UWL REPOSITORY

repository.uwl.ac.uk

An SRN-based model for quantitative evaluation of IoT quality attributes

Goudarzi, Shidrokh ORCID: https://orcid.org/0000-0003-0383-3553, Sanahmadi ahmadi, Arman and Abdollahi Azgomi, Mohammad (2023) An SRN-based model for quantitative evaluation of IoT quality attributes. Internet of Things, 23. ISSN 2543-1536

http://dx.doi.org/10.1016/j.iot.2023.100894

This is the Accepted Version of the final output.

UWL repository link: https://repository.uwl.ac.uk/id/eprint/10273/

Alternative formats: If you require this document in an alternative format, please contact: open.research@uwl.ac.uk

Copyright: Creative Commons: Attribution-Noncommercial-No Derivative Works 4.0

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy: If you believe that this document breaches copyright, please contact us at open.research@uwl.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

An SRN-Based Model for Quantitative Evaluation of IoT Quality Attributes

Arman Sanahmadi[1], Mohammad Abdollahi Azgomi[2,*] and Shidrokh Goudarzi[3]
[1,2] Trustworthy Computing Laboratory, School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran
[3] School of Computing and Engineering, University of West London, St Mary's Road, Ealing, London W5 5RF

E-mail: arman_sanahmadi@comp.iust.ac.ir, azgomi@iust.ac.ir, shidrokh.goudarzi@uwl.ac.uk

Abstract

Today, the Internet of things (IoT) is widely used in various fields, including health control, smart cities, intelligent buildings, and so on. One of the severe concerns in IoT systems is the issue of energy consumption and its management. IoT systems have limited energy resources, and in this regard, these limited resources must be managed appropriately. To design and build IoT systems, various aspects such as usable chips, types of communication protocols, timing of sending and receiving data, and so on, directly affect the system's energy consumption. Therefore, it is necessary to model and evaluate the energy consumption of IoT systems before building and implementing the system. Using an appropriate model makes it possible to investigate and understand how much the system consumes energy and how it is in conformity with the system's demands. This paper presents a stochastic reward net (SRN)-based model for modeling and quantitative evaluation of system energy consumption. To solve and evaluate the model, the proposed model is converted into an SRN model based on a series of automatic transformations. The proposed model is used in a case study to show how the model works and the results are given in the paper.

Keywords: Internet of things (IoT), power management, quantitative evaluation, stochastic reward nets (SRNs).

1. Introduction

Today, billions of Internet of things (IoT) nodes are connected to the Internet, and this number is constantly increasing. These devices continuously exchange data with each other. The communication between nodes in these systems relies on a lightweight protocol. Due to the advancement of technologies such as 5G and the number of nodes, the discussion of data management and energy management in these systems has become a major challenge [1]. IoT systems are widely used in various fields, such as health control, smart cities, smart homes, etc. Most of these systems have a limited or unreliable energy source, which shows that the energy management issue is critical [2].

The primary function of an IoT system is to collect data on a reliable platform and then transmit it to a sink node for data collection and processing. Then, based on data processing, decisions are made to perform system tasks. On the energy consumption of an IoT system, various factors such as the type of hardware, software, and the selected communication protocol will be practical. These factors must be considered in conjunction, as optimizing one factor may impact others. IoT systems need energy to perform various tasks such as data processing, sending/receiving, and changing the system's operational state. The highest energy consumption is related to data transmission. Therefore, to manage an IoT system's energy, we should first identify energy consumption points and incorporating them into energy evaluation and management [1][3].

Another essential aspect in the field of energy consumption management in IoT -based systems is the working modes of the node at different times and under varying conditions of the node. Each node has different working modes. For example, the node may have a sleep mode that is activated under specific conditions to reduce energy consumption [4].

In many scenarios, accessing the IoT system's nodes is complex or even impossible, and this issue can be a severe challenge. If the system has a defect, it can lead to high costs in the operating environment. Therefore, evaluating and modeling energy consumption during the system design phase can reduce many operating environment costs. One of the ways to investigate energy evaluation in computing systems is to use formal modeling and quantitative evaluation methods, which are discussed in [5][6]. The initial design will be modeled and evaluated using this model to determine if the system meets the requirements. Another advantage of using modeling is examining the change of parameters and its effect on the performance of different system parts. One of the problems in IoT systems is that the

* Address for correspondence: School of Computer Engineering, Iran University of Science and Technology, Hengam St., Resalat Sq., Tehran, Iran, Postal Code: 16846-13114, Fax: +98-21-73225322, E-mail: azgomi@iust.ac.ir.

model which is specific to these systems does not exist in the literature. Even if such a model exits, it has defects, such as a lack of supporting tools. To the best of our knowledge, no research in IoT applications has been focused on extending the stochastic reward nets (SRNs) to introduce a high-level model specifically for modeling and quantitatively evaluating energy consumption. The reasons for choosing SRN over other methods are as follows:

1. We were looking for a model that can be solved analytically, its solution method is precise, and simultaneously, it can be simulated. The SRN model and most distributions of the Petri model have an analytical solution method, and they also have the possibility of solving the transient and steady state of the studied system.
2. Petri nets provide a formal and graphical representation, making it easier to model and comprehend the system.
3. Among Petri net distributions, it was possible to use GSPN, SAN, etc. However, we were looking for a distribution that would provide the modeler with more features to make modeling more accessible. For example, the existence of the guard function in SRN has made modeling easier than in GSPN. Also, the model used is more well-known and has a complete tool that supports it.
4. There is an efficient tool that supports SRN models called SPNP. This tool can receive the model file and produce its analytical results. The core of the SPNP tool is used to support the SB-SRN model in the tool presented in this article. The tool presented in the article receives the SB-SRN model, then runs the conversion rules presented in the article on it and converts it into an SRN model. Finally, it runs the SRN model automatically in the SPNP tool for solving and evaluation.

One of the deficiencies in the field of modeling IoT systems is the absence of a specific modeling method for these systems with the appropriate tool. This article presents a method based on Petri nets for modeling and quantitative evaluation of IoT systems. The presented modeling method is based on the SRN extension. The following are the contributions of this paper:

1. We propose a model in which the concepts needed to evaluate the IoT systems' energy consumption are included. These concepts encompass energy source and source-dependent transitions, aiming to reduce the complexity of energy consumption modeling.
2. We present a model that does not depend on a specific case study, and the presented model is general.
3. We propose the formal definition and method of solving and evaluating the model.
4. We develop a tool that supports the model, enhancing its practical implementation.
5. We examine the utility of the model through a case study.

The rest of this paper is organized as follows: Section 2 reviews related work. Section 3 explains fundamental concepts. In Section 4, the proposed method is explained. Section 5 presents case studies and model evaluation. Finally, some concluding remarks and future works are given in Section 6.

2. Background

In this section, the basic concepts that are needed to better understand the article are given.

2.1. Internet of things

Ashton first introduced the concept of Internet of things (IoT) in 1999. In this definition, the IoT is a collection of objects connected through RFID, each with a unique identity. The general definition of the IoT is “a dynamic network infrastructure that has the capability of self-configuration based on standards and protocols; Objects have identities and can use intelligent interfaces and integrate as an information network.” From another point of view, the IoT consists of two words, “Internet” and “Things,” which means connecting objects from the Internet for use in various applications [1].

Usually, the architecture of an IoT system is presented as a service-oriented architecture, which consists of four layers. Figure 1 shows this architecture. Each of these four layers is [7]:

- Sensing Layer: It includes hardware objects responsible for measuring the environment and collecting information. The challenges of this layer are object size, cost, energy consumption, object heterogeneity, and maintenance.
- Network Layer: The network layer, whether wired or wireless, facilitates communication between objects. This layer's challenges are optimizing energy consumption and the variety of different protocols.
- Service Layer: It creates and manages services based on user requirements. Challenges within this layer include service composition, service discovery, and trust management.
- Interface Layer: This layer provides the interface through which users can communicate with the services.

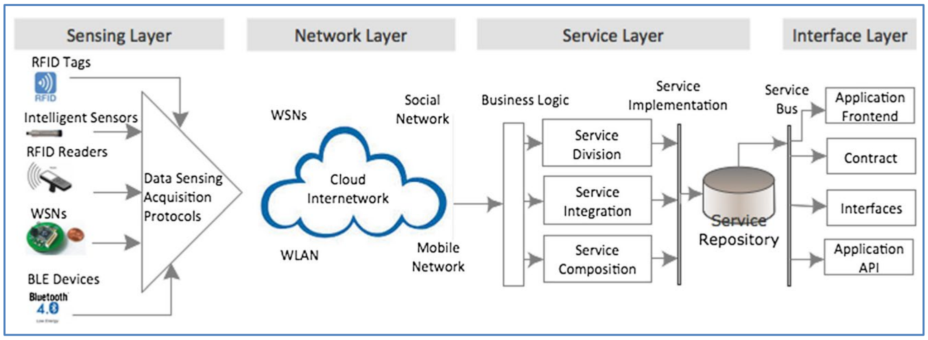

Figure 1. The layers of IoT system [7]

2.2. Energy consumption in IoT

Figure 2 shows the general model of different parts of energy consumption in an IoT system's node. As shown in Figure 2, energy consumption management has three main parts. Which include [8]:

- The first whose task is to collect energy and convert it into usable energy for the system.
- The second part is responsible for maintaining and supplying energy (like a battery).
- The third part consumes the available energy for various tasks such as sensing, processing, and so on.

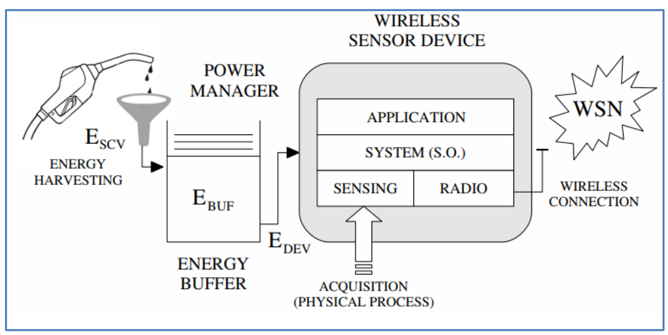

Figure 2. The energy consumption management system [8]

A node of the IoT system or wireless sensor network needs energy to perform various activities, which we will explain below [8]:

- Sending/receiving a packet: which is equal to the probability of the transmitter being active multiplied by the energy consumption of sending a packet multiplied by the service rate
- Waking up the node: the amount of power required to wake up the node
- Being in idle mode: the energy consumption in this mode multiplied by the time in this mode
- Being in sleep mode: the energy consumption in this mode multiplied by the time in this mode
- Packet processing: packet processing rate multiplied by the energy consumption for processing a packet

2.3. Stochastic reward nets

Petri net is a formal modeling language for the description of distributed systems that are represented graphically by a directed bipartite graph [9][10]. Many extensions have been developed for Petri nets, one of which is the stochastic reward network (SRN). The SRN is used to model and analyze computing and complex systems [11].

This formalism, like Petri nets, has two main components called *place* and *transition*. Each transition is enabled or disabled by a guard function. Transitions can be timed or immediate. This formalism is supported by SPNP [12]. A stochastic reward network is defined as a 4-tuple (*P, IT, TT, GF*), where [11]:

- *P*: the set of places
- *IT*: the finite set of immediate transitions

- *TT*: the finite set of timed transitions
- *GF*: the finite set of guard functions. If *gf* is a member of *GF* and has m inputs, this function is defined as $gf: N^m \rightarrow \{true, false\}$

The SRN formalism provides a powerful modeling environment for dependability, performance, and performability analysis. SRN*s* substantially increase the modeling power of generalized stochastic Petri nets (GSPNs) by adding guard functions, marking dependent arc multiplicities, general transition priorities, and reward rates at the modeling level [12]. The model can be solved using the SPNP tool. This tool works by converting the input SRN model into a Markov reward model, then by solving the MRM, transient state, steady state, cumulative and sensitivity criteria can be extracted.

Considering that the SRN formalism has been introduced to evaluate the measurable criteria of computational and complex systems and also the existence of the powerful SPNP tool to support the SRN model, in this article the SRN model has been used as the basic model of the presented work.

3. Related work

This section reviews the related works in modeling and evaluating energy consumption in IoT systems.

- Ever et al. [4] evaluated the availability, efficiency, and energy consumption of IoT systems using an analytical model. This model considers different states, such as sleep/on mode, node failure, and communication channel failure. It also considers various energy consumption factors, such as packet sending, receiving, and power-on processes. The results are obtained using two different approaches: the spectral expansion solution method and the system of simultaneous linear equations. Both analytical solution methods yield results that align closely with the simulation results, with a maximum difference of less than 0.1 percent. Furthermore, the developed models have demonstrated that implementing an on-demand sleep scheduling scheme can lead to savings of up to 70% in idle power consumption. Based on the research conducted on the results, it is possible to determine the range of optimal sleep performance suitable for the sleep schedule technique based on traffic load. The computational nature of analytical models allows us to study various thresholds related to the system.
- Lages et al. [5] has presented the energy consumption modeling of an LPWAN-based system using the SPN extension. Considering that in IoT systems, the most energy consumption is in the physical layer and at the time of sending and receiving data, the authors focus on the energy consumption at the time of sending and receiving data according to the size of the packets. In this article, the discussion of packet loss has also been seen. The initial work considers a star topology, and the experimental results show that the model can accurately capture the energy consumption in the physical layer with an error of less than 2%.
- Ruiz et al. [13] discuss the modeling and evaluation of an energy-aware algorithm using the timed colored Petri net (TCPN) model. The article presents the viewpoint that energy-aware protocols should be evaluated and reviewed before implementation. In this article, the effect of controlling the node's sleep mode on energy consumption is evaluated, and a TCPN model is presented for this purpose. The presented model is specific to evaluating an energy-aware algorithm based on Bluetooth. In this article, the analysis is done using two complementary approaches, including state space analysis and performance evaluation. By evaluating the state space, the correctness of the algorithm and the matching of the model with the system have been investigated, then the solution of the Petri net model and the energy consumption of the algorithm has been discussed using simulation.
- In [14], the authors presented a method for modeling wireless sensor networks. In this article, to evaluate energy consumption, various criteria such as the working status of the processor, the way nodes are connected, the type of nodes workload, and the general state of the system have been discussed. In this article, the processor and sensor node are modeled using colored Petri nets (CPN). Also, using this model, it has been shown that it is not practical to turn off the processor after calculations quickly or to keep it on all the time. However, it is possible to identify a threshold to turn the processor off or on, which can lead to energy savings. In this article, one of the problems is the relatively long simulation time until reaching the steady state. The presented model is specific to wireless sensor networks. One thing that is not considered is the lack of modeling of the battery module and how it is charged and discharged, which directly affects the system's performance.
- Boutoumi et al. [15] presented a GSPN model to model energy consumption in a wireless sensor network (WSN) with the aim of optimizing energy consumption and reducing network delay. Additionally, the article proposes an algorithm to improve energy consumption in these systems. In the model of this article, it is assumed that the network connectivity is constant and does not change over time. The model considers the on/off policy of the

nodes, where nodes are turned off or on at certain time intervals based on specific conditions. By Considering the combination of energy consumption and network delay, the article finally proposed a method in which each node has a limited buffer. If the number of packets in the buffer is less than n, the node goes to sleep mode, Otherwise, it is on. However, due to the potential increase in network delay, each node may randomly switch its state to active.

- In [16], the authors discuss modeling energy consumption according to network elements. One of this method's problems is not considering parameters such as the channel and environmental conditions.
- In [17], the modeling of energy consumption and packet loss is also discussed, but the article does not consider the channel condition.
- Berrachedi et al. [18] presented a modeling method for wireless sensor networks using DSPN. With this model, the performance and energy consumption of the system has been evaluated. This paper evaluated the presented model on a wireless sensor network with a static structure. Based on this model, it has been concluded that the cluster head's energy consumption strongly depends on the number of nodes connected.
- Lee et al. [19] present a systematic approach to model and evaluate energy consumption in wireless sensor networks. The method is evaluated through various scenarios, demonstrating a close estimation of energy consumption compared to real-world results. However, one limitation of the model is its exclusive consideration of data sending and receiving criteria, neglecting other factors such as processing and sensing. The model is applied to a system based on the ZigBee protocol, and the corresponding results are presented.
- Homssi et al. [20] presented an analytical framework to obtain the lower bound of energy consumption in IoT systems with cellular networks. This paper formulates the effects of network regularity on the energy consumption boundary using three geometric models. Interference measurements and Monte Carlo simulations are presented to validate the proposed analytical framework. The results show that the performance of current network infrastructures is bounded between two extreme geometric models.
- Bouguera et al. [21] presented a model to optimize energy consumption in Lora and Lorawan technologies. To evaluate energy consumption, different scenarios have been tested. In this model, various factors such as spreading factor, rating code, payload size, etc., which influence the system's energy consumption, have been tested. It has been shown that optimizing energy consumption is a trade-off between these criteria.
- Ullah et al. [22] discuss a method for predicting energy consumption in smart buildings using hidden Markov models. The proposed method is tested on multiple intelligent buildings in Seoul and compared with three well-known prediction methods, namely ANN, SVN, and CART, demonstrating improved results. The method relies on collected data from the system for accurate predictions.
- Nourredine et al. [23] presented a model for the energy consumption modeling of autonomous devices using xHPN, which is an extension of the hybrid Petri net. In this method, instead of discretization the consumption and charging of the energy source, a continuous concept has been used, and the results have been presented on devices in Algeria with the ability to charge through solar energy.
- In [24], the authors propose a modeling based on stochastic Petri nets to describe the path taken by packets to reach the base station from each sensor node in wireless sensor networks. This formulation examines the case where the network is composed of structured clusters and each cluster consists of two leaders consisting of a cluster head and a sink that cooperate to route packets from source to destination. This configuration aims to save energy by balancing it across the grid.
- In a different study [25], a model based on GSPNs has been proposed to evaluate the energy usage of WSNs that harvest energy from the environment. This model also includes the representation of sensors and their relationships. The study considers several factors, such as sleeping mechanism, retrial message attempts, battery recharge/discharge, and battery-over breakdowns, and evaluates how these input parameters affect the efficiency of the system. However, similar to other studies, a comprehensive and overarching model encompassing all aspects of energy consumption is not provided.
- Lages et al. [26] developed a method based on CPNs to assess the energy usage of LPWAN IoT systems. They focused on the energy consumption of processors and communication, specifically discussing the physical layer's energy usage. However, one limitation is that they did not consider the energy source and harvesting from the environment.
- Felipe et al. [27] proposed a theoretical approach to predict the minimum and maximum energy consumption in wireless networks using probability distribution functions. They evaluated their method using various scenarios but faced challenges in applying their model to different systems due to its complexity.

- Chenchen et al. [28] introduced a method for evaluating energy consumption in urban rail transit systems (URTs) using Petri nets modeling and evaluation. This method evaluated some measures related to URTs' energy flow direction and consumption process.

The difference between our work and the previous work is in the following aspects:

1. In our work, an energy consumption evaluation model for the Internet of things systems has been presented, in which the main and required concepts of energy consumption modeling have been added to the model.
2. The model includes energy source components and energy-dependent transitions, which has reduced the complexity of energy consumption modeling.
3. The presented work does not depend on a specific case study and the presented model is general.
4. The formal definition and method of solving and evaluating the model are presented.
5. A tool that supports the proposed model is developed.

The classification of related work is given in Table 1.

Table 1. Summary of related works

Reference	Model name	System being modeled	Application
[4]	Analytical	IoT	General
[13]	TCPN	IoT	Energy-aware algorithm
[14]	CPN	WSN	General
[15]	GSPN	WSN	General
[16][17]	PN	IoT, WSN	General
[18]	DSPN	WSN	General
[5]	SPN	IoT	LPWAN-based system
[19]	PN	Hybrid dynamic systems	General (applied to a system based on the ZigBee protocol)
[20]	Analytical	IoT	Cellular networks
[21]	Analytical	IoT	Lora and Lorawan
[22]	Hidden Markov model	Residential buildings	General
[23]	xHPN	IoT	Autonomous devices
[24]	SPN	WSN	Dual Cluster Heads Configuration in Energy-Harvesting WSNs
[25]	GSPN	WSN	Sleeping mechanism, retrial message attempts, battery recharge/discharge, and battery-over breakdowns
[26]	CPN	IoT	CPU and communication energy consumption
[27]	Analytical	WSN	Predicts the maximum and minimum energy consumption of the system
[28]	PN	Urban rail transit system	Compute the measured related to URTs

4. Proposed model

In this section, we propose a model for modeling energy consumption in IoT systems. To support the modeling and evaluation of energy consumption, the SRN model is used as a base model and we have tried to introduce two new elements to be used together with the SRN primitives to be able to construct high-level models of IoT systems. These two elements are called *energy source* and *energy-dependent transition*. By using these high-level elements, energy consumption modeling becomes more manageable, and the modeler does not need to deal with the details of energy consumption modeling. We call the proposed model source-based SRN (SRN) in the following.

To evaluate the SRN, a series of transformation rules are introduced to convert the new elements into the original SRN elements. Then, the obtained SRN model can be analytically solved or simulated. Energy source element

4.1. Energy source element

The energy source element is one of the main concepts needed for energy consumption modeling. Each node in an IoT system can have one or more energy sources, and the activities of that node can depend on one or more energy sources. Any system activity can have positive or negative dependence on resources. If the dependence of the activity is negative, this activity will reduce the energy level of the resource or resources. If the energy level of the source is lower than the dependence value, the activity stops. If the activity dependency is positive, this activity increases the energy level of the resource or resources. If the capacity of the resource is complete, the activity does not affect the resource. Energy sources are divided into two main categories: rechargeable and non-rechargeable energy sources. The following will explain how to support these two types of resources in the proposed model.

In Figure 3, the energy source model is shown. Each source has a characteristic that indicates the maximum capacity of the source, which is displayed as $\mathrm{S}\{name, capacity\}$, where the first parameter specifies the source's name. The second parameter indicates the capacity of the source, which is a natural number greater than zero. In the following, it is explained how resource-dependent transitions connect to these resources. In each resource, the number of tokens inside the resource shows its energy level. According to the resource, each token can be equivalent to X_{mj}, and the dependency of activities is based on tokens. For example, if an activity reduces $4X_{mj}$ per minute, its dependency is equal to 4 tokens, and it reduces four tokens from the number of source tokens per minute.

Figure 3. The energy source element

4.2. Source-dependent transition

Each node in the IoT -based system may perform several activities, each of which may require energy. A computing node's activities include sending, receiving, processing, changing working mode, etc. Any resource-dependent transition can depend on one or more resources. This dependence can be in two different ways, which are:

1. Firing the transition depends on a certain level of resources, and finally, firing the transition reduces the level of resources according to the influence of the transition.
2. Firing the transition increases the energy level of the source. These transitions have a positive effect on the resource. The application of this type of transition is for rechargeable sources.

An example of the source-dependent transition is shown in Figure 4. The transition in Figure 4 has rate 2 and is dependent on source *S1* and source *S2*. This dependence is such that the transition firing decreases one unit of source S1 and three units of source *S2*.

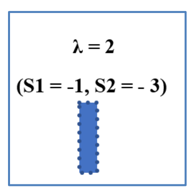

Figure 4. Source-dependent transition

An example of the proposed model is shown in Figure 5. There is an energy source in this model. The node's activity is that it receives packets, performs processing on these packets, and then sends the packets to another destination. The battery in this node is a rechargeable type charged by the charge transition. In general, there are three source-dependent transitions in this model, which are:

- *Process transition:* This transition rate is one and has a dependency of -1 on the source; in the case of a process transition, one token is subtracted from the source level.
- *The send transition:* This transition rate is two and has a dependency of -2 on the source. Two tokens will be subtracted from the source level if the send transition is fired.
- *Charge transition:* This transition rate is one and a +2 dependency on the source; if the charge transition is fired, two tokens are added to the source level.

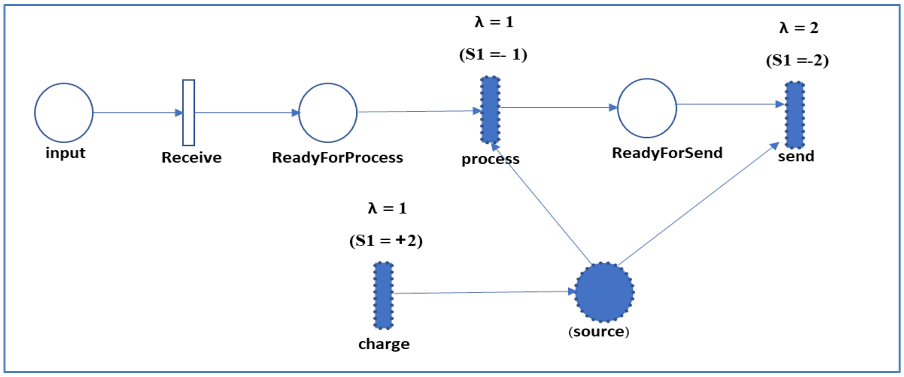

Figure 5. An example of SRN model

4.3. Formal definition

The formal definition of the SRN is given below:

- P is a finite set of places, including Petri net places and energy sources. This set has a minimum of zero and a maximum of $|P|$ sources., which is denoted by P_{sb}.
- $\forall p_s \epsilon P, \exists\ c \in \mathbb{N}$ specifies the capacity of each resource.
- T is a finite set of transitions, including timed, immediate, and source-dependent transitions. From the set of transitions, minimum zero and maximum $|T|$ are dependent on the source, which is represented as T_{sb}. Also, $P \cap T = \emptyset$.
- μ_0 is the initial marking.
- $\forall t\ \epsilon T, \lambda_t : \mathbb{N}^{|P|} \longrightarrow \mathbb{R}^+ \cup \infty$ is the rate of the exponential distribution for the firing time of transition t. If the rate is ∞ in a marking, the transition firing time is zero.
- $\forall t\ \epsilon T, \omega_t : \mathbb{N}^{|P|} \longrightarrow \mathbb{R}^+$ Indicates the weight assigned to immediate transitions.
- $\forall p_{sb}\ \epsilon P, \varphi_t : \mathbb{N}^{|P|} \longrightarrow ((+/-, \mathbb{R}_1), (+/-, \mathbb{R}_2), \ldots, (+/-, \mathbb{R}_{|P_s|}))$ It shows the dependency of source-dependent transitions to each of the sources in the model.
- $\forall t\ \epsilon T, g_t : \mathbb{N}^{|P|} \longrightarrow \{true, false\}$ is the guard function for transition t. If $g_t(\mu) =$ false in μ marking, transition t is inactive.
- $>$ It specifies the priority of the transitions.
- $\forall p\ \epsilon P, \forall t\ \epsilon T, D^-_{p,t} : \mathbb{N}^{|P|} \longrightarrow \mathbb{N}, D^+_{p,t} : \mathbb{N}^{|P|} \longrightarrow \mathbb{N}, D^\circ_{p,t} : \mathbb{N}^{|P|} \longrightarrow \mathbb{N}$ It represents input, output and inhibition arcs. A transition in the marking μ is active if its guard functions have the value true and if $\#(p, \mu)$ is the number of place tokens p in the marking μ $\forall p\ \epsilon P, D^-_{p,t}(\mu) \leq \#(p, \mu) \wedge D^\circ_{p,t}(\mu) > \#(p, \mu) \vee D^\circ_{p,t}(\mu) = 0$ and also if the transition depends on the sources and has dependency $(s_1, s_2, \ldots, s_{|sb|})$ the number of source tokens is greater than its dependency value. $\#(S_1, \mu) \geq \mathrm{s1} \wedge \ldots \wedge \#(S_{|sb|}, \mu) \geq s_{|sb|}$.

4.4. Transformation rules

This section explains the rules for converting the energy source module and source-dependent transition to the SRN model.

4.4.1. Conversion of the energy source element into the SRN *submodel*

To convert the energy source into the SRN model, the following Algorithm 1 can be used.

ALGORITHM 1: CONVERT ENERGY SOURCE INTO SRN SUBMODEL

Input: *energy source model*
Output: SRN *submodel*
Energy source replaced with the SRN *place*
While *(each transition **t** has output edge to source)* ***do***
    *Add a guard function **gf_t** for transition **t***
    gf_f ← p(source) + ***dependence_value*** *of the **t** on the* **source** < ***source_capacity***
end

4.4.2. Conversion of the source-dependent transition into the SRN *submodel*

To handle the conversion of source-dependent transitions, two distinct scenarios can arise based on whether the dependence is positive or negative. In such cases, Algorithm 2 can be employed as follows:

ALGORITHM 2: CONVERTING THE SOURCE-DEPENDENT TRANSITION INTO THE SRN SUBMODEL

Input: *source-dependent transition*
Output: SRN *submodel*
if *(transition does not have a positive dependence on a source)* ***then***
    Add an edge from the source to the transition
    Edge weight ← *dependency value of transition to source*
else
    If *(The transition has no output edge)* ***then***
        Add an edge from the transition to the source
        Edge weight ← *dependency value of transition to source*
        Add guard function for transition
        Guard function ← (edge weight + ***p(source)*** < ***source_capacity)***
    else
        A new place will be added
        An arc is added from the transition to this new place
        Two immediate transitions are added. **source_full** and **source_not_full**
        priority of immediate transitions← *highest possible value*
        An arc is drawn from the newly added place to both immediate transitions
        An arc is added from the transition to this new place
        An arc is added from ***source_not_full*** *to* ***source***
        Add ***source_full_guard*** *to* ***source_full*** *transition*
        Add ***source_not_full_guard*** *to* ***source_not_full*** *transition*
        source_full_guard ← *the* ***source*** *has no capacity*
        source_not_full_guard ← *the* ***source*** *has capacity*
    end
end

A source-dependent transition convert instance is shown in Figure 6, which has an outgoing edge to another place.

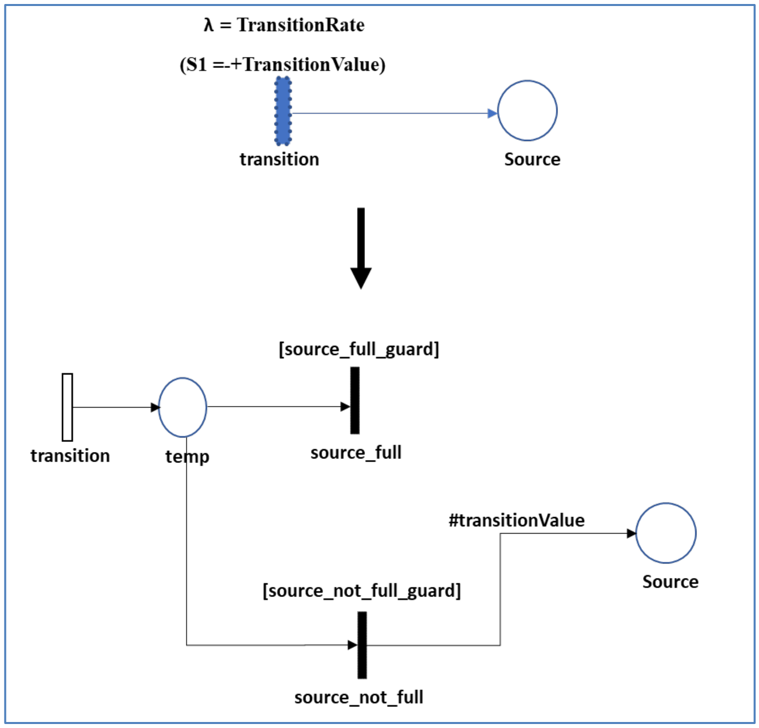

Figure 6. Example of source-dependent transition convert

4.4.3. Conversion of the SRN *model into the* SRN *model*

To convert the SRN model into the SRN model, the following Algorithm 3 can be used on the entire model:

ALGORITHM 3: CONVERTING THE SRN INTO THE SRN MODEL

Input: SRN *model*
Output: SRN *model*
Initialization of variables: SRN_Model = new SRN()
foreach *(source **s** in* SRN *model)* ***do***
  converted_source *←ALGORITM1(**s**)*
  add ***converted_source*** *to* ***SRN_Model***
end
foreach *(source-dependent transition **st** in* SRN *model)* ***do***
  converted_transition *←ALGORITM2(**st**)*
  Change ***converted_transition*** *to an* SRN *transition*
  add ***converted_transition*** *to* ***SRN_Model***
end
foreach *(place **p** in* SRN *model)* ***do***
  *add **p** to* ***SRN_Model***
end
foreach *(transition **t** in* SRN *model)* ***do***
  *add **t** to* ***SRN_Model***
end
foreach *(edge **e** in* SRN *model)* ***do***
  *add **e** to* ***SRN_Model***
end
return ***SRN_Model***

4.5. Assumptions, Constraints and Scalability

Two key assumptions regarding the presented model in this article are taken into consideration:

- **First assumption:** The presented model considers the energy source and its behavior discretely. Resource capacity is considered a set of tokens, and each token is equivalent to x Joules. In the modeled system, each token can be equivalent to the smallest amount of energy consumption or production, and the rest of the dependences on the source can be approximated based on it. The reason for using this method is to simplify the presented model and prevent the model from becoming complicated by introducing continuous processes and related solution methods. Also, discretizing and approximating continuous processes is a standard method for modeling continuous processes.

- **Second assumption:** In the presented model, the random processes of the system are considered processes that follow the exponential distribution. Usually, the processes of computer systems follow the exponential distribution. Therefore, this model supports most of the study cases in the field of the Internet of Things. However, adding support for other random distributions to the proposed model is possible, which is presented as future work in this area.

The following explains the constraints and how to apply the boundary conditions to the model.

One of the constraints of Petri net models is the issue of state space explosion. If the details of a system increase and the corresponding model becomes large, the number of system states may increase exponentially, and this causes the state space to explode. Considering that the presented model in this article is also based on Petri nets, this constraint also exists in this model. To avoid this problem, it is necessary to refrain from dealing with the details that have an insignificant effect on the system's behavior, and the system's main processes should be considered and modeled. Therefore, in system modeling using Petri models, complex assumptions of system behavior should be ignored to avoid the problem of state space explosion in the model's analytical solution and the generation of the system state space. To correctly model the behavior of any system, there is a set of boundary conditions that control the system's behavior. In the presented model, these boundary conditions are applied using the guard functions of the transitions in the model. Each transition has a guard function, which controls the active/inactive status of the transition. The desired transition is active if the output of the guard function is true. Otherwise, the transition is inactive. (For example, the transition that leads to the charging of the energy source is active if the energy source has capacity. This condition is included in the model using a guard function.) The definition of these guard functions is given in Section 4.3. The presented model in this article can be applied to all Internet of Things systems, provided that the studied system covers the assumptions and constraints of the model. The desired system must have the following features:

- The processes in the system follow the exponential distribution. Therefore, the random processes of the system can be approximated by an exponential distribution. If the desired system has this feature, it is possible to apply the model.

- It is possible to ignore some details of the system to simplify the model and prevent the explosion of the state space. If the system is vast and its state space becomes enormous, it will be impossible to identify and solve the state space analytically. In this case, the system should be divided into smaller parts and modeled separately.

- The discretization of the energy source and its behavior should not severely impact the system.

Its is worth-mentioning that the proposed method is generally scalable. The scalability of the model can be investigated by two cases:

- **If the system under study increases in terms of the number of nodes:** In this case, the values of the model's parameters must be changed, and the model will cover this type of scalability. For example, a parameter in a model can indicate the number of nodes connected to a sink node; by changing this parameter, the scale of the system in terms of the number of nodes in the system can be changed.

- **If the working conditions of the system under study become wider:** In this case, the model's structure should be changed. These details must be considered in the model using new places and transitions.

However, one limitation in Petri net-based modeling is the issue of state space explosion. Oversizing the model can lead to an increase in the state space, and, as a result, the model cannot be evaluated. The proposed method has this limitation since it is also based on the Petri nets.

4.6. A tool developed to support the model

To support the presented model in this article, a tool has been developed that can be used to model and evaluate Internet of Things systems using the SB-SRN model. This tool is named PNDE[†]. An image of the developed tool environment is shown in Figure 7. The implementation and operation of this tool are as follows:

- At first, the desired system model is designed using the user interface, and the relationships between places, transitions, and energy sources are determined.
- At this stage, definitions related to guard functions should be added to the model.
- Then, the evaluation method and related parameters should be set.
- Then, the developed tool automatically converts the SB-SRN model into an SRN model using the conversion rules presented in Section 4.4.
- SB-SRN model can be used in the SPNP tool after converting to SRN. The transformed model is automatically run in SPNP, and the evaluation results are shown to the user.

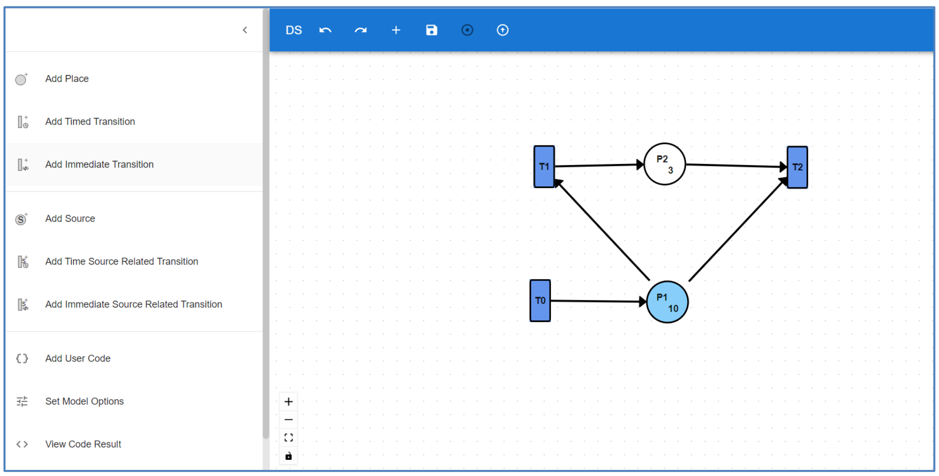

Figure 7. The *PNDE* tool environment

5. Case study

In this section, we present a case study to investigate the usefulness of the proposed model.

5.1. The problem description

The case study is selected from [29][30] articles. Each computing node in the system consists of different parts:

- **Receiving and sending section:** The node sends the information received from other nodes or the information sensed by the node to the neighboring node or the central node.
- **Energy management:** including node battery and energy consumption management policies
- **Sensor part:** It consists of various sensors that can sense events.
- **Processing Unit:** The task of this unit is to process data and information before sending or after receiving packets.

Each node receives information from sensors and other nodes and then processes this information. After processing, it sends this information to other nodes through the channel. The node needs the energy to perform some of its activities, such as listening to the channel, receiving packets, sending packets, etc., for which a battery is considered in the node. The built-in battery can be charged by receiving energy through sunlight. Also, for better energy

[†] https://gitlab.com/groups/pnde

management, a mechanism has been considered to make the node go to standby mode and reduce the energy consumption of the node. In the presented model in [30] article, in the section on receiving energy from the environment, different modes are considered, which are:

- **The level of solar radiation based on the season:** The level of solar radiation is considered in different seasons. In this way, this level is at the highest possible level in the summer and at the lowest possible level in the winter, and an average level is considered in the spring and autumn seasons.
- **No energy harvesting at night:** energy harvesting is not done at night due to the absence of the sun, and energy harvesting is only during the day.

Also, there can be three different strategies regarding how to change the mode of the node and go to standby mode:

- **Simple mode:** In simple mode, the node goes to sleep when the battery energy is less than a certain level and wakes up again.
- **Dual standby mode:** in dual mode, the rate is different for changing the sleep and wake modes during the day and night. In this way, the node sleeps less during the day and wakes up faster, and sleeps more at night and wakes up later.
- **Buffer threshold:** When the number of messages in the input buffer is less than a threshold, the node goes to sleep

5.2. The problem modeling

In this section, the modeling of the case study is discussed using the proposed model in the article. Each part of the model is explained separately, and finally, the model of the whole system is presented

5.2.1. Sending and receiving section

The node has a section for receiving and sending data; the modeling of this section is shown in Figure 8. The node constantly listens to the channel to receive new packets if it is working and not in standby mode (the model will have a place called *standby* that indicates whether the node is in working mode or not, which will be shown later). Received packets are stored in a buffer, and after initial processing, the packet will be sent to other nodes.

The model of this part of the system has 3 energy-dependent transitions, which are:

- ***Working transition*:** This transition reduces the energy source level by the value *s1* at a specified rate when the node is not in *standby* mode
- ***Listening transition*:** This transition listens to the channel. Also, this transition depends on the energy source with the value of *s2*. This transition is active if the node is not in standby mode.
- ***Transmit transition*:** This transition sends processed packets to other nodes. The node needs the energy to send data; this transition is source-dependent, and its dependence rate equals *s3*.

Table 2 shows the variables related to this part of the model. Also, in Table 3, the definitions of the guard functions related to the transitions of the model are given.

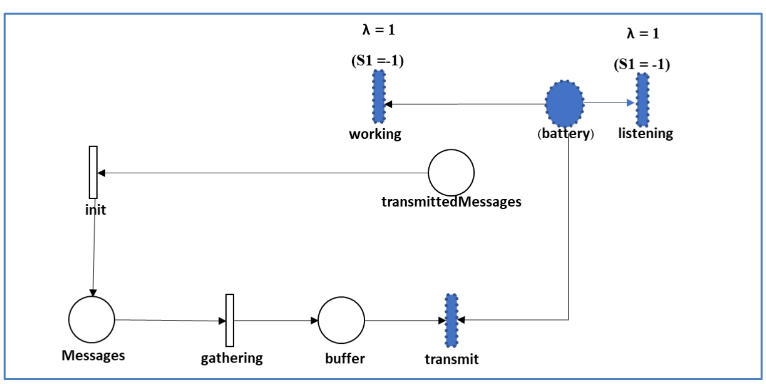

Figure 8. Sending and receiving section

Table 2. Variables of sending and receiving section

Variable name	values
WorkingRate	0.01
TransmitRate	1/60m
ListeningRate	1/60m
InitRate	1/24h
GatheringRate	1
WorkingDependency	1
TransmitDependency	1
ListeningDependency	1

Table 3. Guardian functions of sending and receiving section

Guard name	Definition
InitGuard	mark("StandBy") == 0
GatheringGuard	mark("CH_Input") == 0
TransmitGuard	mark("StandBy") == 0
WorkingGuard	mark("StandBy") == 0
ListeningGuard	mark("StandBy") == 0

5.2.2. Energy harvesting section

In this section, according to the method presented in [30] article, three different rates have been considered for receiving energy from the environment, and these rates have been used in different seasons. In the summer season, due to more sunlight, this rate is at its highest level; in the spring and autumn seasons, there is an intermediate rate, and also, in the winter season, this rate is at its lowest possible state. Also, for modeling different seasons in the model, there is a section that shows the current season, which is located on the left side of the model in Figure 9.

This model has three energy-dependent transitions, which are:

- ***H_Harvest transition:*** This transition leads to an increase in the energy level of the source with the dependence of $x1$ and rate of $y1$ if the energy source has capacity.
- ***M_Harvest transition:*** This transition leads to an increase in the energy level of the source with the dependence of $x2$ and rate of $y2$ if the energy source has capacity.
- ***L_Harvest transition:*** This transition leads to an increase in the energy level of the source with the dependence of $x3$ and rate of $y3$ if the energy source has capacity.

Table 4 and Table 5 show the variables and guard functions definition.

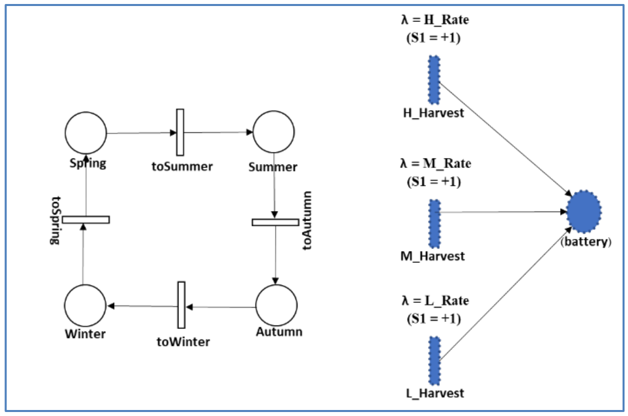

Figure 9. Energy harvesting section

Table 4. Variables of the energy harvesting section

Guard name	Value
Harvest_Dependencyc	1
ChangeSessionRate	1/(3*24h)
H_Rate	0.2
M_Rate	0.05
L_Rate	0.033

Table 5. Guard functions of the energy harvesting section

Variable name	Value
H_Harvest	if (mark("*Battery*") < BatteryCapacity && mark("*Night*") == 0 && mark("*Summer*") == 1)
M_Harvest	if (mark("*Battery*") < BatteryCapacity && mark("*Night*") == 0 && (mark("*Spring*") == 1 || mark("*Autumn*") == 1))
L_Harvest	if (mark("*Battery*") < BatteryCapacity && mark("*Night*") == 0 && mark("*Winter*") == 1)

5.2.3. Day and night section

This section models how the node wakes up and goes to sleep during the day and night. Node falls asleep less during the day and wakes up earlier, and node falls asleep more often at night and wakes up later. The model related to this section is shown in Figure 10.

If the energy level of the node is less than a specific value, the node immediately goes to sleep; this happens through the immediate sleep transition.

Table 6 and Table 7 show the variables and guard functions definition.

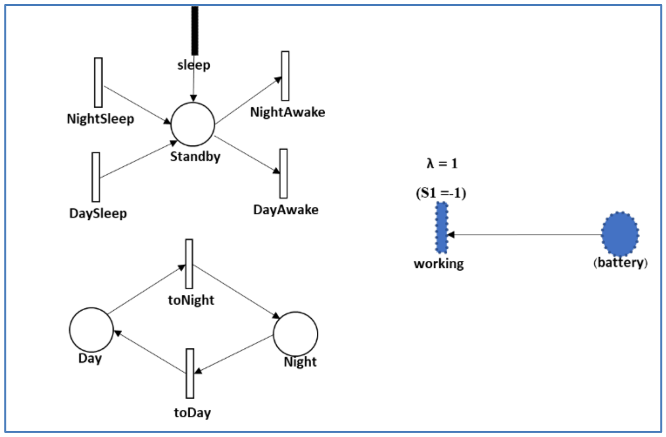

Figure 10. Day and night section

Table 6. Variables of day and night section

Variable name	Value
NightSleepRate	1/2m
NighAwakeRate	1/3m
DaySleepRate	1/3m
DayawakeRate	1/2m
ToDay	1/12h
toNight	1/12h

Table 7. Guard functions of day and night section

Guard name	Definition
Sleep_*guard*	(mark("*Battery*") < *threshold*)

5.2.4. System model

In the previous sections, the model of each part of the system was described separately. Finally, the model of a system node using the SRN model is shown in Figure 11. The connection between the energy-dependent transitions and the energy source is not drawn for the model's beauty.

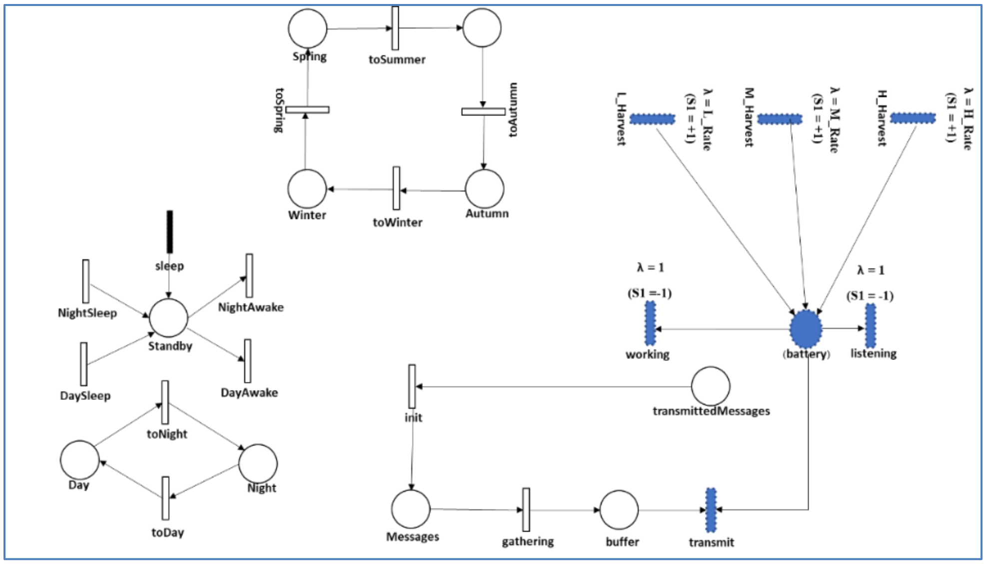

Figure 11. The SRN model of the system

5.2.5. Model execution modes

In this section, the implementation of the case study is discussed. Different scenarios have been implemented on the model, and the change of different system parameters has been checked on the system criteria. Also, the model has been implemented in different modes, which are explained in Table 8. Also, the parameters whose effect on the system criteria has been investigated are:

- Different energy harvesting rates,
- Different sending and receiving rates,
- The number of daily messages, and
- Different waking and sleeping times during the day and night.

The different modes of system configuration are as follows:

- ***SL***: This configuration is the most straightforward. In this case, the difference in energy harvest in different seasons is not considered, and the energy harvest rate is considered an average rate in all seasons. Also, there is no difference in the time and rate of sleep and awakening of nodes during the day and night.
- ***DS***: Unlike the *SL* mode, a double sleep model is considered in this mode. That is, the duration of sleep and wakefulness of the node are different during the day and night. However, the energy harvesting rate is the same as the *SL* model in different seasons.
- ***SS***: This mode is the exact opposite of the *DS* mode. In this case, the energy harvesting rate is considered different in different seasons, but the sleeping and waking time of the node is considered simple.
- ***SS_DS***: This mode is a combination of *SS* mode and *DS* mode. In this configuration, the energy harvesting rate is different in different seasons, and the time and rate of node sleep and wake up are different day and night and are dual.
- ***LB***: This mode is an extended configuration of the *SS_DS* configuration. In *LB* mode, if the packets in the buffer are less than a threshold, the node goes to sleep and wakes up again after increasing the incoming packets in the buffer.

Table 8. Model configurations

name	definition
SL	Seasons are not considered, and the sleep mode is simple
DS	Seasons are considered, and the sleep mode is dual
SS	Seasons are considered, and the sleep mode is simple
SS_DS	Seasons are considered, and the sleep mode is dual
LB	SS_DS + considered buffer threshold for sleep mode

5.3. Model evaluation

The following scenarios have been designed and implemented on the configurations illustrated in the previous section. In these scenarios, various system parameters, such as the energy harvesting rate, the number of daily messages, the sending and receiving rate, and so on, have been changed, and their effects on the system performance has been investigated. To evaluate the models, the steady-state solution has been used. The developed PNDE tool has been used to design and implement the model. The evaluated scenarios are as follows:

- **The effects of different energy harvesting rates on energy source level and the latency**: In this scenario, we changed the energy harvesting rate, and the impact of this rate change on the energy level of the energy source and the message delay in the system is investigated, and different configurations are compared.
- **The effects of the number of daily messages on the energy resource level and the latency**: In this scenario, the number of daily messages of the node is the variable, and the effects of the number of daily messages on the energy level of the resource and the delay of the messages is investigated. In this experiment, the number of daily messages varies from 10 messages per day to 70 messages per day.
- **The effects of different sending rates on the latency**: In this scenario, the variable is the different sending rates, which is changed from 0.2 to 1.0, and the impacts of this sending rates on the message delay is evaluated.
- **The effects of sleep time on the delay**: In this scenario, the effect of node sleep time on message delay is studied. This scenario is conducted on three configurations with a sleep mode. The goal of this scenario was to find the best sleep duration for the nodes without significantly increasing the latency
- **The duration of the node's sleeping versus the node's awakening**: In this scenario, the node's activity and sleep in different configurations is compared considering the daily messages equal to 20.

The models presented in this articled are based on the proposed method (SB-SRN). These models are firstly converted into the SRN models using the proposed transformation rules, and secondly are solved using the SPNP tool. The model solution by the SPNP tool. The SPNP tool itself converts the SRN model into Markov reward models (MRMs). Then, it uses the steady state solution methods of MRMs. Finally, the measured are computed.

In our experiments, we have used a system with the following specifications: Intel Core i7 CPU, 12 GB RAM, and Microsoft Windows 11 64 bits operating system.

5.3.1. The effect of different energy harvesting rates on energy source level and latency

Figure 12 shows the node's battery energy level according to different energy harvesting rates. The average energy harvest rate changed from 0 to 1. As evident in the figure, when the average energy harvesting exceeds 0.10, the energy source level reaches 100% in *SIMPLE* and *DS* modes. In these two cases, due to the inaccuracy of the model and the same energy harvesting in all four seasons, the battery energy level has reached 100%. The model is more accurate in *SS* mode due to considering different energy harvesting in different seasons, and the battery energy level in a steady state reaches 60%. In the *SS_DS* mode, due to the dual sleep strategy, the node's energy consumption is reduced, and in the steady state, this energy level reaches 68%. In *LB* mode, because the number of buffer packets is less than the set threshold, The node goes to sleep, so the node's energy consumption decreases due to the increase in the sleep time of the node, and the energy level of the source increases to about 74%.

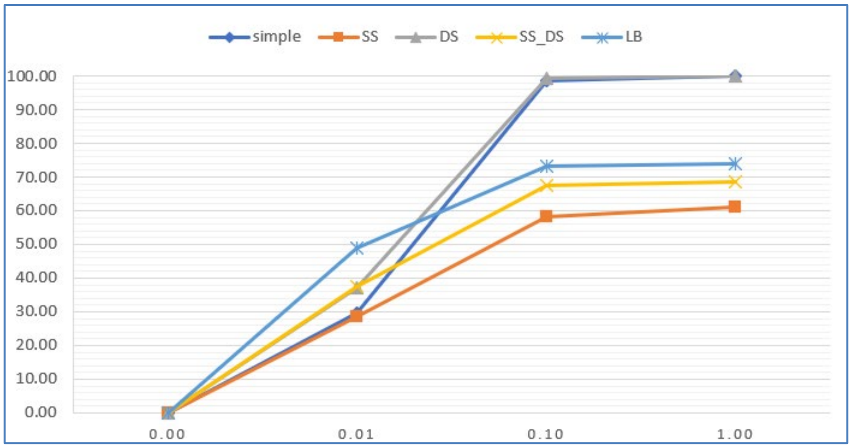

Figure 12. Effect of different energy harvesting rates on energy source level

Figure 13 shows the effect of different energy harvesting rates on delay. As it is evident in the diagram, in simple and *SS* mode, due to the absence of sleep mode and the presence of energy harvesting in all day and night intervals, increasing the energy harvesting rate does not have a significant effect on the delay, and the reason is that the node is constantly working. Furthermore, it has the minimum required energy level. In *SS_DS* and *DS* modes, when the rate of receiving energy is low because the energy source level reaches the node's sleep threshold, the node goes to sleep mode, which leads to an increase in delay. However, with the increase in the rate of energy withdrawal, this delay decreases. Will find. In *LB* mode, the delay is higher than in other modes because the node's standby time is longer than in other modes. In the case where the average energy harvesting rate is 1, the delay in *LB* mode is about 0.6 *ms* longer than in *DS* and *SS_DS* modes.

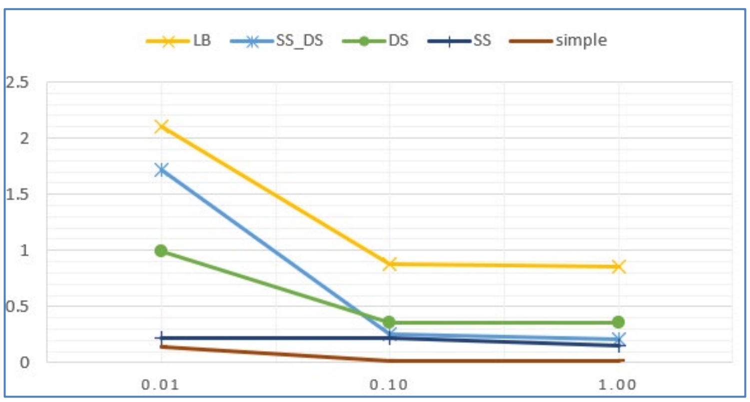

Figure 13. Effect of different energy harvesting rates on latency

5.3.2. The effect of the number of daily messages on the energy resource level and latency

Figure 14 shows the effect of the number of daily messages on the battery energy level. As shown in the figure, the battery energy level stays mostly the same with the number of messages, and the increase in the number of messages leads to a slight decrease in the energy level. Figure 15 shows the ratio of delay to the number of daily messages. In all models except *LB*, the message delay increases with the number of messages. This increase is more in *DS* and *SS_DS* modes. However, in *LB* mode, because the delay of messages is more affected by the number of messages in the buffer and the node sleeps according to the number of messages in the buffer, the increase in the number of messages does not affect the delay of messages. So that when the number of messages exceeds 30 messages. Almost the delay of the messages increases at a slower rate; the reason is that when the number of messages exceeds 30, the node always works at its maximum capacity.

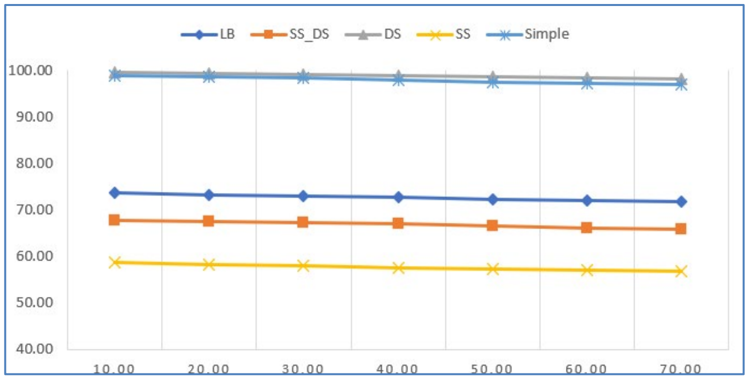

Figure 14. Effect of the number of daily messages on the energy resource level

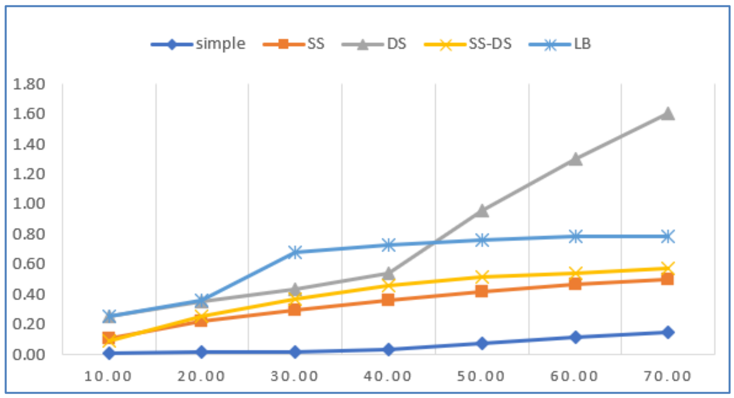

Figure 15. Effect of the number of daily messages on the latency

5.3.3. *The effect of different sending rates on latency*

Figure 16 shows the effect of different sending rates on message delay. In this scenario, the message-sending rate has increased from 0.2 to 1 with a step length of 0.1. As the figure shows, increasing the rate of receiving messages does not have a significant effect on the delay of messages, and the delay of messages is constant in almost all cases, and increasing the sending rate will lead to a 5% reduction in delay. Of course, this effect is more significant in *DS* mode. In most cases, a sending rate of 0.3 is a good option. Moreover, increasing the sending rate by more than 0.3 only increases the cost and does not affect the message delay much.

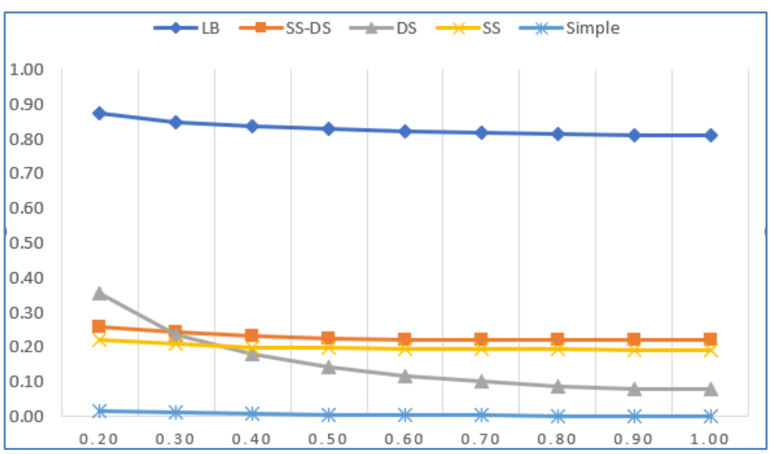

Figure 16. Effect of different sending rates on latency

5.3.4. *The effect of sleep time on latency*

The effect of sleep time on message latency in three configurations is investigated here. As it is evident in the figure, in *LB* mode, increasing the duration of sleep does not have a significant effect on the delay; the reason is that in this configuration, the main delay of messages is due to the presence of the node sleep rule in case of low workload. However, in the *DS* configuration, the message delay increases as the sleep time increases, so this increase increases as the sleep time exceed 5 minutes. Therefore, according to the diagram in Figure 17, the sleep time of the node can be increased to about 4 minutes so that it does not have a significant impact on the system performance.

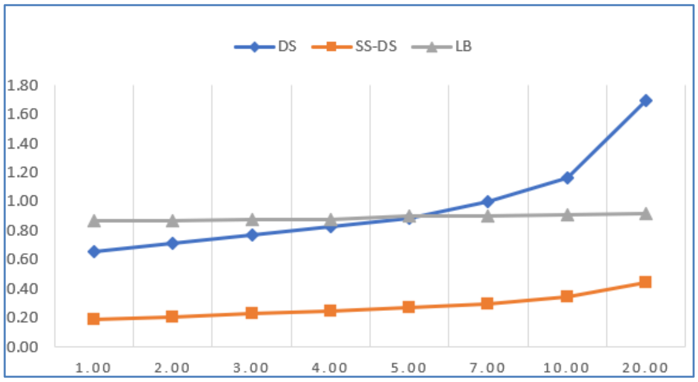

Figure 17. Effect of sleep time on delay

5.3.5. *The duration of the node's sleep versus the node's awakening*

Figure 18 shows the percentage of node sleep and wakefulness in 4 different system configurations. In all these configurations, the number of daily messages is 20. As is evident in the diagram and as expected, the lowest activity time is related to the *LB* and *DS_SS* configurations, which is about 11.99% for the *LB* configuration and 22.39% for the *DS_SS* configuration. As it is known, the reason for the lower activity time in the *LB* configuration is to be in sleep mode if the number of packets in the buffer is low.

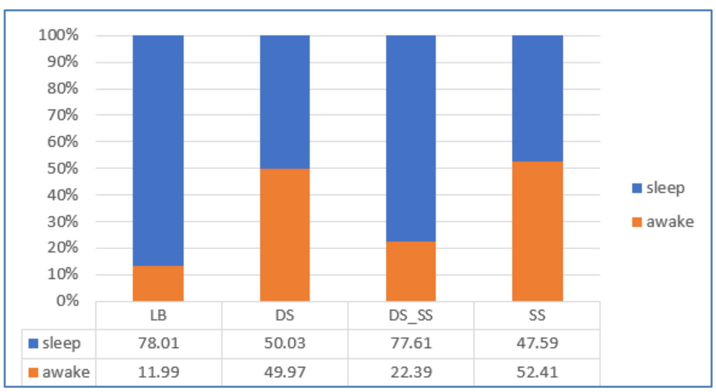

Figure 18. Sleep time *vs.* activity time

6. Comparison

This section presents a comparative analysis of the presented model in the article with other related works in the field. The focus lies on comparing the features and capabilities of the proposed model with existing models, as well as validating the obtained results by contrasting them with findings from related benchmark methods.

6.1. Comparing the features and capabilities

In this article, a model is presented that provides the basic concepts required for modeling energy consumption in a system. In fact, in this model, the energy source, transitions related to it, and their related concepts are intrinsically defined in the model, and the modeler works directly with these concepts to model an Internet of Things system. The desired model is general, and if the system is compatible with the model's assumptions, it can be modeled and evaluated with this model.

One of the works presenting a general model is the article [4], which does not depend on a specific scenario. The model of the article [4] is entirely formal and analytical. One of the problems of this model is the complexity of the analytical model to be used in modeling. Also, using analytical models requires high-level mathematical knowledge, but the analytical model is precious for evaluating other models and can confirm the correctness of other models.
In the articles [14, 15, 27], modeling of energy consumption using Petri nets has been discussed. The models in these articles are specially designed to investigate a specific scenario. In fact, in these articles, a specific scenario has been considered. Scenario modeling has been done using one of the Petri model distributions, and Its results have been presented. These articles have yet to present a comprehensive model for modeling energy consumption in the Internet of Things systems. This comprehensiveness is covered in our work by adding components required for energy consumption modeling. Also, in other works, a series of particular criteria have been considered according to the type of case study. Table 9 compares these related works in this field with the presented work in this article. Also, in this article, a tool has been developed to support the presented model, the details of which are given in section 4.6.

Table 9. Comparing the features and capabilities

Article	Criteria	General	Model	Modeling basic concepts
Current article	Influence of various factors on energy harvesting Latency Receive and send rates Processor working states The possibility of measuring other criteria due to the generality of the model	✓	SRN	✓
[4]	Receive and send rates Processor working states Node and channel failure	✓	Analytical	×
[27]	Influence of various factors on energy harvesting	×	GSPN	×
[15]	Buffer size Latency	×	GSPN	×
[14]	The working policy of the processor according to the workload	×	PN	×

6.2. Comparing the results obtained from the model

As mentioned earlier, the study case presented in Article 4 has been used in this article. Article [4] discusses evaluating the case study using the GSPN model and actual world results. Also, the *LB* configuration is presented in our paper, and we compare its results with other configurations in the following:

- The results of article [30] have shown that the energy source level in the *SS_DS* configuration in a stable state is about 70%, which is consistent with the results of the model presented in this article and confirms the correctness of the modeling using our method. Also, the energy level of the *LB* configuration tested in this paper is about 4% higher, which indicates that adding the buffer level threshold mode can lead to lower energy consumption.
- It is shown in Figure 16 that the packet delay is about 0.2 seconds according to the sampling rate in ss, simple configurations, which is consistent with the results presented in Article [30]. However, this number is higher in the LB configuration due to the number of sleep times, which is about 0.9 seconds.
- Also, by taking into account the number of different daily messages, it is shown in Figure 14 that the *LB* configuration has about a 6% higher energy level than the *SS_DS* configuration.

The results obtained using the modeling method presented in this article compared to article [30] show that the results are correct and the system is correctly modeled. Also, the *LB* configuration presented in this article in the steady state has lower energy consumption than other configurations, but the packet delay in this configuration is about 0.6 seconds longer; this shows that using this configuration in applications where packet delay is not significant can help a lot to improve energy consumption.

In this section, the model's results presented in our article are examined with the model of [4], which is an analytical model. For this purpose, the existence of sleep mode in the system was evaluated according to its workload, its effect on the energy source level was investigated, and its results were analyzed with the results of Article 4. In the experiments, it was shown that if the system load is low, the presence of sleep mode can cause energy consumption, and the reason for this is the use of energy to turn the node on and off. This result has been confirmed using the analytical model of the article [4], which shows that the model presented in this article is correct in terms of performance. The results of these tests are given in the diagram of Figure 19.

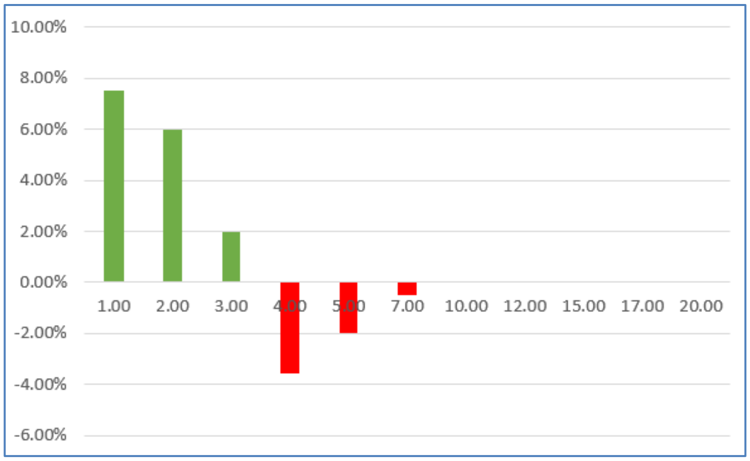

Figure 19. The difference in energy source level with and without sleep mode considering different workloads

7. Conclusions

This article presented a modeling method called SRN to model and evaluate energy consumption in IoT -based systems. The presented model contains the concepts required for modeling energy consumption, including energy source and source-dependent transition. To evaluate and solve the model, transformation rules were presented that transform the SRN model into SRN*s*.

The presented model was evaluated in a case study, and the desired system was evaluated with five different configurations based on different parameters. The obtained results showed that the *LB* configuration has the lowest energy consumption, but this configuration has a longer delay in sending the message to the destination. *SS_DS* configuration is placed after the *LB* configuration regarding energy consumption, with a lower sending delay. In the *LB* configuration at steady state, the energy level is about 74%, while in *SS_DS*, it is about 70%. However, on the other hand, the message-sending delay in *LB* configuration is about 0.8 milliseconds, while this number is about 0.4 for *SS_DS*.

Unlike previous works, past studies did not specifically present a modeling method for energy consumption assessment. However, the presented model in this article incorporates the necessary concepts for evaluating energy usage. In this study, the focus was primarily on the case study itself, rather than delving into the details of how to construct the underlying model for these concepts.

This article presented a comprehensive model for modeling and evaluating the energy consumption of IoT systems with the support of related concepts. For the case study presented in this paper, we transformed the SRN models into the original SRN models manually. For the automatic transformation of SRN models, we are developing a software tool. This tool will facilitate model construction, automatic transformation of the SRN models into SRN*s* and then solving the transformed model using the existing SPNP tool.

References

[1] M. Raval, S. Bhardwaj, A. Aravelli, J. Dofe, and H. Gohel, “Smart energy optimization for massive IoT using artificial intelligence,” Internet of Things, vol. 13, Mar. 2021.

[2] K. Georgiou, S. Xavier-de-Souza and K. Eder, "The IoT Energy Challenge: A Software Perspective," in IEEE Embedded Systems Letters, vol. 10, no. 3, pp. 53-56, Sept. 2018.

[3] F. Javed, M. K. Afzal, M. Sharif and B. -S. Kim, "Internet of Things (IoT) Operating Systems Support, Networking Technologies, Applications, and Challenges: A Comparative Review," in IEEE Communications Surveys & Tutorials, vol. 20, no. 3, pp. 2062-2100, 2018.

[4] E. Ever, P. Shah, L. Mostarda, F. Omondi and O. Gemikonakli, "On the performance availability and energy consumption modelling of clustered IoT systems", Computing, vol. 101, no. 12, pp. 1935-1970, 2019.

[5] D. Lages, E. Borba, J. Araujo, E. Tavares and E. Sousa, "Energy Consumption Evaluation of LPWAN: A Stochastic Modeling Approach for IoT Systems," 2021 IEEE International Systems Conference (SysCon), pp. 1-8, 2021.

[6] A. Lekidis and P. Katsaros, “Model-based design of energy-efficient applications for IoT systems,” in: S. Bliudze and S. Bensalem (eds.), Methods and Tools for Rigorous System Design (MeTRiD 2018) EPTCS 272, pp. 24–38, 2018.

[7] S. Li, L. Da Xu, and S. Zhao, “The internet of things: a survey,” *Inf. Syst. Front.*, vol. 17, no. 2, pp. 243–259, 2015, doi: 10.1007/s10796-014-9492-7.

[8] B. Martinez, M. Montón, I. Vilajosana and J. D. Prades, "The Power of Models: Modeling Power Consumption for IoT Devices," in IEEE Sensors Journal, vol. 15, no. 10, pp. 5777-5789, Oct. 2015, doi: 10.1109/JSEN.2015.2445094.

[9] T. Murata, "Petri nets: Properties, analysis and applications," in Proceedings of the IEEE, vol. 77, no. 4, pp. 541-580, April 1989.

[10] M. Ajmone, B. Gianfranco, C. Gianni, D. Susanna, and F. Giuliana, Modeling With Generalised Stochastic Petri Nets. 1998.

[11] J. K. Muppalaa and K. S. Trivedia, “Composite Performance and Availability Analysis Using a Hierarchy of Stochastic Reward Nets,”, pp. 335–349, 2010.

[12] C. Hirel, B. Tuffin, and K. S. Trivedi, “SPNP: Stochastic petri nets. Version 6. 0,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 1786, pp. 354–357, 2000.

[13] M. C. Ruiz, C. Garrido-Hidalgo, D. P. Gruska, T. Olivares, D. Hortelano and L. Roda-Sanchez, "Modeling and Evaluation of a Power-Aware Algorithm for IoT Bluetooth Low Energy Devices," 2019 IEEE International Conference on Smart Internet of Things (SmartIoT), pp. 28-35, 2019.

[14] A. Shareef and Y. Zhu, "Energy Modeling of Wireless Sensor Nodes Based on Petri Nets," 2010 39th International Conference on Parallel Processing, pp. 101-110, 2010.

[15] B. Boutoumi and N. Gharbi, "An energy saving and latency delay efficiency scheme for wireless sensor networks based on GSPNs," 2017 4th International Conference on Control, Decision and Information Technologies (CoDIT), pp. 0645-0650, 2017.

[16] B. Martinez, M. Montón, I. Vilajosana and J. D. Prades, "The Power of Models: Modeling Power Consumption for IoT Devices," in IEEE Sensors Journal, vol. 15, no. 10, pp. 5777-5789, Oct. 2015.

[17] Ali Shareef, Yifeng Zhu, "Effective Stochastic Modeling of Energy-Constrained Wireless Sensor Networks", Journal of Computer Networks and Communications, vol. 2012, pp 67-71, 2012.

[18] A. Berrachedi and M. Boukala-Ioualalen, "Evaluation of the Energy Consumption and the Packet Loss in WSNs Using Deterministic Stochastic Petri Nets," 2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA), pp. 772-777, 2016.

[19] J. Lee, M. Zhou, and P. Hsu, “Multiparadigm Modeling for Hybrid Dynamic Systems Using a Petri Net Framework,” *IEEE Trans. Syst.*, vol. 38, no. 2, pp. 493–498, 2008.

[20] B. Al Homssi, A. Al-Hourani, S. Chandrasekharan, K. M. Gomez and S. Kandeepan, "On the Bound of Energy Consumption in Cellular IoT Networks," in IEEE Transactions on Green Communications and Networking, vol. 4, no. 2, pp. 355-364, June 2020, doi: 10.1109/TGCN.2019.2960061.

[21] Bouguera T, Diouris J-F, Chaillout J-J, Jaouadi R, Andrieux G. Energy Consumption Model for Sensor Nodes Based on LoRa and LoRaWAN. Sensors. 2018; 18(7):2104. https://doi.org/10.3390/s18072104

[22] Ullah I, Ahmad R, Kim D. A Prediction Mechanism of Energy Consumption in Residential Buildings Using Hidden Markov Model. Energies. 2018; 11(2):358. https://doi.org/10.3390/en11020358

[23] O. Nourredine, D. Abderrezak, A. Karima and H. Mira, "A Fluid Approach to Model and Assess the Energy Level of Autonomous devices in IoT with Solar Energy Harvesting Capability," 2022 5th International Symposium on Informatics and its Applications (ISIA), M'sila, Algeria, 2022, pp. 1-6, doi: 10.1109/ISIA55826.2022.9993564.

[24] O. Nourredine and B. Menouar, "A New Stochastic Petri Nets Modeling for Dual Cluster Heads Configuration in Energy-Harvesting WSNs," 2020 2nd International Workshop on Human-Centric Smart Environments for Health and Well-being (IHSH), Boumerdes, Algeria, 2021, pp. 142-147, doi: 10.1109/IHSH51661.2021.9378741.

[25] Oukas, Nourredine and Menouar Boulif. "A Colored Petri Net to Model Message Differences in Energy Harvesting WSNs." Communication Systems and Applications (2020).

[26] Lages, D., Borba, E., Tavares, E. et al. A CPN-based model for assessing energy consumption of IoT networks. J Supercomput 79, 12978–13000 (2023). https://doi.org/10.1007/s11227-023-05185-4

[27] F. Correia, M. Alencar and K. Assis, "Stochastic Modeling and Analysis of the Energy Consumption of Wireless Sensor Networks," in IEEE Latin America Transactions, vol. 21, no. 3, pp. 434-440, March 2023, doi: 10.1109/TLA.2023.10068847.

[28] Chenchen, Z., Dongyin, L., Xuemei, X. and Yanhui, W., 2022. Modeling and analysis of global energy consumption process of urban rail transit system based on Petri net. Journal of Rail Transport Planning & Management, 21, p.100293.

[29] N. Oukas, M. Boulif, H. Hadiouche and C. Bengharabi, "A New Petri Nets for WSNs to Model the Behaviour of Solar-Energy Harvesting Sensors with Double Sleeping Strategy," 2022 2nd International Conference on Computing and Information Technology (ICCIT), pp. 237-242, 2022.

[30] Oukas, N., Boulif, M. Sensor Performance Evaluation for Long-Lasting EH-WSNs by GSPN Formulation, Considering Seasonal Sunshine Levels and Dual Standby Strategy. Arab J Sci Eng (2022)