Dynamic Power Management Model for a Wireless Sensor Node

Rakhee Kallimani, Sridhar Iyer*

Department of Electronics and Communication Engineering, SG Balekundri Institute of Technology, Karnataka, India

Received 10 September 2021; received in revised form 30 October 2021; accepted 31 October 2021

DOI: https://doi.org/10.46604/peti.2021.8455

Abstract

Dynamic power management (DPM) is an efficient technique to design low-power and energy-efficient nodes for wireless sensor networks. This article demonstrates the stochastic behaviour of an input event arrival which is modelled with first-in first-out (FIFO) queue and a single server. An event-driven sensor node is developed based on semi-Markov model. The article investigates the factors affecting the performance of the individual sensor node with detailed analysis considering power consumption and lifetime to be the performance metrics under study. The results demonstrate the impact of the change in event arrival and the probability of change detection on the performance of the node. It is observed that (i) the number of generated events increases with the change in the average value of the distribution which affects the service time in turn resulting in a variation of the server utilization, and that (ii) the increase in the detection probability increases the power consumption decreasing the lifetime of the node.

Keywords: dynamic power management, wireless sensor node, semi-Markov model

1. Introduction

With recent progress in wireless communication and network technology, the conventional methods of information gathering, processing, and communication are no longer applicable to the present and the future user demands. Furthermore, recent advance in embedded technology has motivated many researchers to analyze wireless sensor networks (WSNs), the networks composed of individual nodes positioned at distinct locations. In WSNs, each sensor node is composed of (i) a dedicated sensing unit to sense the variation in the environment, (ii) a processing unit to process the information, and (iii) a communication unit to communicate with the neighboring node or a sinking node. Each unit in the sensor node is supplied with power for the node’s effective functioning, and the node can be portable or non-portable, based on the source of power and application. In specific, portable and battery-operated devices have attracted researchers’ attention recently. Furthermore, WSNs have been extensively used in various applications such as military, health care, biomedical, agricultural, etc.

Researchers and developers have shown keen interest in the field of embedded systems, especially in WSNs. WSNs embed sensor nodes; each individual node collects the information from the environment and then communicates to either a next node or gateway for further analysis [1]. These nodes are deployed at remote places where human intervention is rare, and hence it is expected for the nodes to be self-monitored, self-controlled, inexpensive, and power-aware [2]. The primary objective of the networks is to monitor the surrounding environment with the nodes powered by a battery. Thus, performance and power are the main design metrics that have attracted the attention of researchers. The significance of power management is gaining the attention that (i) the nodes are defined to perform tasks with efficient performance, (ii) the replacement/recharging of batteries is a challenge, and (iii) the networks may fail prematurely due to the failure of an individual power-hungry node. In other words, there is a need for managing the power of each node individually and improve the network performance. In the state-of-art networks, the performance is dependent on the battery-powered nodes as these nodes are resource-constrained.

* Corresponding author. E-mail address: sridhariyer1983@gmail.com
In the embedded systems, there are numerous methods employed to manage the power at different levels of design. The current investigation is related to the power management at the system level such that the power consumption is reduced dynamically and there is no cost of hardware replacement. One such efficient management technique is dynamic power management (DPM). The technique tracks the system activity and the power manageable components, then dynamically switches the system from active state to passive state (low-power state), or switches off the components if required. This contributes to significantly improving the power and performance of the nodes. Academicians and researchers have therefore conducted various stochastic techniques to design such nodes. To improve the performance of the entire networks, this study proposes a DPM based sensor node model. The rest of the article is structured as follows. Section 2 reviews the related work. The methodology of the proposed DPM node is detailed in section 3. Section 4 presents the assumed simulation parameters in modelling the node, and provides the discussion on the obtained results. Lastly, the study is concluded in section 5.

2. Related Work

In a WSN embedding several sensor nodes, each node has subunits for sensing, processing, and communication. These units are powered by a unit responsible for sourcing power. The power unit sources the power to each unit [3]. Much analysis is carried out in managing the power of these energy-hungry nodes at the communication unit; however, the processing unit and power unit are neglected. Popovici et al. [4] explored the power management methods at the system level. The work focused on ultra-low-power wake-up radio receivers. The work also presented the design of a wake-up radio receiver and demonstrated the low-power consumption at low cost. Pughat et al. [5] conducted a detailed survey to discuss the different power management techniques at the operating system level along with their limitations. They also provided the scope for researchers to contribute the power saving techniques; their work motivated the present study towards stochastic control of power-saving.

Healy et al. [6] reviewed the available operating systems for sensor nodes. They listed four operating systems: TinyOS, Mantis OS, SOS, and T2; the supporting hardware was also presented. In their work, the need of managing the power at the operating system level was addressed. The component-based modelling is more flexible and easier for developers to experiment with power consumption for enhancing the battery lifetime. Their work pointed out the critical issue of operating system in power management, and this motivated the present work.

John et al. [7] presented a model for power consumption considering sensor nodes to be in different low-power modes, and simulated the WSN using OPNET simulator. The major objective of their work was to determine the average power consumption and lifetime of the nodes. The researchers analyzed the energy consumption using OPNET simulator of ad hoc nodes (IEEE 802.11). Chung et al. [8] emphasized the power management techniques considering the service request and controlling the policy in the design of modelling at the system-level power management. The time-out based, predictive, and stochastic techniques of DPM policy was discussed with the merits and the limitations. The DPM in the known stationary and unknown stationary environment was studied considering the nature of workloads. The authors introduced the adaptive management policies for non-stationary loads based on the sliding window concept. The power management for non-stationary workloads at multiple devices remains the limitation of the work.

Lee et al. [9] introduced a novel concept in DPM; the analysis of their proposed hybrid model was done by using rescaled range. The time-out method was used in the hybrid model to perform the comparison between the predicted value and threshold value of time, and a decision was taken based on the rescaled range analysis. To model the time series prediction problem, artificial neural network (ANN) was used with multilayer perceptron (MLP) and back propagation (BP). Their interest in work motivated the attention towards the analysis of workload characteristics of portable devices. The research on identifying the cluster head in WSN by the smart-in-a-smart-intelligent approach using evolutionary game theory was proposed by Arivalai et al. [10].
Markov model is used to analyze the power and performance of WSN in the stochastic environment. Many researchers have contributed to the improvement of WSN by performing the analysis for power consumption and lifetime. Pughat et al. [11] developed a WSN model related to stochastic modelling, and compared its power efficiency with the conventional node. Yamawaki et al. [12] used an adaptive suspending scheduling method to improve the battery lifetime, and mentioned different techniques to manage the power for WSN applications. Pughat et al. [13] analyzed the trade-off between the power and performance of a WSN node. Kallimani [14] surveyed the power management techniques at the system level, and emphasized the need for designing a power-managed embedded system. Pughat et al. [15] worked on the stochastic behaviour of event arrivals to sensor nodes considering modelling the sensor nodes to sense the forest fire detection. The novelty of the work is the analysis of power consumption performed by implementing the control logic of switching the inactive components to low-power modes by fuzzy logic. Their work motivated the present work in performing the analysis of power consumption. The authors also mentioned that their proposed approach can be applied to all energy constraint nodes. Kallimani et al. [16] developed a stochastic sensor node using PYTHON, and demonstrated the performance analysis of the discrete event-based environment. Also, they emphasized that the performance can be further improved using an analyzer unit.

Ever et al. [17] investigated the performance and availability requirement, and pointed out the impact of queue capacity and system scalability of the clustered Internet of Things (IoT) systems. Concerning the industrial IoT networks, a novel power management scheme was proposed in the work of Kiran et al. [18]. The authors formulated a mathematical model, and analyzed the network’s performance. Their proposed model achieved 74.82% energy-saving using the 3D Markov chain and M/G/1/K queue. IEEE 802.15.4 MAC was analyzed, and the effectiveness of the developed model was compared with simulation and real-time testbed deployed at the authors’ campus. The authors concluded that there is 97% accuracy with simulation and 94% accuracy with experimental results compared to the proposed model.

Lukman et al. [19] focused on power consumption, mobility, and end-to-end communication. The authors developed a sensing platform to sense environmental parameters such as humidity and temperature. They demonstrated the results on power consumption and end-to-end communication (WSN-IoT). The work is at the MAC layer of IoT based stack for schedule-driven sensor nodes, using Zigbee communication modules to establish end-to-end communication. The researchers emphasized the need of enhancing the lifetime in terms of the battery of individual nodes.

Jawad et al. [20] performed an extensive analysis study on power consumption on wireless agricultural system (WAS) using a sleep/wake scheme. The authors proposed an algorithm called SWORD to minimize the power consumption and also the data communication of the sensor nodes. The results of the proposed algorithm proved to be 86.54% more reliable than the sleep/wake scheme. The work majorly focuses on reducing the power consumption by transmitting only the data which is more than the threshold value; otherwise the communication unit will be in the sleep mode. Cañete et al. [21] contributed detailed performance analysis and queuing systems for monitoring application. The analysis’s outcome shows the significance of reliability at the data link and the network layer. The reason for the congestion of the network is demonstrated to be the delayed occupation of the queue.

Phung-Duc et al. [22] carried out a study on delay performance considering a multi-server queue system using Markov chain. Zhang et al. [23] used Markov decision process to analyze the decision policy on energy consumption of the nodes by switching the nodes’ status. The proposed model is applicable for all the events that arrive at the sensor input. However, this is also the model’s limitation as the event arrived could be an undesired event and the event serviced without the analysis could lead to an increase in the power consumption. Considering that there are queued events in the memory, the extensive work is conducted by De Cuypere et al. [24]. Their proposed model considers two queue systems representing the battery and the sensor data buffer, and the results of experimentation reveal that the data arrival has a limited impact on the performance measures.
Nguyen et al. [25] identified the best routing protocol with minimum delay, and compared their analysis with simulation results. The delays of candidate routes from sensor nodes to sink nodes were evaluated by the team using queuing theory. Sun et al. [26] investigated the M/G/1 queuing theory, and proposed a three-way handshaking in-band full-duplex medium access control (MAC) mechanism. The obtained results show that the full duplex access point outperforms the half-duplex access point. Furthermore, Li et al. [27] investigated and proposed a displacement sensor for structural health monitoring, and detected the early corrosion based on fiber Bragg sensor. The WSN in the industrial environment is an important application, and Cao et al. [28] formulated an optimization problem considering the deployment problem. The authors continued the work of Cao et al. [29] for wireless data center network, and formulated an optimization problem considering the coverage to be objective.

There is also the contribution from researchers in reducing the power consumption of WSN for varied applications using different technologies; however, there is limited work concerning the contribution on lifetime analysis and extending the lifetime by dynamically switching the processor power into the low-power modes. The existing literature details the limitation of developing the model using a stochastic approach. The existing work in literature also uses either discrete or continuous Markov model. However, the current scenario demands the consideration of a semi-Markov model for WSN applications to control the power of a node.

The aforementioned research motivates the need of managing the power at the processing unit simultaneously with the power unit. Furthermore, if the power is optimally managed at the node level, it would enhance the lifetime of the network to promising levels. Thus, in this article, to improve the performance in terms of the lifetime by optimally managing the power at the node level, the DPM technique is implemented [30]. To manage the power at the node level, the best possible approach is to extend the lifetime of a battery by employing energy harvesting capabilities, or to dynamically manage the power of the processing unit by using any of the DPM algorithms. The effect of WSN parameters affecting the lifetime of wireless sensor nodes is presented in the work of Younus [31].

As a novelty in this study, the analysis of a node is considered in terms of performance parameters such as lifetime and power consumption by varying the probability of change detection and event arrival considering first-in first-out (FIFO) queue discipline and single server using the DPM technique based on semi-Markov model.

3. Methodology

This article demonstrates a simulation model of a WSN node considering the stochastic environment. The simulation of the WSN node is developed based on queuing theory, considering the event arrival to be stochastic in nature. The proposed semi-Markov model is at the node level with an event generation subsystem, a power manager subsystem, and a single server demonstrating the use of DPM. The input is the event arrival generated by the event generation subsystem. The purpose of the power manager subsystem is to monitor the arrival of events and compute the power consumption and lifetime of the node.

In the current article, a Markovian model for an event-based application is proposed to meet the modelling requirement. The semi-Markov model is defined by a set of states, and the transition time varies for each state. The considered node is consisting of states, namely $S_0$, $S_1$, $S_2$, and $S_3$, as pre-processing, analysing, main processing, and communication states, respectively. Fig. 1 explains the transition of states. The transition from $S_0$ to $S_1$ is with the transition probability of 1, and the transition probability of $S_1$ to $S_2$ is denoted as $\alpha$, i.e., the probability of change detection. The transition from $S_1$ to $S_3$ is $(1-\alpha)$. The transition probability of $S_2$ to $S_3$ is 1. Here, $S_0$ is the pre-processing state, which filters the event arrived. $S_1$ is an analyser state, wherein the filtered event is analysed for the change in the value from the previous value. If there is a change detected, then it is termed as a desired event and moves to the next state for processing, otherwise it transits to the previous state. $S_2$ depicts the main processor, which wakes up only upon the desired event, otherwise the main processor would be in its low-power state. $S_3$ is the communication state, which communicates only when the desired events are processed by the main processor.
Two hypothesis statements are formulated. The first statement states that the power consumption can be managed dynamically by waking up the main processor only for the desired event passed by the analyser state. For all the undesired events, the main processor would be in its low-power state. The second hypothesis statement proposes to improve the lifetime of the node as multiple computations of the processor can be minimized for the same event. The assumptions for developing the simulation model are as follows:

1. Poisson distribution is used with a mean of “5” for generation of event arrival “λ”.
2. Exponential distribution is used with a mean of “1” for the inter-arrival time distribution.
3. The probability of change detection “α” is varied in the range of 0 to 1.
4. The system has four power states and simulation parameters, as shown in Table 1 and 2, respectively [8].
5. The switching cost of transition from $S_1$ to $S_2$ and $S_0$ is neglected.
6. The event-based random number generator is used to generate entities with Poisson distribution.
7. The time-based entity generator is used with exponential distribution.
8. Generated events are stored in the FIFO queue.

The model is developed using the following equations:

$$E_{avg} = T_{an} P_{S_1} + a(T_{pr} P_{S_2} + T_{tx} P_{S_3} + C_p + C_R)$$

(1)

$$T_{avg} = T_{an} + a(T_{pr} + T_{tx})$$

(2)

$$Power\ Consumed\ (P) = \frac{P_{S_0} + \lambda(E_{avg})}{1 + \lambda(T_{avg})}$$

(3)

$$Life\ Time = \frac{Battery\ Energy}{Power\ Consumed} = \frac{1 + \lambda(T_{avg}) Bat_Eng}{P_{S_0} + \lambda(E_{avg})}$$

(4)

where the battery energy ($Bat_Eng$) is assumed to be of AA battery and is used to calculate the lifetime of the node with average energy consumption ($E_{avg}$). The power ($P$) is consumed by the node and is calculated using $P_{S_0}$, $P_{S_1}$, $P_{S_2}$, and $P_{S_3}$ power state modes. The time spent in analyser state is denoted as $T_{an}$. $T_{pr}$ is the time spent in processing state, $T_{tx}$ is the time spent in transmission state, and $T_{avg}$ is the average time consumption. $C_p$ is the energy cost of processing and $C_R$ is the energy cost of communication unit [16].

The model is developed using MATLAB, SimEvents library package [32]. The basic abstract model of the system is shown in Fig. 2, and the simulation of the proposed model is shown in Fig. 3. The abstract model consists of a service provider, a service requestor, and a power manager unit. The service requestor is analogous to the generating events and is passed to the
memory, holding these events to be serviced by the service provider such as a processor. The main processor has power manageable components, wherein the power manager block observes the states of each block and accordingly issues a command to the processor and manages the power dynamically by switching the power manageable components into the power states. The DPM model as shown in Fig. 3 is developed by integrating the event generation subsystem to a single server and a power manager subsystem. The events to be served by the server are departed from the FIFO queue in the event generation subsystem. The power manager subsystem has an energy subsystem. The time subsystem is developed based on Eqs. (1)-(2) and a unit to detect the change in the arrival. The utilization of the server is obtained from the output port of the single server. The power consumption is computed by the power manager subsystem built on the governing Eq. (3). The subsystem also evaluates the lifetime of the node based on Eq. (4).

Table 1 Power state of the system [16]

<table>
<thead>
<tr>
<th>States</th>
<th>S₀</th>
<th>S₁</th>
<th>S₂</th>
<th>S₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>PS₀</td>
<td>On</td>
<td>Off</td>
<td>Off</td>
<td>Off</td>
</tr>
<tr>
<td>PS₁</td>
<td>On</td>
<td>On</td>
<td>Off</td>
<td>Off</td>
</tr>
<tr>
<td>PS₂</td>
<td>On</td>
<td>On</td>
<td>On</td>
<td>Off</td>
</tr>
<tr>
<td>PS₃</td>
<td>On</td>
<td>On</td>
<td>On</td>
<td>On</td>
</tr>
</tbody>
</table>

Table 2 Simulation parameters [16]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PS₀</td>
<td>2.3 mW</td>
</tr>
<tr>
<td>PS₁</td>
<td>2.3 mW</td>
</tr>
<tr>
<td>PS₂</td>
<td>237.5 mW</td>
</tr>
<tr>
<td>PS₃</td>
<td>307.9 mW</td>
</tr>
<tr>
<td>Cₛ</td>
<td>2.206 mJ</td>
</tr>
<tr>
<td>Tₛ</td>
<td>2 s</td>
</tr>
<tr>
<td>Tₑ</td>
<td>0.175 s</td>
</tr>
<tr>
<td>Tₑₑ</td>
<td>0.05 s</td>
</tr>
<tr>
<td>Bat_Eng</td>
<td>1863 J</td>
</tr>
<tr>
<td>Cᵣ</td>
<td>6.9 µJ</td>
</tr>
</tbody>
</table>

Fig. 2 Abstract model

Fig. 3 Developed DPM model
The event generation subsystem shown in Fig. 4 generates the events by Poisson distribution with a mean of 5 using the random generator block. The inter-arrival time is provided with an exponential distribution of the mean of 1 using the time-based entity generator. The set attribute block sets the characteristics of the event. The schedule time-out block schedules the time-out attribute. The generated event is then passed through a FIFO queue; FIFO is set with 1000 events as capacity and is further passed to the single server for the service. The output of this subsystem is the number of arrivals.

The power manager block is depicted in Fig. 5, which has basic functionality to observe the workload of the service requestor (arrival) or/and service provider (server) and issues the command to dynamically switch the states. The power manager block is built using the time subsystem and energy subsystem. This subsystem also computes the performance parameters such as $\alpha$ (the probability of change detection) and lifetime in days. The energy subsystem is shown in Fig. 5, and the time subsystem is built using the governing equations of the model. The functionality of the block is to compute the average energy and time using the simulation parameters fed as constant and $\alpha$ as varying in the range of 0 to 1.

4. Simulation Results and Discussion

The obtained simulation results for a single server queuing model are shown in Fig. 6. The scope of Simulink is used to obtain these graphs. It can be observed from the figure that the number of events increases with an increase in time, and the probability of change detection increases with the change in the events increasing over time thus simulating the real-time scenario.
Furthermore, the power consumption increases with the variation in time. The major power consumption occurs in the communication state and the processing state. As the processor needs to compute the events before communicating to the transmission unit, it needs to be in the active state. This leads to more power consumption as the processor is active for all the events that occur, irrespective of the need to be computed or not. The lifetime of the node is seen to decrease with the variation of the probability of change detection, thus reflecting the deviation in the sensed value with the event arrived and suggesting that it needs to be computed before communicating to the next node.

The first hypothesis statement stated is that the power consumption is reduced if the event is passed through the analyser state. In other words, this state will analyse if the event is desired or not, and if the event is analysed as an undesired event, then the main processor do not need to compute. Otherwise, the event is desired and needs to be processed and communicated. Thus, the consumed power of the node is minimum if the event is passed through the analyser state. Furthermore, the consumption of power is observed to increase as the probability of change detection increases, indicating that the occurred event is the desired event which needs to be processed by the main processor and hence switching the processor to an active state. If the event is undesired or the same event occurs, then the main processor can be switched to a low-power sleep mode to save power dynamically.

Fig. 6 shows the event arrival from the event generation subsystem, the probability of change detection termed as $\alpha$, the computed power consumption and lifetime of the node from the power manager subsystem, the utilization of the server, the energy consumption from the energy subsystem, the number of events departed from the queue, and the waiting time with the variation. The obtained results demonstrate the impact of the probability of change detection on the power consumption and the lifetime of the node. The probability of change detection is directly proportional to power consumption, wherein it is inversely proportional to the lifetime of the node. It can also be observed that the generated events are increasing. Also, the utilization of server is observed to be initially at 100 %. As the number of events increases and the waiting time of the events increases, the utilization of the server is affected.

![Figure 6](image-url) Impact of change detection and event arrival on lifetime and power consumption
Table 3 Analysis with the variation of mean arrival rate

<table>
<thead>
<tr>
<th>Arrival rate (mean)</th>
<th>Change detection ($\alpha$)</th>
<th>Power consumption (mW)</th>
<th>Lifetime (days)</th>
<th>No. of event arrival ($\lambda$)</th>
<th>Utilization of server</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.984</td>
<td>236.98</td>
<td>78.61</td>
<td>64</td>
<td>64%</td>
</tr>
<tr>
<td>2</td>
<td>0.9868</td>
<td>237.26</td>
<td>78.51</td>
<td>76</td>
<td>75%</td>
</tr>
<tr>
<td>3</td>
<td>0.9878</td>
<td>237.37</td>
<td>78.48</td>
<td>82</td>
<td>81%</td>
</tr>
<tr>
<td>4</td>
<td>0.9884</td>
<td>237.44</td>
<td>78.46</td>
<td>86</td>
<td>85%</td>
</tr>
<tr>
<td>5</td>
<td>0.9888</td>
<td>237.40</td>
<td>78.47</td>
<td>84</td>
<td>83.34%</td>
</tr>
<tr>
<td>6</td>
<td>0.9888</td>
<td>237.46</td>
<td>78.46</td>
<td>87</td>
<td>86.35%</td>
</tr>
<tr>
<td>7</td>
<td>0.9888</td>
<td>237.46</td>
<td>78.46</td>
<td>86</td>
<td>86.35%</td>
</tr>
<tr>
<td>8</td>
<td>0.9888</td>
<td>237.45</td>
<td>78.44</td>
<td>88</td>
<td>85.35%</td>
</tr>
<tr>
<td>9</td>
<td>0.9888</td>
<td>237.44</td>
<td>78.44</td>
<td>90</td>
<td>89.36%</td>
</tr>
<tr>
<td>10</td>
<td>0.9888</td>
<td>237.5</td>
<td>78.44</td>
<td>90</td>
<td>89.36%</td>
</tr>
</tbody>
</table>

Table 4 Analysis of the performance parameters affected by the change in event arrival and the probability of change detection

<table>
<thead>
<tr>
<th>Event arrival ($\lambda$)</th>
<th>Change detection ($\alpha$)</th>
<th>Power consumption (mW)</th>
<th>Lifetime (days)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.984</td>
<td>236.98</td>
<td>78.61</td>
</tr>
<tr>
<td>2</td>
<td>0.9868</td>
<td>237.26</td>
<td>78.51</td>
</tr>
<tr>
<td>3</td>
<td>0.9878</td>
<td>237.37</td>
<td>78.48</td>
</tr>
<tr>
<td>4</td>
<td>0.9884</td>
<td>237.44</td>
<td>78.46</td>
</tr>
<tr>
<td>5</td>
<td>0.9888</td>
<td>237.40</td>
<td>78.47</td>
</tr>
<tr>
<td>6</td>
<td>0.9888</td>
<td>237.46</td>
<td>78.46</td>
</tr>
<tr>
<td>7</td>
<td>0.9888</td>
<td>237.46</td>
<td>78.46</td>
</tr>
<tr>
<td>8</td>
<td>0.9888</td>
<td>237.45</td>
<td>78.44</td>
</tr>
<tr>
<td>9</td>
<td>0.9888</td>
<td>237.44</td>
<td>78.44</td>
</tr>
<tr>
<td>10</td>
<td>0.9888</td>
<td>237.5</td>
<td>78.44</td>
</tr>
</tbody>
</table>

(g) Number of events vs time  
(h) Average time

Fig. 6 Impact of change detection and event arrival on lifetime and power consumption (continued)
Table 3 assists in studying the effect of the variation of event arrival rate that follows the Poisson distribution. The number of events arrived per hour can vary depending on the application. If the application demands a fast arrival of the event, then a faster arrival rate is needed. Otherwise, for a certain application, where a change in the event occurs at a very slow rate, a slower arrival rate can be required. The effect of arrival rate on the lifetime and consumed power of the node is shown in Table 3, wherein the arrival rates are varied from 1 to 10. It reflects that the number of events generated increases with the change in the mean of the distribution and affects the service time. Thus, there is a variation in the utilization of the server. It can be observed that after the mean of 5, the probability of change detection is constant, indicating that there is not much deviation of values from the sensed and the stored value.

Table 4 also demonstrates the effect of the change in event arrival and the probability of change detection on performance parameters such as the lifetime and consumed power of the sensor node. The event arrival is varied in the range of 1 to 100, and \( \alpha \) (the probability of change detection) is in the range of 0 to 1. The results show that the minimum value of the probability of change detection has a maximum lifetime as the probability of change detection indicates the difference between the sensed value and the necessity to be computed. This stresses the importance of the analyser state to detect the change if occurred and to process it before it communicates to the neighbouring state. The results also show that the second hypothesis is proved as true as the analyser state detects any changes in the event from the stored event and also reduces the need for multiple computing/processing.

5. Conclusions

To achieve dynamic power management, the present work proposes a WSN node model based on semi-Markov model at the system level. Compared to the previous work, this implementation is more suitable in the stochastic environment or application, and is more advantageous as it proposes an improvement in performance without any changes in the hardware design. The developed model is based on queuing theory as the considered event arrival is stochastic in nature. The model is analysed for the power consumption and lifetime based on the arrival of events and the probability of change detection. From the obtained results, it is observed that the lifetime of the node is affected by the arrival rate, and the power consumption is affected by the probability of change detection. In future work, the aim will be to analyse the proposed model considering other performance parameters which can affect the lifetime and power consumption of the node. The proposed work can hence be extended to the network level.

Conflicts of Interest

The authors declare no conflict of interest.

References


