Modeling and Assessing the Power Consumption Behavior of Sensor Nodes using Petri Nets

Alaa E. S. Ahmed[1, 2]
College of Computer and Information Sciences
Al Imam Mohamed Ibn Saud Islamic University (IMSIU), Riyadh, Kingdom of Saudi Arabia[1]
Department of Electrical Engineering, Faculty of Engineering at Shoubra, Benha University, Egypt[2]

***Abstract*—Power Consumption is an influential and important concern in Wireless Sensor Networks (WSNs). Assessing and determining the impact of the power behavior of sensor nodes is an inclusive concern that should be executed at the network pre-deployment phase rather than post-deployment phase. Providing an accurate power modeling improves the development process of WSN applications and protocols. This paper introduces the use of Colored Petri Nets (CPNs) to model the power behavior and relations among different sensor node components when operated in an event driven environment. The objective is to figure out the overall power behavior of the nodes by considering power consumed accompanied with different states transition at a specific packets arrival and service rate values. Colored Petri Nets is a modeling language employed for validating and evaluating concurrent and distributed systems. The introduced model is beneficial since it provides the network designer with a way to determine surrogate designs or to check the compatibility of an existing power model behavior. The proposed model has been validated through the comparison of the power behaviors of two sensor nodes Mica2 and Telos. The results demonstrate the efficiency of using the proposed model to draw and analyze the WSNs energy consumption behavior.**

Keywords—Wireless sensor networks; power modeling; event-trigger; colored petri net; WSN protocols; distributed systems

I. Introduction

Wireless Sensor Networks (WSNs) are made up of spatially distributed motes with sensing, processing and communicating abilities. Maximizing the network life time is fundamental issue in developing WSN applications. Network life time is strongly related to the problem of optimizing the amount of consumed power throughout different stages executed by the network. It is the responsibility of the protocols or application designers to employ best strategies that lead to minimum power consumption. Power consumptions evaluation is considered as prerequisite for the deployment process to be able to avoid critical issues such as power consumption bottleneck, short network lifetime or higher cost maintenance due to replacing nodes with drained batteries.

Saving energy consumption in wireless sensor network can be achieved through the design of efficient strategies for scheduling the active and sleep moods of the node. Through these strategies, a time threshold value needs to be determined to allow optimal switching between the CPU sleep and active moods [1]. The benefit of this is to keep the cost of sensors replacement or maintenance as low as possible. However, having a power efficient designs need a good understanding of the power consumption behavior of the sensor nodes since they form the basic unit for constructing the WSNs structure. Understanding power consumption behavior can be addressed whether through real implementation, simulation or modeling.

The most accurate strategy of evaluating the power consumption is to use real measurement for the operated hardware. This could be done by applying battery level check pointing at regular period of time bases [2]. The cons of this procedure is the high cost due to the necessity of reproducing the environment, the huge number of nodes involved in the network and finally errors resulted from the measurement read by the human.

Modeling is another alternative way to evaluate the power consumption. This modeling process could be executed either analytically or by simulation. Even though results achieved from the modeling process is less accurate than those achieved by real measurement, it avails a flexible tool that help designers to evaluate and check various scenarios without affecting the actual environment. Modeling is considered an efficient way that enables the WSN designer to check and analyze these models in case of different states transitions [3, 4].

According to the previous argument, in this paper a method for modeling the power consumption behavior of sensor nodes is presented. A set of models has been defined to express power consumption accompanied with different state transition occurs within the sensor node component which are CPU, Transmission unit and sensing unit. The proposed models are implemented using Colored Petri Net [5–7] which represent a well-structured extension of Petri net that can be used in implementing and analyzing concurrency and asynchronous behaviors of distributed systems such as WSNs.

The remainder of the paper is organized as follows: In Section 2, the related work is introduced. Modeling bandwidth allocation scheduling strategies with CPNs is defined in Section 3. Section 4 shows a discussion for the simulation results. Finally, the paper ends the con collusion and future work in Section 5.

II. Related Work

Technology development and user requirements both are increasing in such a way that necessitate the designers of limited energy sensor devices to consider many challenges and requirements such as energy consumption, reliability, bandwidth allocation and data aggregation [8][9] [10]. Finding, modeling and analyzing strategies for predicting the WSN lifetime and optimizing the consumed energy is essential.

In literature, many approaches have been presented for modeling the power and energy consumption behaviors in embedded systems. In [11], Hyprid Automata is used in modeling TinyOs. TinOs is a component-based operating system designed for low power wireless devices. The provided automata model is used for analyzing the Node power dissipation based on its geographical location in the field area. Another model based on automata is presented in [12]. The aim of the generated model is to be used as a test beds for sensor network to provide priori information about the expected power consumed by the network.

In [13], the active and sleep operation modes are modeled using Markov models. The authors provided a Steady state equation that determines the number of packets needed when applying different modes of operation which in turn give indication about the system energy performance. Authors in [14] presented a method for predicting the energy level of nodes in WSNs using the Hidden Markov Model. They build the model based on the assumption that the energy level per nodes is divided into several levels and the state of these levels at any time is considered the random variable that can be used by Markov chain. The generated model performs training process while the WSN is executing their tasks so that an adjusted energy consumption prediction can be determined.

Another Markov based model is presented in [15]. They provide a model for the sensor node. They estimated the average life time in both triggers driven and duty cycle driven modes for different packets arrival time. The estimated value takes into consideration the effect of both communication and state transition overheads. Modeling using Markov chains is beneficial only in case the appropriate equation that describes the system is generated. However, the task of getting the appropriate equation can be time consuming and difficult in some cases.

Petri net is considered as an effective tool that can be used to model complex concurrent systems. Petri net is a directed graph where nodes consist of two components places and transitions. The directed arcs always connect places with transitions. The example shown in Fig. 1 consist of two places P0 and P1 and one transition T1. Places may contain tokens which used in firing the next transition connected to that place. In the figure, Place P0 contains one token that enable the transition T0. The enabled transition will fire either instantly or according to different timing parameter and the token will be moved to the output place P1. This behavior can be repeated in different ways to model the behavior of systems that utilize the flow of tokens to simulate a control mechanism adopted by different system processes.

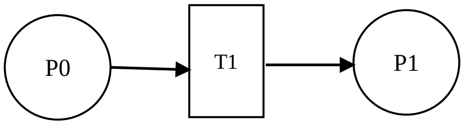

Fig. 1. Simple Petri Net Example.

Colored Petri Net (CPN) [16] is an executable model that combines the abilities of Petri Net and the abilities of high-level programming languages. It also provides the ability of providing time base models that can be used to simulate stochastic behaviors that can be utilized in evaluating concurrent systems performance. In literature, it has been used in modeling different aspects related to WSN. For example, in [17], authors used colored Petri net to validate and evaluate the WSN EQ-MAC protocol and they were able to validate its correctness. Average delay and packets delivery ratios are the two metrics used in evaluating their proposed model. Also, in [18] authors modeled an attack-resistant and efficient localization scheme using Petri net. They showed that there are no insecure states are reachable by adopting the state analysis methods. In [19] Petri nets were used to construct a formal model for the wireless sensor network. The model is utilized in testing and simulating the behaviors of different applications used by the network. They looked at the effect of QoS Parameters on the system from the application point of view. In [20] colored Petri nets are used to model the Wireless sensor network in order to capture the effect of different data aggregation techniques on the overall network performance. In this work the node behavior is represented by the M/M/1 queuing model. Results and analysis are implemented using the CPN tool.

In this paper, we handle the power consumption from the hardware point of view by providing a colored Petri net model that simulates the behavior and the interaction between these components.

III. Modeling of Sensor Node using Colored Petri Net

In this section, we propose a colored Petri net-based model to characterize the power consumption behavior of sensor nodes. The model utilizes the component-based architecture of node hardware components. As in Fig. 2, there are three main components included in the operation of the sensor node, Processing Unit (PU), Sensing Unit (SU) and Transceiver Unit (TU). These components operated in an event driven mode because every transaction executed by the sensor node was originally generated as a reaction to external events. Events are characterized by its time of occurrences which may follow different distributions. Events are classified into two categories source-driven or sink-driven. The first type "source-driven" is generated by either the surrounding environment or the neighbor nodes and then captured by the sensing unit of the sensor node. The second type "sink-driven" is generated by a query generated by the sink node. The proposed work deals only with the source-driven events type.

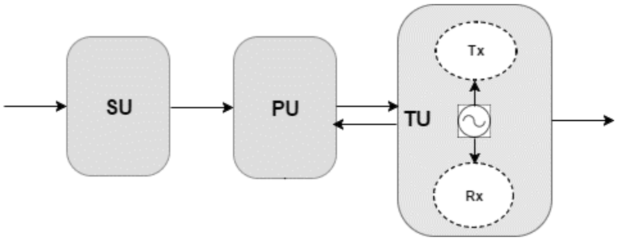

Fig. 2. Component in Operations of the Sensor Node.

A. State-Event-Transition Description

Here, a deeper look in the model dynamics is presented. These dynamics is specified in terms of sequencing and timing and using the state diagrams shown in Fig. 3. All components stay in a certain state until it is stimulated to change by an event. Each component passes over different power states. For example, the SU unit has two states on and off. The sensing process cycle starts when an event generated from the environment and read by the sensing unit (SU). Then, the event is forwarded to be processed by the processing unit (PU). The processing unit has three states, idle, sleep and run. Later, the event is either transmitted to or received from another node. The transceiver unit proceeds in five states, off, idle sleep, Tx and Rx which are connected according to the paths indicated in Fig. 2.

B. Modeling Power Consumption of a Sensor Node

In the Petri net model, places are used either to represent the power state of a specific component of the sensor node or to emulate a particular event-driven situation. In the other hand tokens are also employed to indicate either events or communication channel allocations. When tokens migrate among different places, it causes different system states to occur. The node model is stimulated by receiving sensed data from the environment emulated by the workload generator model shown in Fig. 4. When transition T1 fires, data is stored in Place "New Arrival" and a token goes back to the "Data Sampling" place to allow transition T1 to fire again to receive another generated packet. The process of Firing transition T1 is repeated randomly and according to an exponential distribution attached to the arc E1 (Table I). After the packets generated, they are formatted using a function attached to arc E2 (Table I). The packet format contains a record with three fields: Data reading "Rvalue", Initial consumed power "RPower" and "Type" which indicates whether the processed packet is to be received or to transmited mode. Table I shows arcs inscriptions used by the packets generator module.

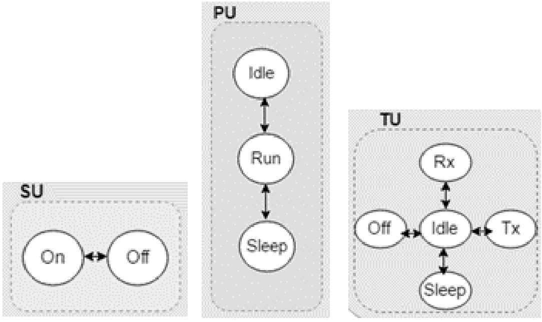

Fig. 3. Event-Driven State Transition Diagram of Sensor node Components.

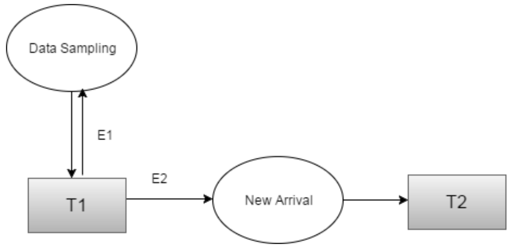

Fig. 4. Packets Generator Module.

TABLE I. SHOW THE ARC INSCRIPTIONS IN PACKETS GENERATOR MODULE

ARC name	inscription
E1	()@+expTime(ArrivalTime)
E2	fun newX() = {Rvalue = randRange, Rpower = 0, Type =0}

Fig. 5 shows the Petri net model of SU unit where initially the unit is in "off" state, and there is a token reside in place "Off-State". When data packets arrived at transition T2, it is fired and a token is put in "On State" place to indicate that the unit is "on" state. As a result, T3 transition fires and forward the received data packets to the PU module through the arc E7. A token is returned back to the "Off State" place to convert the unit to "off" state again and then wait for next packets arrival.

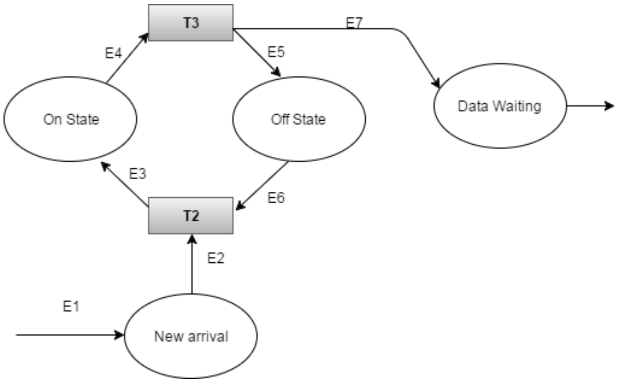

Fig. 5. SU Model.

Fig. 6 illustrates the PU model which has nine places and six transitions. The model is an implementation to the different states mentioned in Fig. 2. Initially, the unit is in "sleep" state which is implemented by having a token reside in place "Sleep". The PU unit switches to the "RUN" state to process the data and a token is generated at place "ON". PU stays in the "RUN" state as far as there are tokens exist at place "Buffer PU". The "Counter" place is used to indicate the number of tokens on the buffer. Transition T9 is fired when the "Counter" place indicates zero packets in the buffer and PU switch to "sleep" state to preserve power.

The PU model executed using the following steps:

1) When data packets arrive at transition T4, it fires and tokens are generated and forwarded to both "Buffer PU" and "Only One" places. Since the PU was previously in the sleep state, transition T7 fires and a token will be generated and move to the "ON" place to indicate that the PU unit is now switched from sleep to Run state. Transition T7 has a guard function used to add up the delay and the power consumed values due to the move from "Sleep" to "Run" state. These values are determined from the sensor node hardware specification data sheet as in Table III.

2) Now, there is a token in both "Buffer PU" and "ON" places and, the "idle" place has another token by default. This combination enables the transition T5 and a token will be moved from the "idle" place to the "run" place to announce the start of the Run state. As a consequence, Transition T8 fires and triggers the execution of a function that simulates the PU service rate. The service rate is represented in terms of a randomly generated delay with an exponential distribution. After this delay is ended up, two tokens will be generated. The first token is moved back to the “idle” place to allow the next arrived packet to be processed. The second moves to the TU unit and per its type (Send or Receive), the TU will perform the appropriate action.

3) In case another packet arrived and the PU is still ON, the previous steps will be repeated.

Fig. 7 illustrates the TU model which has nineteen places and thirteen transitions. The TU unit receives input from two sources. The first is the data that comes from the PU unit after has been sensed by the SU. In this case, the TU unit acts as a transmitter to send the data to the next node. The second is the data comes directly from other neighboring nodes to be processed by the PU and then retransmitted to other neighboring nodes.

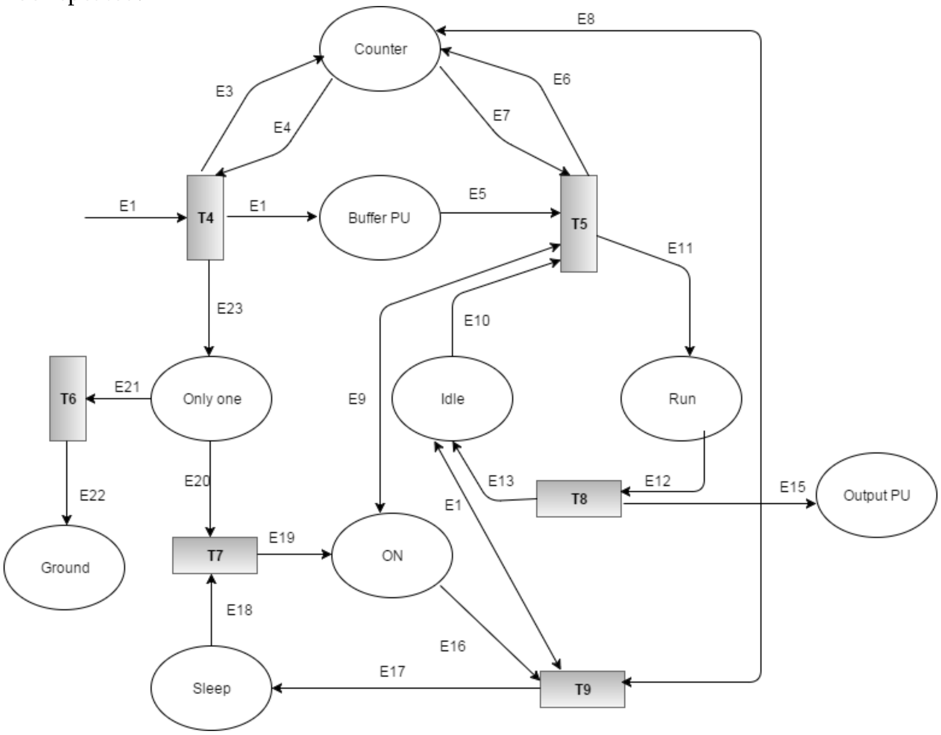

Fig. 6. PU Model.

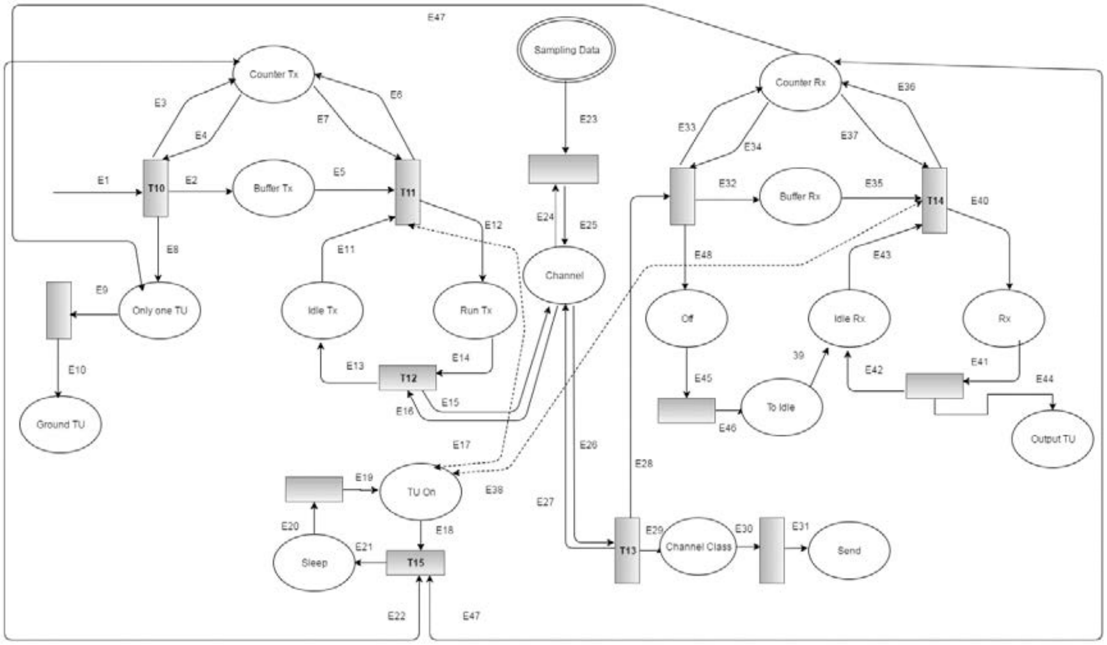

Fig. 7. SU Model.

The TU model executed using the following steps:

1) When data packets comes from the PU unit, Transition T10 fires and tokens generated and moved to the place “Buffer Tx”. The place “Counter Tx” is used to count the number of packets received by the TU to control the firing rate of transition T11. As long as there is a token in the “buffer Tx” place, T11 is enabled. There are default tokens in the place “idle Tx” to indicate that the TU is initially in the idle state. When T11 fires, a token move from the “idle Tx” place to the “Run Tx” place. In other words, the TU switches from “idle” to “Tx” or transmission state. Transition T12 fires and a randomly exponentially generated delay is added up to the total delay of the packets. Also, power consumed by the Tx is added to the total power consumed due to the packet transmission.

2) Now, Tokens generated and moved to the places “idle Tx” and “Channel. “Channel” place is used to implement the channel interface used in transmitting the packets. Tokens moved to the “idle_tx” are used to enable the receiving of other new arrived packets. While tokens moved to the place “Channel” place are characterized by type “T” that stands for “Transmit” so that they will be forwarded outside the node. The place “Channel” receives another input token as a result of receiving packets from other neighboring nodes. Packets related to these tokens are marked as “R” that stands for “Receive” to be forwarded to the PU unit. Packets Types classifications is executed at transition T13. The “T” type tokens are moved to the places “Channel Class” and “send”. The “R” type tokens moved to the “Buffer Rx” place to enable the Rx state.

3) When “R” types tokens exist in the “buffer Rx” place, the “Counter Rx” indicates their counts and at the same time fires transition T14. Since by default the receiving part of the TU is idle state, i.e. there are tokens in the “idle Rx” place, place “Rx” fires and tokens move to the “Output Tu” place.

4) TU unit switch from the idle state to the sleep state when there are no tokens in both “Buffer Rx” and “Buffer Tx”. This transition implemented in the model using

5) Transition T15 that will be enabled when both the counter Tx, counter Rx is zero and the token in the “TU On” place move to the “Sleep” place.

6) The transition from the “off” state to the “idle” state is implemented in the model using the places “off” and “To idle”. When tokens received in the “buffer Rx”, the token is generated and is passed to the “idle” place which consequently moved to the “to idle” place and finally reach the “idle Rx” place to indicate that the current state is the idle state. This step is done to prepare the node to switch to the idle to either Rx or Tx.

IV. Power Consumption Evaluation of the Proposed Model

In this section, we evaluate the performance of the proposed model using the CPN tool. The evaluation process is conducted by considering two different nodes architecture telos and Mica 2. We call the first one SN1 and the second SN2. SN1 is composed of microcontroller-MSP430F4794 to act as a PU, a transceiver-CC2420 to act as a TU, and a temperature sensor called Dallas Semi.DS1820 to act as a SU. SN2 is composed of ATmega 128L to act as a PU, component CC1000 to act as a TU and the same temperature sensor used in SN1. Table II summarize he components parameters and the consumed power due to states transition [21].

The simulation parameters used in the evaluation process in case of different workload conditions are listed in Table III.

TABLE II. Performance Parameters and Power Consumption and State Transition Time Parameters for SN1 and SN2

Node Architecture			**Value**	
Component		***State***		
Sensor Node			***SN2***	***SN1***
Component Power consumption	PU	Idle	6 mW	3.9 µW
		Run	15 mW	1.2 mW
		Sleep	24 µW	0.3 µW
	SU	ON	3 mW	3 mW
		Off	0 W	0 W
	TU	Idle	28.8 mW	14.058 mW
		Tx	49.5 mW	62.04 mW
		Rx	28.8 mW	57.42 mW
		Sleep	3 µW	66 µW
		Off	0.6 µW	0.066 µW
Time of Transitions between states	PU	Idle to Run	5 µs	6 µs
		Run to Idle	5 µs	6 µs
		Idle to Sleep	40 µs	50 µs
		Run to Sleep	40 µs	50 µs
		Sleep to Run	65 ms	80 ms
	SU	On to Off	36 µs	36 µs
		Off to On	400 ms	400 ms
	TU	Tx to Idle	2 µs	2 µs
		Idle to Tx	270 µs	192 µs
		Rx to Idle	2 µs	2 µs
		Idle to Rx	250 µs	192 µs
		Sleep to Idle	200 µs	600 µs
		Idle to sleep	250 µs	192 µs
		Off to Idle	2.2 ms	1 ms

TABLE III. Simulation Parameters

Parameter Configuration	**Value**	
Sensor Node	SN2	SN1
Simulation Time	50 sec	50 sec
Packet arriving distribution	1 to 5 sec	1 to 5 sec
Service time in PU	Exponential distribution □=1	Exponential distribution □=1
Sensor sampling period	5 sec	5 sec

The simulation process is conducted by computing the node overall power consumption by adding up different states power during the simulation time. As time proceeds, the average number of tokens deposited in various places is computed to indicate the percentages of time that packets occupy different states. The power consumed per state is computed by multiplying the computed percentage by the state power values listed in Table II. The summation of power values of different states gives the node overall value. Packets arrival rate, components services rates and time delay between different states affect and control the number of tokens exist in places and consequently the value of the computed percentages.

Results in Fig. 8, 9 and 10 evaluate the effect of multiple workloads on the power consumed by the node components on different states. As the packets inter-arrival time increases the arrival-rate decreases and the power consumption becomes lower. These results can be explained as when sensor nodes exist near active areas of the WSN such as the sink node and the cluster heads, the inter-arrival rate of packets arrival and the energy consumed increases. These experiments could be used to evaluate the effect of changing the sampling rates on the power consumed by the sensor node components. In Table IV, the power evaluation in SN1 with different inter-arrival time is shown.

TABLE IV. SIMULATION RESULTS OF POWER CONSUMPTION IN SN1

Component	**States**	**Arrival Time (1 sec)**	**Arrival Time (3 sec)**	**Arrival Time (5 sec)**
PU	Idle	0.8	0.41	0.22
	Run	180	86.4	48
	Sleep	0.04	0.021	0.021
TU	Idle	3753	1743	815
	Tx	9306	4466	2481
	Rx	6890	3100	1148
	Sleep	17.5	8.18	3.83
	Off	0.007	0.003	0.001

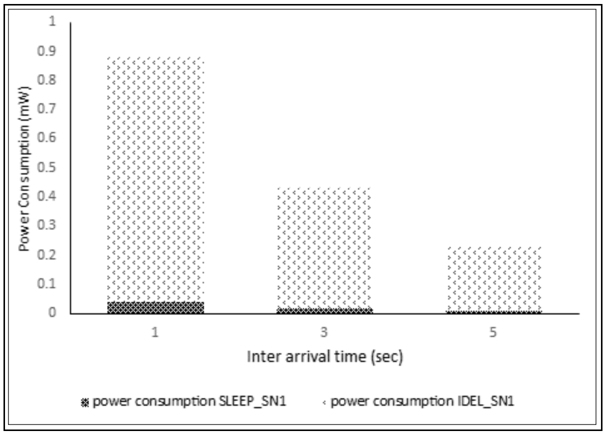

Fig. 8. PU Power Simulation for SN1 (SLEEP and IDLE States).

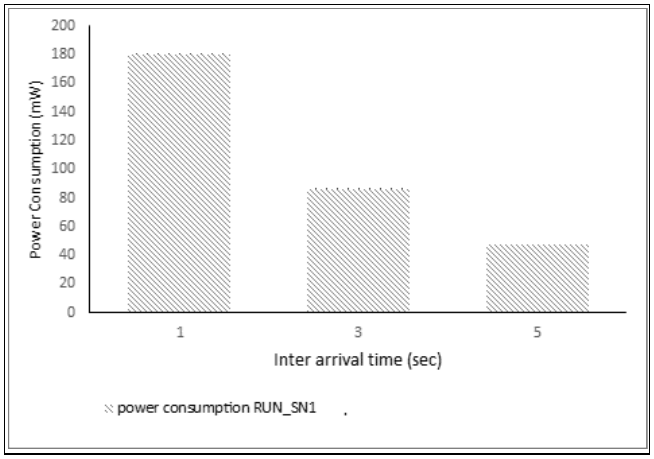

Fig. 9. PU Power Simulation for SN1 (Run State).

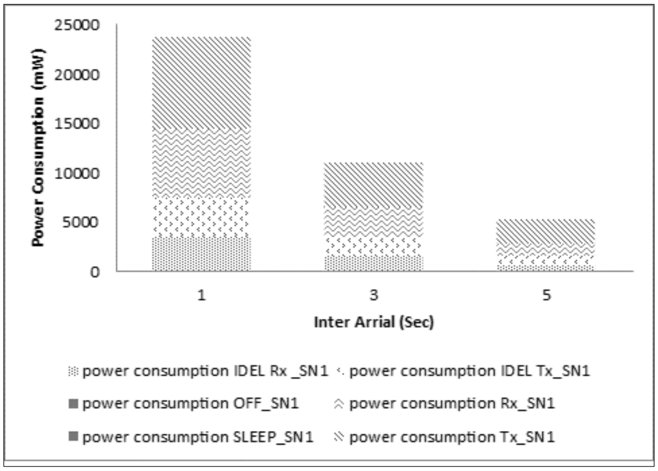

Fig. 10. TR Power Simulation for SN1.

To evaluate the effect of node architecture on power consumption level, the experiment repeated for another node architecture termed SN2. Fig. 11 and 12 show the power consumed in all components whereas the inter arrival rate increases the power consumed by the node increases.

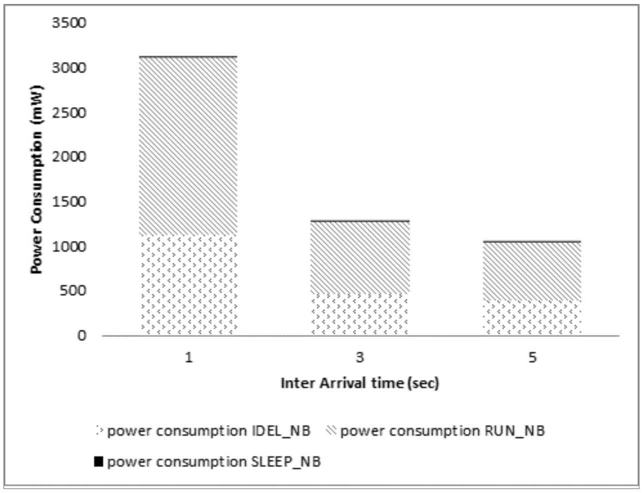

Fig. 11. PU Power Simulation for SN2.

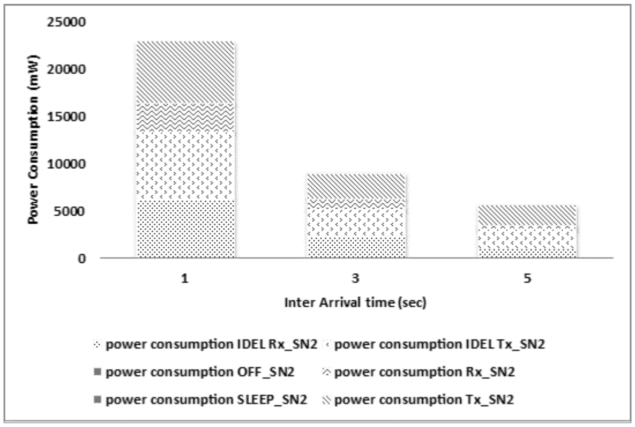

Fig. 12. TU Power Simulation for SN2.

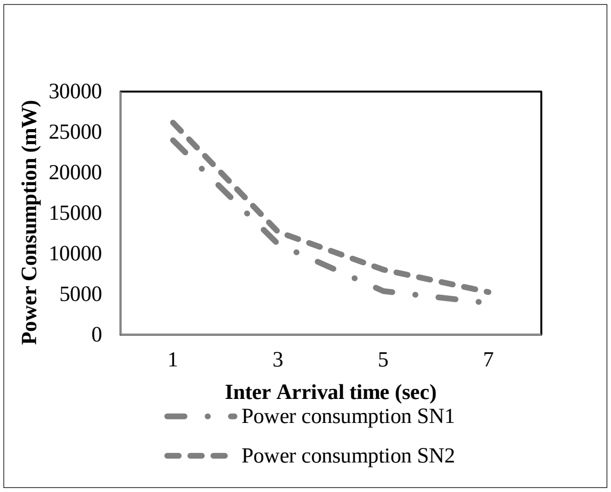

Fig. 13. Power Consumption in SN1 and SN2.

Table V shows the values of these results under different simulation conditions. Comparing the power consumption of SN1 and SN2, Node SN1 has less power consumption than the SN2 as shown in Fig. 13. This is due to the main functional components of SN1 have more ability to operate in low power situation than SN2. The model can be used to characterize the power behavior of various sensor nodes architectures and guide the network designers toward using the architecture that fit his needs.

Assuming another WSN node (termed SN3), which has different PU component ARM SA-1100. Table VI summarize the components parameters and the consumed power due to states transition which obtained from the datasheets of SN3 [21].

In Table VII, Fig. 14 and Fig. 15, the power evaluation of SN3 with different inter-arrival time is shown. From these results, we notice that most the power is consumed in the TU unit However, in Node SN1, most of the power consumed in the PU unit. This result indicates the ability to use the proposed model to help WSN designers in understanding the power consumption behavior for different sensor node architectures. Comparing the power consumption in SN1 and SN3, Node SN1 has less power consumption than the SN3 as shown Fig. 16.

TABLE V. SIMULATION RESULTS OF ENERGY CONSUMPTION IN NS2

Component	**States**	**Arrival Time (1 sec)**	**Arrival Time (3 sec)**	**Arrival Time (5 sec)**
PU	Idle	1128	468	390
	Run	1980	810	660
	Sleep	2.73	1.2	1.05
SU	On	90	54	102
	Off	0	0	0
TU	Idle	6710	2592	1555
	Tx	6543	2673	2178
	Rx	7113	2822	2016
	Sleep	0.69	0.27	0.16
	Off	0.06	0.022	0.007

TABLE VI. PERFORMANCE PARAMETERS AND POWER CONSUMPTION AND STATE TRANSITION TIME PARAMETERS FOR SN3

Node Architecture			**Value**
Component		***State***	
Sensor Node			***SN3***
Power consumption	PU	Idle	50 mW
		Run	230 mW
		Sleep	0.16 mW
	SU	ON	3 mW
		Off	0 W
	TU	Idle	14.058 mW
		Tx	62.04 mW
		Rx	57.42 mW
		Sleep	66 μW
		Off	0.066 μW
the transition time between states	PU	Idle to Run	10 μs
		Run to Idle	10 μs
		Idle to Sleep	90 μs
		Run to Sleep	90 μs
		Sleep to Run	160 ms
	SU	On to Off	36 μs
		Off to On	400 ms
	TU	Tx to Idle	2 μs
		Idle to Tx	192 μs
		Rx to Idle	2 μs
		Idle to Rx	192 μs
		Sleep to Idle	600 μs
		Idle to sleep	192 μs
		Off to Idle	1 ms

TABLE VII. SIMULATION RESULTS OF POWER CONSUMPTION IN SN3

Component	States	Arrival Time (1 sec)	Arrival Time (3 sec)	Arrival Time (5 sec)
PU	Idle	9150	5600	2750
	Run	29440	17940	8740
	Sleep	17.92	11.2	5.76
SU	On	108	126	60
	Off	0	0	0
TU	Idle	3092.76	1602.612	787.248
	Tx	7941.12	4839.12	2357.52
	Rx	5397.48	2181.96	1148.4
	Sleep	14.52	7.524	3.696
	Off	0.006204	0.002508	0.00132

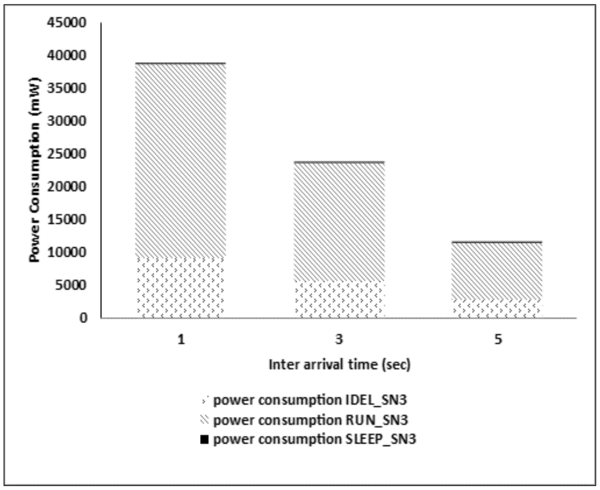

Fig. 14. PU Power Simulation for SN3.

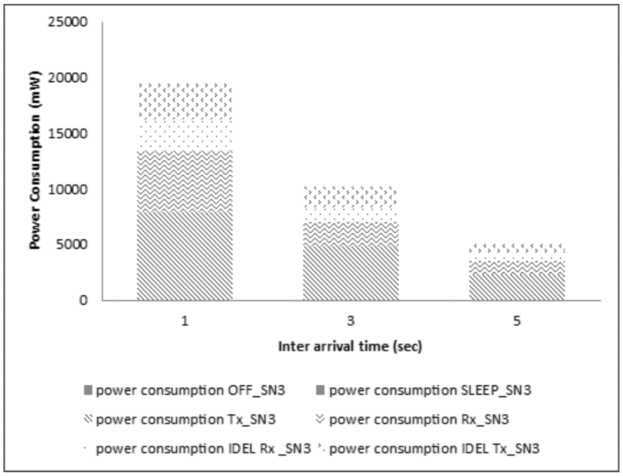

Fig. 15. TU Power Simulation for SN3.

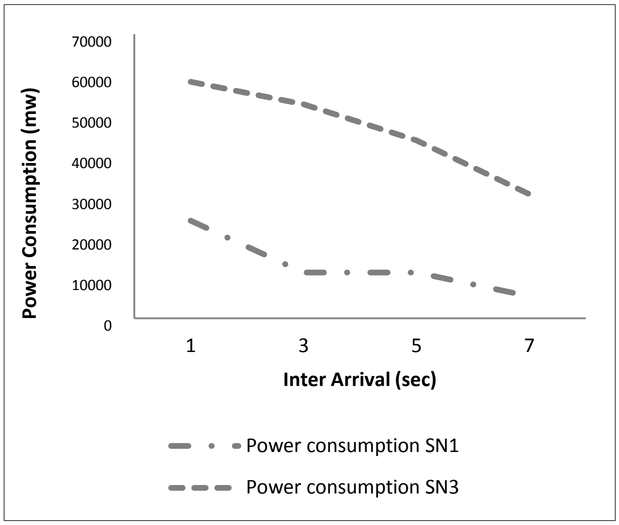

Fig. 16. Power Consumption in SN1 and SN3.

V. CONCLUSION

In this paper, we provided a petri net model of power consumption of WSN node that is described through the event-driven. The model has been evaluated by emulating the power behaviors of two real sensor nodes Mica2 and Telos. The results demonstrated the efficiency of using the proposed model to draw and analyze the WSNs power consumption behavior. Thus, the provided results and analysis showed that the proposed model is valid and easy to debug and evaluate the power consumption for different WSN node architectures. As a future work, the proposed model can be considered as an initial template where other performance parameters could be employed. For example, the effect of data aggregation, compressive sensing and bandwidth allocation on power consumption can be modeled and evaluated.

REFERENCES

[1] J. Liu and P. H. Chou, “Idle energy minimization by mode sequence optimization,” ACMTransactions on Design Automation of Electronic Systems, vol. 12, no. 4, article 38, 2007.

[2] F. A. Katsriku, J. Abdulai, K. S. Adu-Manu, and F. K. Banase, "Prolonging the Lifetime of Wireless Sensor Networks: A Review of Current Techniques,", Wireless Communications and Mobile Computing, vol. 2018, 2018.

[3] H. P. Xie, H. Y. Zhou, D. C. Zuo, and P. Zhou, “Energy optimization and modeling in wireless sensor networks: a survey,” Computer Science, vol. 10, pp. 15–20, 25, 2012.

[4] N.R. Roy, P. Chandra, "Energy dissipation model for wireless sensor networks: a survey," International Journal of Information Technology, 2019.

[5] K. Jensen, L. Kristensen, "Coloured Petri Nets: Modelling and Validation of Concurrent Systems," Springer: New York, NY, USA, 2009.

[6] K. Jensen, "Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use," Springer Verlag: London, UK, 1995.

[7] K. Jensen, L. M. Kristensen, "Colored Petri nets: a graphical language for formal modeling and validation of concurrent systems," Communications of the ACM, vol. 58, no. 6, pp. 61-70, 2015.

[8] K. Alshammari and A E. S. Ahmed, "An efficient approach for detecting nodes failures in wireless sensor network based on clustering," International Symposium on Networks, Computers and Communications (ISNCC), Marrakech, pp. 1-6, 2017.

[9] A. Dˆamaso, D. Freitas, N. Rosa, B. Silva, and P. Maciel,"Evaluating the power consumption of wireless sensor network applications using model," Sensors vol.13, no.3, pp. 3473–3500, 2013.

[10] E. Troubleyn, J. Hoebeke, I. Moerman, and P. Demeester, "Broadcast aggregation to improve quality of service in wireless sensor networks," International Journal of Distributed Sensor Networks, vol. 10, no. 3, 2014.

[11] S. Coleri, M. Ergen, and T. J. Koo, “Lifetime analysis of a sensor network with hybrid automata modelling,” in Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA '02), pp. 98–104, ACM, New York, NY, USA, September 2002.

[12] S. Kellner, M. Pink, D. Meier, and E. O. Blaß, “Towards a realistic energy model for wireless sensor networks,” in Proceedings of the 5th Annual Conference on Wireless on Demand Network Systems and Services (WONS '08), pp. 97–100, January 2008.

[13] Y. Zhang and W. Li, “An energy-based stochastic model for wireless sensor networks,” in Proceedings of the Wireless Sensor Network, vol. 10, pp. 322-328, 2011.

[14] P. Hu, Z. Zhou, Q. Liu, and F. Li, “The HMM-based modeling for the energy level prediction in wireless sensor networks,” in Proceedings of the 2007 2nd IEEE Conference on Industrial Electronics and Applications (ICIEA ’07), pp. 2253–2258, May 2007.

[15] D. Jung, T. Teixeira, A. Barton-Sweeney, and A. Savvides, “Model-based design exploration of wireless sensor node lifetimes,” in Proceedings of the 41rth European Conference on Wireless Sensor Networks (EWSN ’07), 2007.

[16] K. Jensen, L. M. Kristensen, and L. Wells. "Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems," International Journal on So.ware Tools for Technology Transfer vol. 9, no.3-4, pp. 213–254, 2007.

[17] J. Ben-Othman, S. Diagne, L. Mokdad, B. Yahya, "Performance evaluation of a hybrid MAC protocol for wireless sensor networks," international conference on Modeling, analysis, and simulation of wireless and mobile systems, Pages 327-334 , ACM, 2010.

[18] D. He, L. Cui et, al. Design and Verification of Enhanced Secure Localization Scheme in Wireless Sensor Networks. IEEE Transactions on Parallel and Distributed Systems, vol. 20, pages 1050 – 1058, IEEE, 2009.

[19] X. Fu, Z. Ma, Z. Yu, and G. Fu, “On wireless sensor networks formal modeling based on petri nets,” in Proceedings of the 7th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '11), pp. 1–4, 2011.

[20] A. E. S. Ahmed and M. E. A. Ibrahim, "Colored Petri Net Models for Clustered and Tree-Bases Data Aggregation in Wireless Snsor Networks", International Journal on Information Technologies & Security, vol 10, no. 3, 2018.

[21] J. Li,1,2 H. Y. Zhou, et, al, “Energy Consumption Evaluation for Wireless Sensor Network Nodes Based on Queuing Petri Net” , International Journal of Distributed Sensor Networks, Vol. 10, no. 4, pp. 1-11, 2014.