

ADP: An Adaptive Feedback Approach for Energy-Efficient Wireless Sensor Networks

Afraa Attiah^{*}, Muhammad Faisal Amjad[†], Omar Nakhila^{*} and Cliff Zou^{*}

^{*}Department of EECS, University of Central Florida, Florida, USA

[†]National University of Sciences and Technology, Islamabad, Pakistan

Email: ^{*}{afraa, ohachum, czou}@eecs.ucf.edu, [†]faisal@nust.edu.pk

Abstract—A broad range of applications has led to various wireless sensor networks (WSNs) with different design considerations. Limited battery power is one of the most challenging aspects of WSN protocol design, and, therefore, energy efficiency has long been the focus of research. One of the most common approaches for energy conservation is to alternate each sensor node between sleep and wake-up states. In this paper, we propose ADP, an adaptive energy efficient approach that meets the requirement of low energy consumption and, at the same time, considers the underlying dynamic traffic load. ADP enhances energy efficiency by dynamically adjusting sensor nodes' sleep and wake-up cycles. ADP utilizes a cost function intended to strike a balance between the conflicting goals of conserving energy (waking up as rarely as possible) and at the same time minimizing sensed events' reporting latency (waking up as frequently as possible). It also incorporates a feedback mechanism that constantly monitors residual energy level and the importance of the event to be reported, as well as predicts the next sensing event occurrence time. Simulation experiments with different traffic loads have shown that ADP improves energy efficiency while keeping latency low.

I. INTRODUCTION

Wireless Sensor Networks (WSNs) are prevalent in many fields, including but not limited to civil engineering, medical, environment, and industrial automation. The design of a wireless sensor network depends specifically upon the application for which it is being deployed. Among all others, energy efficiency is regarded as one of the most critical concern because of the limited battery life of the sensor nodes. Most of the recent studies have focused on how to maximize the lifetime of the system without sacrificing other factors such as latency and throughput. In this paper, we focus on extending the sensor node's lifetime by saving on energy consumption and keeping latency low. In wireless sensor network applications, the underlying sensing traffic load can vary during different times of the day, and each node may have a varied density of participating in activities based on its location in the system, e.g., whether the node is close to the sink or not. Consequently, different sensor nodes have different levels of energy consumption during sensing and communication. The level of energy consumption is one of the most important factors for sensor networks because sensor nodes usually have very limited battery capacity. In addition, replacing the sensor's battery is a maintenance nightmare, and

in some applications replacing the battery is impossible [1]. Dissipation of sensor energy results in quickly diminishing the network lifetime and can affect network performance.

Sensor sleeping is an effective technique to prolong a network lifetime by reducing idle listening. The technique to schedule a sensor's sleep and wake-up cycle can be used in any level of the protocol stack, such as the application layer, routing layer, and medium-access control layer. In sleep mode, a node turns off the radio and goes to sleep in order to save energy instead of staying idle. Control of a sensor waking up can be internal or external [2]. In this paper, we follow the internally controlled wake-up policy, wherein the node periodically wakes up (duty cycling).

Optimization is a sufficient method that can be dynamic or static. Both of the optimization methods assist designers in meeting the application requirements: the static remains fixed for the WSN's lifetime and is appropriate for stable environmental events. However, dynamic optimization is appropriate for changing application requirements and real environmental events; in addition, dynamic methods provide more flexibility and accuracy [3]. The current approaches to sleep and wake-up scheduling are mostly static, i.e., a node always wakes up after a fixed sleep time. This static approach is not efficient since it does not consider the dynamic occurrence behaviors of most underlying sensing events. In addition, a static approach does not consider the critical factor of the remaining energy resource of a sensor node. It is intuitively clear that when a sensor node has less and less remaining battery, it should be more cautious and conservative in waking up to report sensing data in order to prolong its lifetime. Therefore, a more energy-efficient sleep and wake-up scheduling scheme should be an intelligent and dynamic approach.

In this paper, we introduce a dynamic and adaptive data sensing scheduling approach for each sensor node wake-up/sleep time called ADP (an adaptive energy efficient approach). It aims to adjust the optimal sleeping period of each sensor node adaptively according to three feedback factors: the prediction of the next occurrence time of an underlying sensing event, the sensor node's residual battery, and the importance of reporting an event for this sensor node. ADP is designed to maximize the network lifetime and save on energy consumption by optimizing the duty cycle of the node. When the frequency of the sensing traffic is high, the node should be adjusted to wake up more frequently in order to quickly

Afraa Attiah is also affiliated with the Collage of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia.

report each sensing event without much latency. When the sensor node has a low battery level, its sleeping time will be adjusted to be longer in order to extend its lifetime. When the sensing event is more critical to report, the node should wake up more frequently in order to reduce the reporting latency.

Sensors have two major operations: sensing and forwarding data [4]. In this paper, we focus on producing an adaptive and energy-efficient scheduling approach for sensors to sense and report events. It can be readily combined with many prior developed systems that focus on energy-efficient data forwarding in order to have a complete energy efficient scheduling system that covers both data sensing and data forwarding operations of sensors. Our simulation experiments show that ADP could greatly extend a sensor node lifetime comparing with a well known scheduling base approach [5] without introducing much latency which is especially suitable for a scenario where sensing events occur with varying frequency.

The rest of this paper is organized as follows: section II describes some of the related works. We present the ADP approach in section III. We test our approach during the simulation, and we discuss the results in section IV. Section V is the discussion and conclusion of the paper.

II. RELATED WORK

Most wireless sensor networks' protocols have been based on application requirements. Recently, researchers have been using sleeping techniques for reducing energy consumption in all layers of the protocol stack in wireless sensor networks [2]. Previous works have shown a broad range of the use of sleeping techniques in different categories. The sleeping techniques can be divided into scheduled wake-up, radio controlled wake-up, and environmentally controlled wake-up. Scheduled wake-up is divided based on time synchronization, where it could be synchronous or asynchronous duty cycling [2] [6]. EE-MAC (Energy Efficiency MAC protocol) [7] is our previous work that introduced an energy efficient medium-access control protocol by achieving a low duty cycle in order to improve energy consumption through a mathematical optimization model while transitioning between sleep and an active state. ER-MAC [8] is a TDMA based MAC protocol that selects the sleep and wake schedules based on a node's criticality by letting the more critical nodes sleep longer. The sleeping techniques can also save energy in routing protocols as some studies showed [2]. [9] proposed a sleeping multipath routing approach that can be applied to any routing protocol by selecting the minimum numbers of disjoint paths to meet the reliability demands and by turning off the rest of the sensor nodes. GTC (Geographical Topology Control protocol) [10] extends the network lifetime by dividing the network into zones and selecting one active node in each zone.

Sensors have two major operations: sensing and forwarding data [4]. In ADP, we focus on producing an energy-efficient way to sense an event based on the feedback. Other researches, such as PW-MAC [11], focus on the forwarding and transmission of sensed data. PW-MAC [11] is an energy-efficient predictive wakeup MAC protocol that enables senders

to accurately predict receivers wakeup times. The protocol minimized idle listening and overhearing by enabling a sender to rendezvous with a receiver quickly according to the predicted receiver wake-up. It could be beneficial to combine PW-MAC technique and our proposed approach together to have a complete energy efficient scheduling system.

III. PROPOSED ADP APPROACH

A. Wake-up Technique

Waking a node up and putting it to sleep periodically instead of keeping the node awake all the time saves significant amount of energy. A periodical scheduling technique could be synchronized, where all the nodes will adjust to the periodic wake-up time synchronously. On the other hand, the scheduling technique could be asynchronous, where each node's wake-up time does not require any synchronization, and each node can adjust its own periodic wake-up time independently [12]. Some existing approaches [5] use a base approach of wake-up technique that gives a node a fixed period of sleeping time throughout the node's lifetime. In the base approach, the node wakes up after a fixed amount of time, which is not suitable with dynamic changing sensing events. As an example application, sensors for monitoring a bridge condition may have very dynamically changing sensing activities to monitor and report throughout a day. During rush hours, sensor nodes will be busy and need to be awake more frequently to report sensing events than during nighttime, when vehicular traffic over the bridge is dramatically decreased. Clearly, fixed wake-up time scheduling depletes sensor nodes an unnecessary high amount of energy at night, and at the same time, sensor nodes may not wake up quickly enough during rush hours in order to sense and report events on time. In contrast, our approach adapts the node waking up scheduling based on the occurrence frequency of environmental events.

Although ADP runs on each sensor node independently, if some sensor nodes have exactly the same settings and observe the same sequence of events, executing ADP on these sensors will enable them to have identical sleep/wake-up scheduling, i.e., they are in synchronous mode. On the other hand, two sensors are in asynchronous mode if they have different settings or observe different events. Therefore, we can say that ADP is a hybrid approach by combining synchronous and asynchronous modes.

B. Criticality of Sensor Node

Unlike existing methods, where all nodes are treated equally all the time, we treat each node in ADP according to its own conditions (which we call criticality), and adapt its sleep/wake-up duty cycle with underlying sensing traffic density. We measure the criticality of a sensor node by the following two parameters:

- **Residual energy of a sensor node:** each node has its own residual energy level, and it varies according to the node activity and past energy consumption during its lifetime.
- **Importance of reporting data:** due to the application requirements, types of sensing data, and node locations,

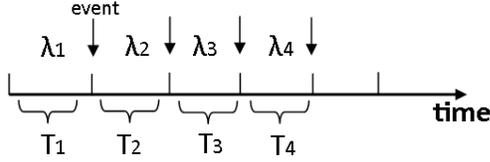


Fig. 1: Illustration of sensing events arrival. T_i is the inter-arrival time between the i -th event and the previous $(i - 1)$ -th event; λ_i is the estimated Poisson arrival rate based on T_i where $\lambda_i = 1/T_i$.

each node could have different values measuring the importance of reporting events it needs to sense and report.

C. Sensing Event Modeling and Prediction

The static behavior of the traditional system under varying sensing event density may increase the energy wastage and could reduce the efficiency of a sensor network. The main idea of ADP is to adjust the optimal sleeping time for each node and to adapt the network sensor node to be appropriate with an environmental dynamic-changing traffic load. In ADP, a sensor node will stay awake for a certain amount of time. If there is no event to report, it will go back to sleep immediately; if there is an existing event to report, it will report the data and then go back to sleep. Its next wake-up time is determined according to the node's criticality and the prediction of the next event arrival time.

We assume that the underlying sensing events follow the Poisson Distribution with the dynamically changing rate λ_t at time t . We estimate λ_t in each sleep/wake-up cycle based on previous observations of event arrivals (i.e., the T_i sequence) and the prior estimated value of λ_t (denoted by $\bar{\lambda}_t$). We apply the idea of estimating the new arrival rate via a low-pass filter [13].

$$\bar{\lambda}_t = (1 - \alpha)\lambda_i + \alpha\bar{\lambda}_{t'} \quad (1)$$

where λ_i is the Poisson arrival rate based on the most recent arrival event, $\bar{\lambda}_t$ is estimated arrival rate, and α is a filter gain coefficient to adjust how smooth we want the estimated $\bar{\lambda}_t$.

Here we explain how we obtain λ_i based on the most recent observation. Figure 1 illustrates sensing events occurrence over time. We denote T_i as the inter-arrival time between the i -th event and the previous $(i - 1)$ -th event. Since we assume that sensing events follow the Poisson arrival process, and sensor nodes know when each previous event happened, we use the observed most recent inter-arrival time T_i to estimate the current time Poisson process rate λ_i , and, hence, we set the value of λ_i as: $\lambda_i = \frac{1}{T_i}$.

For reader's convenience, we list the main mathematical notations used in this paper in Table I.

D. Feedback Optimization

1) Feedback Optimization Model for General Distribution

Our proposed approach is based on optimizing a cost function with the goal of minimizing the cost of energy consumed while matching with traffic density and maintaining an acceptable latency. The optimization tries to achieve a balanced

Notation	Definition
λ_t	The dynamic Poisson arrival rate for sensing events at time t
$\bar{\lambda}_t$	Estimated Poisson arrival rate at time t for sensing events
T_i	Inter-arrival time between the $(i - 1)$ -th event and i -th event
t_s	Sleeping time
ξ	Remaining battery of sensor node
r	Critical factor of remaining battery of sensor node, $r = \frac{1}{\xi}$
c	Factor of importance of reporting sensed event
P	Prob. of wasting energy when a sensor node wakes up without any event to report
Q	Prob. of finding event occurred during the sensor node's prior sleep period
d_{t_s}	Average sensing data report latency
w_1, w_2	Weight factors in cost function, where $w_1 + w_2 = 1$
t_s^*	Optimal sleeping time

TABLE I: List of Notations

trade-off between the energy consumption and sensing data report latency. There are two types of cost that we consider in our formula:

- The cost of energy wastage when the node wakes up without any sensing event happening during its previous sleep period.
- The cost of sensing data report latency when the node is sleeping during the occurrence of an event, thus introducing a time delay when it wakes up and reports the event.

The first case happens when the occurrence frequency of underlying sensing events is low and the node wakes up too often. The node will consume an undesired amount of energy in that awake time without reporting any events. In the second case, the cost of latency becomes high when the occurrence frequency of underlying sensing events is high, and the node wakes up less frequently. In this case, the sensor node sleeps longer than desired, whereas there are some events the node needs to report more responsively. Let random variable X denote the inter-arrival time between sensing events. We define the general formula of the combined cost function as:

$$f(t_s) = w_1 r P + w_2 c d_{t_s} Q \quad (2)$$

where r represents the criticality of remaining battery of a sensor node, c represents the importance of a sensed event, and t_s is sleeping time. The average latency is represented by d_{t_s} . $P = Prob.(X > t_s)$ is the probability of wasting energy when waking up in the absence of a sensing event (first case); $Q = Prob.(X \leq t_s)$ is the probability of finding an event occurrence during the prior sleep period (second case).

w_1 and w_2 are weight factors that should be set up by the network operator to achieve a balance between energy saving and data report latency. The cost function shows that the absolute values of w_1 and w_2 do not matter; what matters is the relative values of these two weight factors. Thus we can let

$$w_1 + w_2 = 1 \quad (3)$$

In order to find the optimal sleeping time t_s^* based on the cost function (2), we just need to take partial derivative of the

cost function against t_s and set it equal to zero, as $\left. \frac{\partial f}{\partial t_s} \right|_{t_s^*} = 0$.

2) Feedback Optimization Model based on Poisson Distribution

The above optimization model based on general distribution is theoretical and abstract. In order to illustrate how we can utilize this feedback optimization model in many sensor network applications, in this section we describe the traffic arrival process as a Poisson distribution and explain how to use the feedback optimization model to improve energy efficiency in a concrete way.

Poisson distribution is the most suitable distribution for the majority of sensor network applications. If there exist a large number of entities each of which has a very small probability to independently generate sensing events, then such event occurrence can be modeled accurately by a Poisson distribution. One example of such an application is in using sensors to monitor the condition of a bridge and the traffic flowing over it. There could be millions of vehicles in the local area of the bridge, but the probability of any one vehicle going over the bridge at a specific time is very small. A similar instance can be found in sensors monitoring wildlife, where the population of wildlife is large, but the probability of a specific animal appears in the specific area for the sensor to detect is small.

As we described above, X represents the inter-arrival time between sensing events. Since we assume the sensing event occurrence follows the Poisson process with a dynamically changing rate λ , this random variable X follows exponential distribution with the same rate λ . ($X > t_s$) denotes the absence of a sensing event during the time interval $[0, t_s]$. The probability of absence of sensing event when waking up is $Prob.(X > t_s)$, which is given by the following formula:

$$P = Prob.(X > t_s) = e^{-\lambda t_s} \quad (4)$$

Similarly, the probability of event occurrence during the sleep time interval $[0, t_s]$ is:

$$Q = Prob.(X \leq t_s) = 1 - e^{-\lambda t_s} \quad (5)$$

Because of the following Poisson process Theorem: "Given that $N(t = n)$, then those n arrival times S_1, \dots, S_n have the same distribution as the order statistics corresponding to n independent random variables uniformly distributed on the time interval $(0, t)$ " [14], we define the average latency d_{t_s} in our cost function (2) as half of the sleeping time $d_{t_s} = \frac{t_s}{2}$.

In addition, we define the critical factor of remaining battery of a sensor node as $r = \frac{1}{\xi}$, where ξ is the fraction of remaining battery energy as compared with the battery's full capacity. The importance of sensed events parameter c is specified manually by the operator for each sensor node according to its location and sensing data type.

After deriving the formulas for all the variables, the cost function becomes:

$$f(t_s) = [w_1 \frac{1}{\xi} (e^{-\lambda t_s})] + [w_2 c \frac{t_s}{2} (1 - e^{-\lambda t_s})] \quad (6)$$

In the above cost function equation, the first part is the cost of wasting energy, and the second part represents the cost of sensing data report latency. In order to drive the optimal sleeping time t_s^* , we need to take partial derivative of the cost function (6) in terms of t_s . Since we don't know the true value of λ , we use the estimated $\bar{\lambda}_t$ from Equation (1) in the cost function. The optimal sleeping time t_s^* should make the derivative equal to zero, which means that t_s^* can be derived from the following equation:

$$\left. \frac{\partial f}{\partial t_s} \right|_{t_s^*} = \frac{w_2 c}{2} + e^{-\bar{\lambda}_t t_s^*} \left[\frac{w_2 c t_s^* \bar{\lambda}_t}{2} - \frac{\bar{\lambda}_t w_1}{\xi} - \frac{w_2 c}{2} \right] = 0 \quad (7)$$

Since Equation (7) does not have a closed-form solution, we apply Bisection algorithm [15] for estimating the root of the Equation (7). When a node wakes up, its value of $r = \frac{1}{\xi}$ updates based on the current remaining battery energy. In addition, the estimated event arrival rate $\bar{\lambda}_t$ updates by the estimation Equation (1), then ADP relies on Equation (7) to determine the node's optimal sleeping time t_s^* for the next round.

Algorithm (1) shows the procedure of the proposed adaptive scheduling approach. It contains three steps in each wake-up cycle: prediction, updating, and optimization. The first step, Prediction, is used to predict when will the next sensing event will happen based on the event's statistical model and the previous events observation. It will make the system adaptive to the dynamics of sensing events. The second step, Updating, is to update all the parameters in the cost function (6). The last step, Optimization, is to derive the optimal next-round sleeping time t_s^* based on the partial derivative (7).

IV. PERFORMANCE EVALUATION

A. Simulation Setup

In this section, we evaluate our approach by comparing ADP with the base approach which applies the same technique but without adaptation, i.e., the sensor nodes in the base approach have a fixed period of sleeping time. We conduct the experiments that test ADP and the base approach over three different underlying sensing event densities in order to illustrate the impact of underlying dynamic sensing event load on the sensor nodes' behavior. In the first scenario, sensing event occurrence follows a constant rate of Poisson process all the time. In the second scenario, the sensing event Poisson arrival rate λ is increased from λ_{low} to λ_{high} rate in the middle of the simulation. The last scenario is the reverse of the second scenario, where λ starts from λ_{high} and decreases to λ_{low} rate in the middle of the simulation.

We simulate our sensing approach and base approach in Matlab. To be realistic, we use the parameter values of TelosB Mote, a low-power wireless sensor module, as battery energy model as specified in [16]. That is to say, the value of power consumption in the wake-up state is $1.8mA$, and power consumption in sleep state is $5.1\mu A$. We set up 10 nodes and classify the nodes into three groups, i.e., nodes in each group have the same settings and observe/report the same sequence

Result: Each node computes the optimal sleep time period t_s^* for the next sleep-wake duty cycle, determines the next wake-up time

Initialization: The network operator sets the values of data importance factor c for each sensor node and sets the value of weight factors w_1 and w_2 in order to achieve a balance between energy saving and data report latency.

begin

When a node wakes up

1.Prediction: estimate the new $\bar{\lambda}_t$ using Equation 1 based on past observations and prior estimated

value

2.Updating: update the feedback information, i.e., derive $r = \frac{1}{\xi}$ and $\bar{\lambda}_t$

3.Optimization: derive t_s^* based on Equation 7 by using the feedback value of $r = \frac{1}{\xi}$ and the estimated Poisson arrival rate $\bar{\lambda}_t$

if event has happened during prior sleep period **do**
Action (i.e., report the event)

end if

Schedule the node next wake-up time:

{next-wake-up-time=current-time + t_s^* }

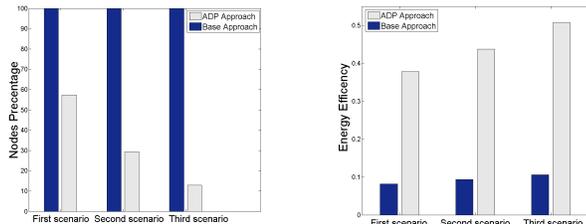
The node goes to sleep

end

Algorithm 1: Procedure of ADP

of sensing events. As explained at the end of Section III.A, sensors in each group will achieve exactly the same scheduling by running ADP independently, as if they synchronize with each other. Node 1 to 3 are in Group 1; Node 4 to 6 are in Group 2; and Node 7 to 10 are in Group 3. In order to get an accurate results, we average the simulation results over 100 runs.

The proposed ADP approach tries to achieve a balanced trade-off between energy saving and data report latency. As shown in the cost function (6), the network operator can adjust the relative values of the two weight factors w_1 and w_2 to achieve energy saving while maintaining an acceptable sensing data report latency. For example, by increasing the value of w_2/w_1 , the operator can reduce data report latency at the cost of saving less amount of energy. We define the latency as the



(a) Percentage number of nodes that have less than 20% of energy. (b) Energy Efficiency = ratio of the sum of nodes remaining energy to the sum of nodes initial energy.

Fig. 2: Performance of ADP compared with the base approach at the end of time.

time interval between occurrence of sensing event and node wake up to report the event. Furthermore, there is no universal amount of delay that can be defined as acceptable latency because it purely depends on the application and should define by the operator. We assume the maximum acceptable latency range for our simulation is $[4 - 6s]$.

Based on the two performance metrics of remaining battery energy and data report latency, we evaluate system performance from two perspectives: first, by the end performance, which illustrates the behavior of sensor nodes at the end of the simulation time; second, by the temporal performance that shows the behavior of the sensor nodes along their lifetime in three experiments. In Experiment I, the value of sleeping time for the base approach is $t_s = 1/\lambda_{avg}$. In Experiment II, we set different values of sleeping time for the base approach, which is $t_s = 1/\lambda_{high}$, which will make the nodes wake up more frequently. In order to show the nodes' behavior in terms of latency and energy saving while assuming a lower acceptable amount of latency, in Experiment III, we keep the value of sleeping time for the base approach as $t_s = 1/\lambda_{avg}$ but with a higher value of w_2 in order to reduce the latency. The performance of ADP is tested using the following metrics:

- Energy Efficiency: ratio of summation of remaining energy for all nodes divided by the summation of initial energy of all nodes.
- Average remaining energy for all nodes along the simulation time and remaining energy for each node at the end of simulation.
- Average latency for all nodes along the simulation time and average latency for each node at the end of simulation.
- Percentage number of nodes that have less than 20% of energy remaining as compared with their initial energy.

B. Simulation results

1) Impact of sensing event density

For the three different sensing event loads as mentioned above, we measure the energy efficiency and percentage number of nodes that have less than 20% of energy at the end of the simulation for both our proposed ADP approach and the base approach. In the base approach, the value of sleeping time is set to be $1/\lambda_{avg}$. Figure 2(a) illustrates the percentage ratio of the number of nodes that have less than 20% of energy level over the three scenarios. This figure shows that all nodes consume more than 80% of their energy at the end of the simulation in the base approach, while 44% of the nodes in ADP still has more than 20% of their energy in the first scenario and 60%, 87% of the nodes still has more than 20% in the second and third scenarios, respectively. Figure 2(b) illustrates the measurement of energy efficiency for ADP and the base approach. Compared with the base approach, ADP achieves a higher energy efficiency for all cases. As the sensing event load changes in the second and third scenarios, ADP still has a higher energy efficiency than the base approach.

2) Energy saving and latency

This section represents the results of three experiments, as mentioned above, over the second scenario of underlying

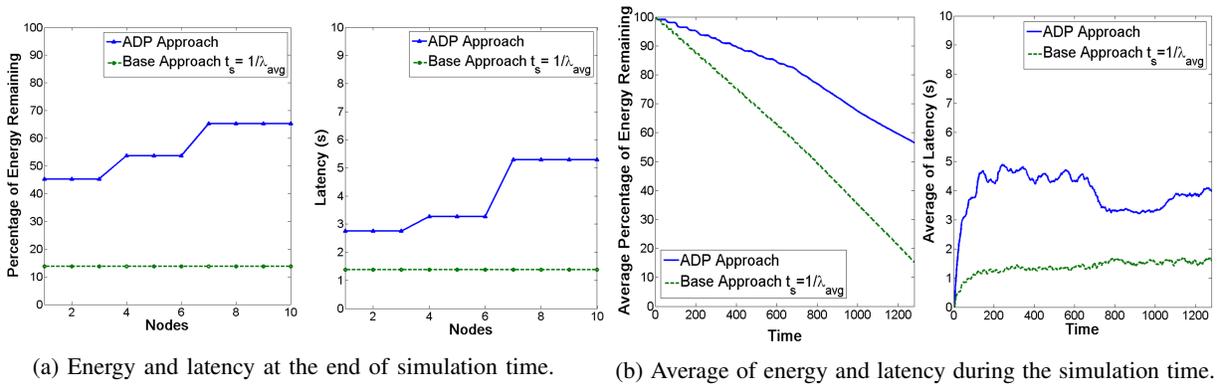


Fig. 3: Performance of ADP and the base approach in Experiment I.(when the value of fixed sleeping time t_s for base approach is $1/\lambda_{avg}$.) ADP gains a high amount of energy saving and keeps latency well below the acceptable latency. In (b), change the middle of the curves refer to the density change of sensing event.

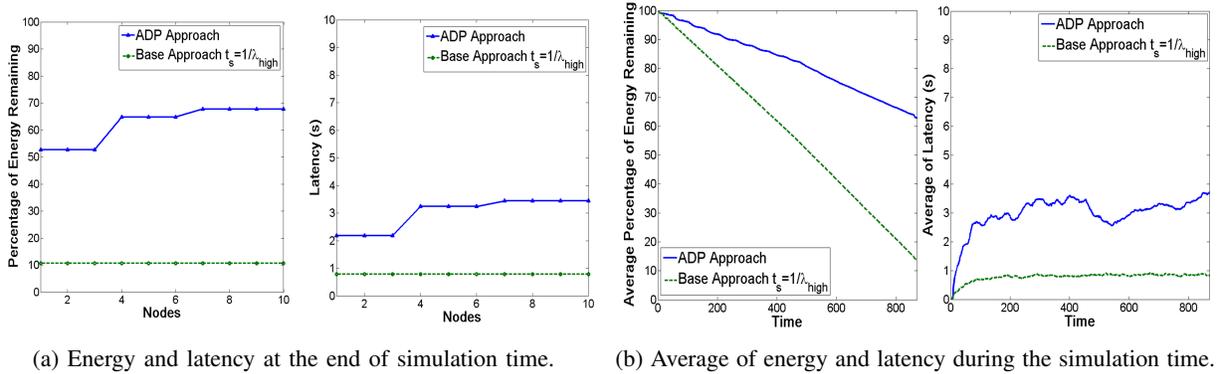


Fig. 4: Performance of ADP in Experiment II. (The results similar to Fig. 3 when the value of fixed sleeping time t_s for base approach is $1/\lambda_{high}$ instead of $1/\lambda_{avg}$ (waking up more frequently).) ADP also gains a higher amount of energy saving.

sensing events; in this second scenario the sensing event Poisson arrival rate λ is increased from λ_{low} to λ_{high} rate at the middle of the simulation. The results show the performance of ADP and the base approach. The following figures 3, 4, and 5 demonstrate the behavior of sensor nodes for saving energy and latency over a dynamic changing underlying sensing event load. Figure 3 represents the performance of ADP and the base approach when the value of fixed sleeping time in the base approach is $1/\lambda_{avg}$. The percentage of remaining energy and the amount of latency for each node at the end of simulation are showed in figure 3(a).

To study the impact of factor c , the importance of reporting data, the first group of Node 1 to 3 has the largest value of c , while the third group of Node 7 to 10 has the smallest value of c . The proposed ADP approach makes nodes consume less energy than the nodes in the base approach, and the latency stays low within the acceptable range set. Figure 3(b) demonstrates the measurement of the average of percentage of remaining energy and latency for nodes throughout the simulation. The graph of average percentage of remaining energy shows the amount of energy saved in ADP is more than that in the base approach by 40%. When the sensing event density changes in the middle of simulation, energy consumption rate in the ADP approach also changes correspondingly. In addition, the

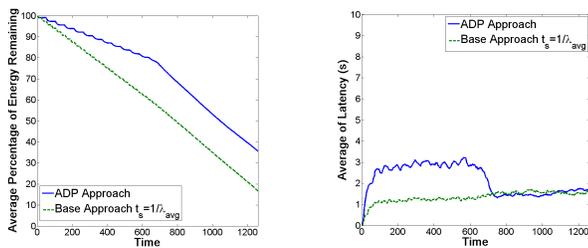
average of latency in our approach decreases in the middle of simulation according to the density change of sensing events.

Figure 4 illustrates the experiment's results when the value of t_s in the base approach is $1/\lambda_{high}$. Figure 4(a) and 4(b) represent the percentage of remaining energy and latency for each node at the end of the simulation, and the average of remaining energy and latency throughout the simulation, respectively. In this instance, we notice that our approach also achieves high performance for saving energy by 45% and keeps latency under the maximum acceptable latency.

To show the nodes' behavior in term of energy saving and latency with more emphasis on reducing the latency than on energy saving, we test ADP and the base approach by changing the parameters of weighted factors in the experiment III. The results of the average energy remaining and latency are represented in Figure 5, and follow the same trend as the previous experiment for saving energy. The figure shows that our approach saves 15% energy when compared to the base approach. It achieves a good improvement in latency, and it could even achieve less latency when compared with the base approach.

V. DISCUSSION AND CONCLUSION

Our feedback optimization model is not restricted to the Poisson process. The model of sensing event occurrence could



(a) Average of energy during time. (b) Average of latency during time.

Fig. 5: Performance of ADP and the base approach in Experiment III ($t_s = 1/\lambda_{avg}$, and this experiment emphasizes more on reducing latency than on energy saving by setting a higher value of w_2 .) Comparing with Fig. 4(b), ADP achieves better latency performance with slight cost of energy.

follow different distributions according to the sensor network applications, such as Pareto distribution, ON/OFF Markov models [17], and Weibull distribution.

The values of weight factors w_1 and w_2 in our feedback optimization model (2) are critical for system performance. Their values can be configured in two ways by the sensor network operator: first, based on the experience of the operator and on the previous usage of the system. Second, if the operator has the model for the sensor network application based on previous observations, the optimal values of w_1 and w_2 can be defined by running the simulation of the system (like what we did in our performance evaluation) repeatedly to achieve the best simulation results. Figure 6 shows this simulation-based configuration process.

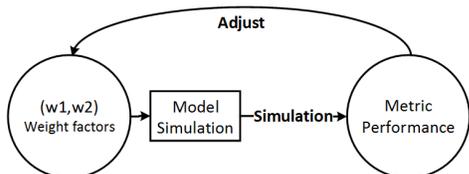


Fig. 6: Simulation-based framework for designing weight factors w_1 and w_2 based on existence model.

In this paper, we have proposed a novel adaptive energy saving approach called ADP for wireless sensors to effectively extend the network lifetime without introducing much data sensing report latency. The goal of ADP is to optimally adjust sleep time for each node dynamically, and to adapt the behavior of the sensor nodes with changing underlying sensing event load, remaining battery level, and the importance of sensing data. Based on the results of our experiments, ADP achieves a significant gain in energy saving, high energy efficiency, and desirable performance for latency over different traffic scenarios.

REFERENCES

[1] L. C. Zhong, R. Shah, C. Guo, and J. Rabaey, "An ultra-low power and distributed access protocol for broadband wireless sensor networks," *IEEE Broadband Wireless Summit*, vol. 3, 2001.

[2] R. Muraleedharan, I. Demirkol, O. Yang, H. Ba, S. Ray, and W. Heinzelman, "Sleeping techniques for reducing energy dissipation," in *The Art of Wireless Sensor Networks*, pp. 163–197, Springer, 2014.

[3] A. Munir and A. Gordon-Ross, "Optimization approaches in wireless sensor networks," *Sustainable Wireless Sensor Networks*, pp. 313–338, 2010.

[4] A. Erdogan, V. Coskun, and A. Kavak, "The sectoral sweeper scheme for wireless sensor networks: Adaptive," 2006.

[5] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, pp. 1567–1576 vol.3, 2002.

[6] O. Yang and W. Heinzelman, "Modeling and performance analysis for duty-cycled MAC protocols with applications to S-MAC and X-MAC," *Mobile Computing, IEEE Transactions on*, vol. 11, pp. 905–921, June 2012.

[7] A. Attiah, M. I. Akbas, M. Chatterjee, and D. Turgut, "EE-MAC: Energy efficient sensor mac layer protocol," in *Local Computer Networks Workshops (LCN Workshops), 2013 IEEE 38th Conference on*, pp. 116–119, IEEE, 2013.

[8] R. Kannan, R. Kalidindi, S. S. Iyengar, and V. Kumar, "Energy and rate based MAC protocol for wireless sensor networks," *ACM Sigmod Record*, vol. 32, no. 4, pp. 60–65, 2003.

[9] O. Yang and W. Heinzelman, "Sleeping multipath routing: A trade-off between reliability and lifetime in wireless sensor networks," in *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, pp. 1–5, IEEE, 2011.

[10] L. Aslanyan, H. Aslanyan, and H. Khosravi, "Optimal node scheduling for integrated connected-coverage in wireless sensor networks," in *Computer Science and Information Technologies (CSIT), 2013*, pp. 1–13, Sept 2013.

[11] L. Tang, Y. Sun, O. Gurewitz, and D. B. Johnson, "PW-MAC: An energy-efficient predictive-wakeup MAC protocol for wireless sensor networks," in *INFOCOM, 2011 Proceedings IEEE*, pp. 1305–1313, IEEE, 2011.

[12] M. R. Ahmad, E. Dutkiewicz, and X. Huang, "A survey of low duty cycle MAC protocols in wireless sensor networks," *Book Chapter in, Wireless Sensor Network*, 2009.

[13] V. Jacobson, "Congestion avoidance and control," in *ACM SIGCOMM Computer Communication Review*, vol. 18, pp. 314–329, ACM, 1988.

[14] M. R. Sheldon, "Introduction to probability models," 2010.

[15] D. Estep, "The bisection algorithm," *Practical Analysis in One Variable*, pp. 165–177, 2002.

[16] MEMSIC Inc, "Telmosb mote platform," Online: http://www.memsic.com/userfiles/files/Datasheets/WSN/telosb_datasheet.pdf, 2014.

[17] Q. Wang, "Traffic analysis & modeling in wireless sensor networks and their applications on network optimization and anomaly detection," *Network Protocols and Algorithms*, vol. 2, no. 1, pp. 74–92, 2010.