statistically optimal channel which maximizes the probability of successful transmission and propose an estimator automata model to pursue global optimum with minimal number of iterations. In [22], the authors proposed a novel congestion control algorithm based on learning automata in healthcare WSNs. The primary objective of this approach is that the processing rate in sensor node is equivalent to the transmitting rate. In each sensor node, better data rate may be chosen on the basis of past experience with congestion with the other data rates. In [23], the authors adjusted the threshold parameters of Auto Rate Fallback (ARF) in IEEE 802.11 WLANs using learning automata.

3. Stochastic Polling Interval Adaptation

3.1. Learning Automata. At the beginning, learning automata techniques were introduced to find a solution in control literatures. Recently, they have been recognized as one of the most powerful methods to select the best action in a stochastic environment. And they have been adopted to solve the stochastic optimal control problems in a wide range of research fields.

The purpose of stochastic learning automata which keep track of possible actions and their probabilities is to maximize the expected reward or to minimize the expected penalty based on the response of possible actions. In learning process, an action from finite set of possible actions is applied to a stochastic environment, and then learning system records a response associated with an action as depicted in Figure 2. The response reflects the condition of the stochastic environment. Learning can be described as follows.

Let us denote the finite set of possible actions as \( \alpha = [\alpha_1(t), \alpha_2(t), \ldots, \alpha_n(t)] \) at time \( t \) and selection probabilities of their actions as \( P = [P_1(t), P_2(t), \ldots, P_n(t)] \) at time \( t \), respectively. And let \( \beta = [\beta_1(t), \beta_2(t), \ldots, \beta_n(t)] \) be the set of automation output at time \( t \). At every time iteration, a
stochastic environment takes the set of action \( \alpha \) as input and generates the output vector \( P \) related to input action based on the response, \( \beta \). Therefore, probability distribution over actions is updated based on the response of the environment and is reinforced to select the optimal action. This process continues until a predefined condition.

As previously mentioned, a wireless sensor network is a part of stochastic environments where traffic loads are not known a priori. In our work, we formulate the problem of polling interval adaptation as an environment that stochastic learning automata select the best action.

3.2. The Proposed Algorithm. Motivated by the above discussion, we devise a control algorithm of maximizing the expected reward for selecting the polling interval \( C \).

Now let us describe the basic operation of our proposed algorithm, which aims to converge to the optimal action, that is, selecting the optimal polling interval. We consider that a sensor node selects the polling interval in \( K \) polling intervals, \( C_1 < C_2 < \cdots < C_K \). And let us denote actio action probability vector \( P \) associated with each polling interval. The sensor node adopts LPL scheme with X-MAC [7]. Additionally, when a sender has a data packet to be sent, it transmits a series of short preambles longer than polling interval \( C_K \) to ensure asynchronous rendezvous between the sender and its receiver. Since the receiver wakes up every polling interval and samples the channel, it detects the short preamble. If the receiver is the intended recipient, it replies with ACK. And the reception of ACK at the sender triggers the transmission of data packet. Otherwise, the receiver goes into sleep and waits until the next polling interval to sample the channel. Note that the operation of X-MAC is illustrated in Figure 1.

Remark 1. Our algorithm operates at receiver-side. If the sensor which acts as the sender has the data packet to transmit regardless of its operation mode (sleep or wake-up mode), it operates in wake-up mode immediately and transmits a series of short preambles to check whether an intended receiver wakes up. Therefore, when the sensor acts as the sender, it does not affect our algorithm.

The sensor node selects the polling interval in each decision time. We denote by \( t = [t_0, t_1, \ldots] \) and \( t(i) = [t_{i-1}, t_i], i = 1, 2, \ldots \), the set of decision time and the length of \( i \)th iteration of our algorithm, respectively. In the following, we assume that there is enough time between consecutive decision times, that is, \( t(i) = [t_{i-1}, t_i], i = 1, 2, \ldots \), to get the correct answer. This means that the sensor node needs enough time to receive one or more packets. At time \( t \), the sensor node selects the polling interval \( C_i, i = 1 \leq \cdots \leq K \), based on the probability vector \( P \). Note that, during \( i \)th iteration, polling interval is not changed. The polling intervals are taken as input to stochastic learning automation, and the sensor node updates the probabilities of polling intervals as a function of the output given by a stochastic environment. In our model, we adopt the energy efficiency, defined as the ratio of total amount of packets delivered to total energy consumption, as output metric. This process continues until stopping condition.

In our learning model, the following measures are considered: \( N_i, i = 1 \leq \cdots \leq K \), as the number of received data packets with polling interval \( C_i \) in each iteration, and \( E_i, i = 1 \leq \cdots \leq K \), as power consumption with polling interval \( C_i \) in each iteration, respectively. \( E_i \) is obtained as follows (the parameters are shown in Table 1):

\[
E_i = E_s + E_r = \left( P_{tx} S_p + P_{rx} S_a \right) \left( \frac{C_i + R_l}{2 (S_p + S_a)} \right) + P_{tx} S_d + P_{sleep} C_i + P_{rx} R_l + P_{tx} R_d + P_{rx} R_{dp}
\]

where \( E_s \) and \( E_r \) indicate power consumption at the sender and the receiver, respectively. Note that if the channel is idle, the sensor (receiver) considers only \( E_r \) as \( E_s \). Hence, the number of data packets delivered per energy unit, \( X_i = N_i/E_i, i = 1 \leq \cdots \leq K \), can be calculated. At every decision time, the sensor node calculates energy efficiency, \( X_i \), and updates the accumulated energy efficiency. Our algorithm finds the polling interval in terms of maximizing energy efficiency at decision time.

We define \( D_i(t), i = 1 \leq \cdots \leq K \), which is the deterministic estimation vector of polling interval \( C_i \) at time \( t, i = 1 \leq \cdots \leq K \), would be

\[
D_i(t) = \frac{J_i(t)}{H_i(t)}.
\]

| Symbol | Meaning | Value |
|--------|---------|-------|
| \( P_{tx} \) | Power in transmitting | 52.2 mW |
| \( P_{rx} \) | Power in receiving or listening | 56.4 mW |
| \( P_{sleep} \) | Power in sleeping | 3 \( \mu \)W |
| \( c_{sl} \) | Average carrier sense time | 2 ms |
| \( t_B \) | Time to transmit or receive a byte | 32 \( \mu \)s |
| \( L_{data} \) | Data packet length | 50 B |
| \( L_{preamble} \) | Preamble packet length | 15 B |
| \( L_{ack} \) | ACK packet length | 11 B |
| \( S_p \) | Time to transmit the preamble at the sender | 0.48 ms |
| \( S_a \) | Time to listen the ACK at the sender | 0.352 ms |
| \( S_d \) | Time to transmit the data at the sender | 1.6 ms |
| \( R_l \) | Wake-up time at the receiver | 10 ms |
| \( R_d \) | Time to transmit the ACK at the receiver | 0.352 ms |
| \( R_d \) | Time to receive the data at the receiver | 1.6 ms |
where \( H_i(t), i = 1 \leq \cdots \leq K\), is the count of how many times polling interval \( C_i\) has been selected up to time \( t \) and \( J_i(t), i = 1 \leq \cdots \leq K\), is the accumulated energy efficiency with polling interval \( C_i\) up to time \( t \), respectively.

Next, let \( U_i(t) \) denote the stochastic estimator vector at time \( t \), \( i = 1 \leq \cdots \leq K\), which means the reward probability of polling interval \( C_i\):

\[
U_i(t) = D_i(t) + R_i(t),
\]

where \( R_i(t), i = 1 \leq \cdots \leq K\), is a random number which is uniformly distributed in the interval \([-\gamma/H_i(t), +\gamma/H_i(t)]\), where \( \gamma \) is a perturbation system parameter set by the sensor node. Other parameters of our algorithm are defined as follows:

- \( P_i(t) \): the probability of selecting polling interval \( C_i\) at time \( t \), \( i = 1 \leq \cdots \leq K\);
- \( R \): the resolution parameter of learning automaton, that is, a positive number, which has property that it determines the stepsize on the basis of probability vector \( P \);
- \( F \): the predefined convergence threshold;
- \( M \): the maximum value of energy efficiency.

In the following, we develop the proposed algorithm based on stochastic learning automata. According to our proposed algorithm, the sensor node processes its observation and updates probabilities of selecting the polling intervals based on the response before selecting the new polling interval. Indeed, in our algorithm, the probability of selecting the optimal polling interval is increased while the probabilities of others are decreased. Additionally, our algorithm achieves asymptotic convergence.

**Algorithm 2.**

1. **Initialization**
   (i): Set \( P(0) = [P_1(0), \ldots, P_K(0)] \) where \( P_i(0) = 1/K, i = 1 \leq \cdots \leq K \).
   (ii): Initialize \( H_i(0)\), \( J_i(0)\), \( D_i(0)\), \( U_i(0)\), and \( M \) to zeros for \( i = 1 \leq \cdots \leq K \).
   (iii): Select the polling interval \( C_m \) according to \( P(0) \).
   (iv): Set \( N_n \) and \( E_n \) to zeros.
   (v): Start to sample the channel with \( C_n \).
   (vi): Maintain the current polling interval \( C_n \) and then record \( N_n \) and \( E_n \) until \( t(1) \).

2. **At time \( t \)**
   (i): Compute \( X_n \) and update the maximum of energy efficiency, \( M \):
   \[
   \text{if } X_n > M, \quad M = X_n.
   \]
   (ii): Update the accumulated energy efficiency, \( J_n(t) \) as \( J_n(t) = J_n(t) + (X_n/M) \).
   (iii): Update \( H_n(t) \) by adding one as \( H_n(t) = H_n(t) + 1 \).
   (iv): Compute \( n \)th element in the deterministic estimation vector \( D_n(t) \) by setting \( D_n(t) = J_n(t)/H_n(t) \).
   (v): For \( i = 1 \leq \cdots \leq K \), compute the stochastic estimator vector \( U_i(t) \) by setting \( U_i(t) = D_i(t) + R_i(t) \).
   (vi): Find the optimal polling interval \( C_m \) which is the highest value in \( U(t) \).
   (vii): Update probability vector \( P \) as follows:
   \[
P_i(t) = \max \left( P_i(t) - \left( \frac{1}{RK} \right), 0 \right), \quad \text{if } i \neq m, \quad i = 1, \ldots, K,
   \]
   \[
P_m(t) = \sum_{i \neq m} P_i(t), \quad \text{if } i = m.
   \]
   (viii): Set \( N_n, E_n \), and \( X_n \) to zeros.
   (ix): If \( P_m(t) > F \), then converge to optimal polling interval \( C_m \) and stop.
   (x): Otherwise, select the new polling interval according to \( P(t) \) and start to sample the channel for communication.

In our algorithm, the update of probability vector \( P \) depends on the deterministic reward vector \( D \) and the random number \( R \). Initially, during a few iterations, the sensor node selects the polling interval mainly depending on random number \( R \). This implies that all of polling intervals have the chance to be selected as the optimal value. Note that like other learning automata based optimization methods, the computational complexity of our algorithm depends on the number of possible actions, \( K \). Therefore, our algorithm requires time complexity \( O(K) \) for updating the probability vector \( P \). Also, with increasing iteration, our algorithm mainly depends on the deterministic reward. Therefore, the polling interval with high probability is selected more frequently and will be the optimal value. Moreover, the asymptotic process of our proposed algorithm is \( \epsilon \)-optimal which is proved in [24].

**Theorem 3.** The proposed polling interval adaptation algorithm is \( \epsilon \)-optimal for stationary duty-cycled WSNs. For any arbitrarily small \( \epsilon > 0 \) and \( \gamma > 0 \), there exists a \( t' \) satisfying

\[
Pr \left[ |1 - P_m| < \epsilon \right] > 1 - \gamma, \quad \forall t > t',
\]

where \( m \) is the index of optimal polling interval in terms of energy efficiency.

**4. Simulation Results**

In order to evaluate the effectiveness of our proposed algorithm, we performed extensive simulation experiments. We employ a single sender-receiver pair in order to monitor receiver’s polling interval according to its variable traffic load defined by node topological distance from the sink. Note that the sensor nodes near the sink have more traffic loads and shorter polling interval than those far away from the sink. The results of simulation showed that our algorithm dynamically...
adjusts the polling interval according to probabilities of polling intervals to adopt traffic condition and then tracks the optimal polling interval.

In the simulation, we assume that the sensor node has seven polling intervals \( K = 7 \): that is, \( [C_1, C_2, \ldots, C_7] = 20, 40, 80, 160, 320, 640, 1280 \) (msec). Also, let us assume that the sender generates the data packets following the Poisson process with rate one (1 packet/sec). The resolution parameter, \( R \), is set by 2, and the convergence threshold, \( F \), is set by 0.99. Also, 1 and 10 (sec) are used for \( \gamma \) and the length of \( t(i) \). The parameters used in simulations are summarized in Table 1.

In Figure 3, we show the trajectory of polling interval with respect to the optimal value at run time. It illustrates that the proposed algorithm adjusts the polling interval. In initial period, the polling interval is fluctuated. Also, our algorithm accommodates the fluctuation, and the polling interval 4 (160 msec) is more often selected than other polling intervals at around the 20th iteration. Note that the polling interval 4 (160 msec) is the optimal value where it achieves minimal energy consumption in simulation experiments. Correspondingly, it can be seen that the polling interval converges within 50 iterations as depicted in Figure 3.

In the proposed algorithm, the probability of selecting a polling interval is updated to search the optimal value. In Figure 4, the seven curves represent the probability of each polling interval and each point in the curve results from each iteration at runtime. In initial period, the probabilities of polling intervals are equal and selecting polling interval is dependent on randomness. As shown in Figure 4, our proposed algorithm increases the choice probability of selecting the optimal polling interval, that is, polling interval 4 (160 msec), at every iteration, and achieves convergence towards the optimal polling interval.

Furthermore, Figure 5 compares the proposed algorithm with DLPL scheme. Here as well, we use fixed up/down threshold \( (U,D) = (1,1) \) which is used in BoostMAC [9], AMAC [10], and PMAC [15]. From the figure, we can see that the proposed scheme is more energy efficient than DLPL scheme. This is because the proposed scheme converges to the optimal polling interval with probability one, while the polling interval of DLPL scheme is fluctuated.

5. Conclusion

To achieve energy efficient communication in duty-cycled WSNs, one of major issues of MAC protocol design is dynamic adaptation of polling interval against network conditions. Additionally, control algorithm is adapted at runtime in response to local observation for each polling interval.

In this paper, we proposed a novel stochastic polling interval adaptation algorithm to tackle this issue. To the best of our knowledge, this paper is the first attempt to apply stochastic learning automata for control of polling interval in practice. In our algorithm, the sensor node dynamically
adjests its polling interval based on response which is the number of packets delivered per energy unit. Using simulation experiments, we observe that our proposed algorithm can adjust the polling interval to converge to the optimal value gradually.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” IEEE Communications Magazine, vol. 40, no. 8, pp. 102–114, 2002.

[2] G. Song, Z. Wei, W. Zhang, and A. Song, “Design of a networked monitoring system for home automation,” IEEE Transactions on Consumer Electronics, vol. 53, no. 3, pp. 933–937, 2007.

[3] W. Ye, J. Heidemann, and D. Estrin, “An energy-efficient MAC protocol for wireless sensor networks,” in Proceedings of the IEEE INFOCOM, pp. 1567–1576, June 2002.

[4] T. van Dam and K. Langendoen, “An adaptive energy-efficient MAC protocol for wireless sensor networks,” in Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys’03), pp. 171–180, Association for Computing Machinery, November 2003.

[5] Q. Yu, C. Tan, and H. Zhou, “A low-latency MAC protocol for wireless sensor networks,” in Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM ’07), pp. 2816–2820, IEEE, September 2007.

[6] J. Polastre, J. Hill, and D. Culler, “Versatile low power media access for wireless sensor networks,” in Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys ’04), pp. 95–107, November 2004.

[7] M. Buettner, G. V. Yee, E. Anderson, and R. Han, “X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks,” in Proceedings of the 4th ACM International Conference on Embedded Networked Sensor Systems (SenSys ’06), pp. 307–320, November 2006.

[8] Y. Zhang, N. Meratnia, and P. Havinga, “Outlier detection techniques for wireless sensor networks: a survey,” IEEE Communications Surveys and Tutorials, vol. 12, no. 2, pp. 159–170, 2010.

[9] K. Stone and M. Colagrosso, “Efficient duty cycling through prediction and sampling in wireless sensor networks,” Wireless Communications and Mobile Computing, vol. 7, no. 9, pp. 1087–1102, 2007.

[10] S. H. Lee, J. H. Park, and L. Choi, “AMAC: traffic-adaptive sensor network MAC protocol through variable duty-cycle operations,” in Proceedings of the IEEE International Conference on Communications (ICC ’07), pp. 3259–3264, June 2007.

[11] K. J. Wong and D. Arvind, “SpeckMAC: low-power decentralised MAC protocol low data rate transmissions in specknets,” in Proceedings of the 2nd IEEE International Workshop Multi-Hop Ad Hoc Networks: From Theory to Reality (REAL-MAN ’06), May 2006.

[12] M. Avvenuti, P. Corsini, P. Masci, and A. Vecchio, “Increasing the efficiency of preamble sampling protocols for wireless sensor networks,” in Proceedings of the Mobile Computing and Wireless Communications International Conference, pp. 117–122, September 2006.

[13] M. Avvenuti and A. Vecchio, “Adaptability in the B-MAC+ protocol,” in Proceedings of the International Symposium on Parallel and Distributed Processing with Applications (ISPA ’08), pp. 946–951, Sydney, Australia, December 2008.

[14] S. Mahleraknacht and M. Bock, “CSMA-MPS: a minimum preamble sampling MAC protocol for low power wireless sensor networks,” in Proceedings of the IEEE International Workshop on Factory Communication Systems (WFCS ’04), pp. 73–80, Vienna, Austria, September 2004.

[15] T. Zheng, S. Radhakrishnan, and V. S. Sarangan, “PMAC: an adaptive energy-efficient MAC protocol for Wireless Sensor Networks,” in Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS ’05), April 2005.

[16] S. Lee, J. Choi, J. Na, and C.-K. Kim, “Analysis of dynamic low power listening schemes in wireless sensor networks,” IEEE Communications Letters, vol. 13, no. 1, pp. 43–45, 2009.

[17] H. Byun and J. So, “Queue-based adaptive duty cycle control for wireless sensor networks,” in Algorithms and Architectures for Parallel Processing: Proceedings of the 11th International Conference, ICA300 2011, Melbourne, Australia, October 24–26, 2011, Part II, vol. 7017 of Lecture Notes in Computer Science, pp. 205–214, Springer, Berlin, Germany, 2011.

[18] H. Byun and J. Yu, “Adaptive duty cycle control with queue management in wireless sensor networks,” IEEE Transactions on Mobile Computing, vol. 12, no. 6, pp. 1214–1224, 2013.

[19] T.-H. Hsu, T.-H. Kim, C.-C. Chen, and J.-S. Wu, “A dynamic traffic-aware duty cycle adjustment MAC protocol for energy conserving in wireless sensor networks,” International Journal of Distributed Sensor Networks, vol. 2012, Article ID 790131, 10 pages, 2012.

[20] R. Jurndak, P. Baldi, and C. V. Lopes, “Adaptive low power listening for wireless sensor networks,” IEEE Transactions on Mobile Computing, vol. 6, no. 8, pp. 988–1004, 2007.

[21] Y. Song, Y. Fang, and Y. Zhang, “Stochastic channel selection in cognitive radio networks,” in Proceedings of the 50th Annual IEEE Global Telecommunications Conference (GLOBECOM ’07), pp. 4878–4882, November 2007.

[22] S. Misra, V. Tiwari, and M. S. Obaidat, “LACAS: learning automata-based congestion avoidance scheme for healthcare wireless sensor networks,” IEEE Journal on Selected Areas in Communications, vol. 27, no. 4, pp. 466–479, 2009.

[23] Y. Song, X. Zhu, Y. Fang, and H. Zhang, “Threshold optimization for rate adaptation algorithms in IEEE 802.11 WLANs,” IEEE Transactions on Wireless Communications, vol. 9, no. 1, pp. 318–327, 2010.

[24] G. I. Papadimitriou, M. Sklira, and A. S. Pomportsis, “A new class of e-optimal learning automata,” IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 34, no. 1, pp. 246–254, 2004.