A queueing model for a wireless sensor node using energy harvesting

Reference:
Blondia Christian.- A queueing model for a wireless sensor node using energy harvesting
Telecommunication systems: modeling, analysis, design and management - ISSN 1018-4864 - 77(2021), p. 335-349
Full text (Publisher's DOI): https://doi.org/10.1007/S11235-021-00758-1
To cite this reference: https://hdl.handle.net/10067/1760100151162165141

A Queueing Model for a Wireless Sensor Node using Energy Harvesting

Chris Blondia
IDLab, University of Antwerp – imec, Antwerp, Belgium
chris.blondia@uantwerpen.be
orcid.org/0000-0003-2194-9728

Abstract

In this paper we propose a generic queueing model that can be used to evaluate the performance of a wireless sensor node that uses energy harvesting. The alteration of such a device between the transmit and sleep mode (or between consuming energy and harvesting energy), is modeled by means of a finite capacity queueing system with repeated server vacations. The duration of a service, resp. vacation, is determined by the available energy at the start of the service, resp. vacation. Therefor we introduce in the model a variable that keeps track of the available energy. The system occupancy and the available energy are observed at inspection instants (i.e., the end of a service or of a vacation), resulting in a discrete-time Markov Chain. We derive closed form formulas for the system occupancy distribution at inspection instants and at arbitrary time instants together with the Laplace transform of the waiting time distribution. The possible use of the model to evaluate the system's performance for various parameter values is illustrated by means of a number of examples.

Keywords: Energy harvesting, WSN, server vacations, finite capacity queue, Markov chain

Funding: Flemish FWO project G0B7915N, "Modelling and Control of Energy Harvesting Wireless Sensor Networks"

Conflict of Interest: none

Availability of data and material: none

Code availability: none

1. Introduction

The Internet-of-Things (IoT) consists of interconnected edge devices, such as sensors, wearables, etc., that gather and process data and that communicate with each other. There are many areas of applications, such as smart homes, e-health, smart logistics, mobility, etc. These devices are usually powered by means of batteries. Hence the research effort to develop communication protocols that are energy aware (e.g., LoRaWAN, NB-IoT, etc.).
However, the battery lifetime limitations pose an important problem, e.g., in hard-to-reach or hostile areas or in massive-scale deployments, since the battery replacement does not only increase heavily the operational IoT network costs but is sometimes even impossible.
To cope with this problem, battery-less devices using energy harvesting seem to be a possible solution. These devices capture energy from the environment, accumulate it and convert it into electrical energy. Moreover, energy harvesting is more environmentally friendly than batteries. For an overview of energy harvesting in wireless sensor networks (WSN), we refer to e.g. [1], [2]. In such a device, the energy is harvested from the environment (solar, thermal, wind, hydro, etc.) and stored in a capacitor. During the transmission and the reception of packets or other communication related functions, the radio uses the stored energy. Packets can only be transmitted or received if the device has harvested enough energy and reaches a certain threshold. Below this threshold, the radio is put asleep and energy is harvested. Therefore, in between transmissions or receptions of packets, the radio may switch from an active to a sleep mode, during which energy is harvested, in order to reach the required level to transmit or to receive the next packet. Hence the need for energy-efficient communication protocols that help to increase the lifespan of these devices.
Keeping the energy consumption low, is key in the decision about when to be in an active mode or sleep mode. On the other hand, the application that runs on the IoT network, may have more or less stringent requirements with respect to its response time and possible loss of data. Hence the need for models that allow to predict the performance of these devices keeping track of the available energy.
The aim of this paper is to propose a queueing system that models the behavior of such a sensor node using energy harvesting. Key in this model is to keep track of the available energy. While sleeping, the device harvests energy according to a specific function. Packets are generated by the sensor and need to be transmitted to other nodes of the WSN, using the available energy according to another function. The energy harvesting and energy consumption process are described by means of these two functions. As the aim of the paper is not to study the energy harvesting process, no further details of these processes are required. The model allows to predict the system occupancy and the packet waiting time.
The paper is organized as follows. Section 2 presents a more detailed description of the system under study. Section 3 gives references to related work. Section 4 is devoted to the detailed description of the model and the determination of the performance measures. Numerical results in Section 5 show how the model can be used to investigate the performance of the system for different parameter values. Conclusions are drawn in Section 6.

2. System Description

The queueing model developed in this paper can be used to evaluate a WSN node consisting of the following components:

- A packet queue: stores packets that contain data obtained from measurements that need to be transmitted to another node (e.g., the sink or an access point)
- An energy harvesting system and capacitor: this subsystem provides the required energy to transmit packets.
- A radio that transmits the packets using the energy available in the capacitor.

A. The Harvesting System and the Capacitor

The energy harvesting system and the capacitor that are used in this paper have been introduced in [3]. Although a detailed harvester model should depend on the type of energy source that is used (vibration, thermal, solar, etc.), in this paper we consider a generic and simplified model. The harvester is modeled as a real DC (direct current) voltage source composed of an ideal DC voltage source and a series resistance. This series resistance limits the power of the harvester. In [3], Section III.A.a, it is shown that this harvester can be used to model the main suitable ambient energy sources for IoT: kinetic, solar, thermo-electric and RF. The capacitor is the part of the system that stores the required energy. Its behavior consists of a succession of intervals during which the capacitor is charged or discharged. The voltage of the capacitor during each interval is characterized by means of the initial voltage at the beginning of the interval and the temporal evolution at an arbitrary instant of the interval. The voltage $v(t)$ at time t spent in the current interval is given by the following formula:

$$v(t) = E\frac{R_{eq}}{r_i}\left(1 - e^{\frac{-t}{R_{eq}C}}\right) + v_0 e^{\frac{-t}{R_{eq}C}} \qquad (1)$$

where C is the capacitance in Farads, R_{eq} is the equivalent resistance of the circuit in Ω, r_i is the series resistance in Ω and v_0 the initial voltage at the start of the interval. The proposed model will be used in Section 5, where applications of the proposed queueing model will be presented. A more detailed description of the harvesting system and the corresponding circuitry can be found in [3].

B. Possible States of the Radio

In what follows we denote the available energy by the variable i. In accordance with [3], we impose that the energy should always remain larger than or equal to a minimum value denoted by i_{min}.

We assume that the radio may be in two states: transmitting a packet or sleeping. We assume that the transmission of a packet consists of different activities (such as inspection of the queue, carrier sensing, MAC protocol, actual transmission of the packet, etc.). The time needed to execute these functions will depend on the available energy at the moment of the inspection of the queue (e.g., the MCS used to transmit a packet may depend on the available energy). During the transmission of a packet the radio consumes energy. The function that governs this energy consumption is denoted by F, where $F(i_0, S_{i_0})$ equals the energy level at the end of the transmission process of a packet, given that at the starting instant the available energy is i_0 and the total time (including all related activities) needed to transmit the packet equals S_{i_0}. Remark that the function F can be derived from Formula (1). In principle, the different activities related to the transmission of a packet could involve different functions, as the energy consumption for the various activities may differ. However, to keep the model simple, we do not take this into account and incorporate all the activities in a single function F.

If the radio is sleeping, energy will be harvested. In this case, the energy harvesting is governed by a function G, where $G(i_0, V_{i_0})$ equals the energy level at the end of a sleeping period, given that at the starting instant the available energy is i_0 and the sleeping time equals V_{i_0}. We refer to this sleep period as the node being in the SLEEP state.

We denote the function that computes the time needed to reach the energy level i_1 starting from level i_0 (with $i_0 < i_1$) while being in a SLEEP state by $I(i_0, i_1)$, where $I(i_0, i_1) = T$ iff $G(i_0, T) = i_1$.

A packet can only be sent successfully if the available energy at the instant the queue is inspected is at least a known threshold i_{th}. This threshold is defined such that if the energy level is i_{th} at the start of a transmission, at the end of it, the energy equals i_{min}. Hence, the threshold i_{th} satisfies the relationship $F(i_{th}, S_{i_{th}}) = i_{min}$.

If at an inspection instant of the queue, it is not empty and the available energy equals i_0, two cases are possible: if $i_0 \geq i_{th}$, then the next packet in the queue is selected for transmission and this transmission will be completed after a time S_{i_0} and the energy level at the completion instant is given by $F(i_0, S_{i_0})$. If on the other hand $i_0 < i_{th}$, then the system goes into a sleep mode until the energy level i_{th} is reached. The time needed to attain this threshold is given by $I(i_0, i_{th})$. Hence if $i_0 < i_{th}$, then the time interval between the inspection instant of the queue and the completion of the transmission (hence the next inspection instant) is given by $I(i_0, i_{th}) + S_{i_{th}}$, and the energy level at that moment is given by $F(i_{th}, S_{i_{th}})$. We refer to the transmission of a packet together with a possible sleep period to reach the threshold, as the TRANSMIT state of the system.

Now assume that the queue is empty at the inspection instant, and the energy level is given by i_0. Then the device will enter the SLEEP state for a time interval that depends on i_0. Let the time that the system stays in the SLEEP state be V_{i_0}. The energy at the end of the SLEEP state can be computed as $i_1 = G(i_0, V_{i_0})$. At the end of this time interval the queue is inspected again. If the queue is not empty, the radio switches to the TRANSMIT state and again two cases are possible. The device starts the transmission of the first packet or starts a sleep period until it reaches the energy required to start the transmission, depending whether $i_1 \geq i_{th}$, or $i_1 < i_{th}$. If on the other hand the queue appears still to be empty, then the device starts a new SLEEP state, the duration of which is determined in a similar way as above. These consecutive SLEEP states take place as long as the queue is empty upon its inspection.

In order to be compatible with the proposed queueing model of Section 3, we introduce the following terminology. The time between the inspection of a non-empty queue and the end of the transmission of the first packet in the queue is referred to as a *service* (apart from the functions related to the transmission of a packet this may include a sleep phase if at the moment of the inspection the available energy level is lower than the threshold i_{th}). This service models exactly the TRANSMIT state. The time interval between the inspection of an empty queue and the next inspection instant (i.e., the SLEEP state) is called a *vacation*.

From the above system description, it is clear that given the system occupancy and the energy level at an inspection instant, it is possible to decide whether the time interval till the next inspection instant will be a service or a vacation. Moreover, the system occupancy and the energy level at the next inspection instant may be computed using the functions F and G. This forms the basis for the queueing model described in the Section 4.

3. Related Work

Energy harvesting in WSNs has been the subject of many papers. An overview can be found in e.g., [1] or [2]. In [4], a model for optimal collection in energy harvesting sensor networks is proposed. [5] evaluates the performance of a WSN node powered by RF energy harvesting regarding packet size and time ratio between standby and active state. MAC protocols that aim to maximize the lifetime of WSNs using energy harvesting are studied in [6] and [7]. The problem of retrieving data from a WSN within a fixed time window and with minimum energy consumption for the sensors is considered in [8].

A number of analytical models for energy consumption in WSN have been proposed. [9] proposes an event-driven Queueing Petri Net modeling technique to simulate the energy behaviors of nodes and to evaluate the system lifetime of the WSN. In [10] a method to determine the tradeoff between packet transmission speed and battery life in WSN was proposed, using a queueing model and a birth-death process capable of quantifying various energy consumption parameters. Switching the radio between busy and idle based on a certain threshold of the number of packets in the queue available for transmission, using a model that is based on a N-policy M/M/1 queue was studied in [11].

Energy harvesting WSN have been studied using analytical models. In [12], a model for a WSN node using energy harvesting is presented based on a threshold-controlled vacation policy. Vacations are repeated until N packets are accumulated in the queue. [13] presents a Markov model that integrates energy harvesting with the slotted CSMA/CA mechanism of IEEE802.15.4. In [14] a Markovian queueing model is proposed to investigate the impact of uncertainty in the energy harvesting process, the energy expenditure, the data acquisition and the data transmission. [15] uses a Markov model to compute the probability of event loss due to energy run out. A Markov model for the joint energy harvesting and communication analysis is presented in [16].

The system described in the previous section shows many similarities with a finite capacity queue where the server takes repeated vacations. Indeed, when the queue is inspected and it is not empty, then a packet will be transmitted and the time between this inspection and the next inspection can be considered as a *service* of a customer (remark that this service time may include a SLEEP state to reach the required threshold i_{th}). If upon the inspection instant the queue is empty, then the time between this inspection instant and the next inspection instant can be considered as a *server vacation*. This vacation is repeated as long as the queue is empty upon the inspection instant. This type of queueing system has been investigated in several papers, e.g. [17], [18], [19] and also in [20], [21] where multi-queueing systems with more general input processes have been considered. However, there is a major difference with these server vacation queueing models. While in the finite capacity model with repeated server vacations the service times and vacation times are independent generally distributed random variables, the duration of a service and the duration of a vacation in our system depends on the available energy level at the moment of inspection. Hence, the consecutive service times and vacation times are not independent any longer. If we want to apply the results of the vacation model, we need to keep track of the energy level and the vacation model needs to be adapted with respect to the duration of a service and a vacation.

There are a number of papers that consider queues where the service process is modulated by a Markov process (e.g. [22], [23]). In those papers the service speed (or the duration of the service of a customer) is determined by an external environment process. In our case, the service duration and the vacation duration are determined by a process that is governed by a variable the value of which is determined by the duration of the previous service or previous vacation and hence is not an external environment process. Furthermore, our system differs from the queueing model studied in [24] and related papers, where the type of the customer (in our case representing the available energy at the start of a service) depends both on the type of the previous customer and on the service duration of the previous customer. In our system, the type of the previous customer determines both the service duration of this customer and the type of the next customer. Moreover, we do not consider batch arrivals and in our model the server may take consecutive vacations.

4. System Model

4.1. The Queueing Model

We define a queueing system that models the system described in Section 2. We let customers arrive according to a Poisson process with rate λ. The system may hold at most N customers (including the customer being served). The distribution that defines the duration of a service, resp. vacation, is determined by the value of a variable E at the start of the service, resp. vacation (remark that E models the energy level in the system of Section 2). We let the variable E take integer values $1, \cdots, i_{max}$.
We let the duration of a service, resp. vacation, that starts with E equal to i, be denoted by S_i, with distribution $S_i(t)$ and Laplace Transform (LST) $S_i^*(\vartheta)$, resp. V_i, with distribution $V_i(t)$ and LST $V_i^*(\vartheta)$. Remark that the duration of a service consists of all tasks related to the transmission of packet, possibly preceded by a sleep period to reach the threshold i_{th}. The evolution of the variable E is governed by the following functions: if at the start of a service, resp. vacation, the variable $E=i$, then its value at the next inspection instant is given by $F(i, S_i)$ for a service, resp. $G(i, V_i)$ for a vacation.
Let $g_{k,i}$, resp. $h_{k,i}$, be the probability that n customers arrive during a service time, resp. a vacation time, that starts with $E=i$. Then clearly

$$g_{k,i} = \int_0^\infty \frac{(\lambda t)^k}{k!} e^{-\lambda t} dS_i(t) \qquad (2)$$

$$h_{k,i} = \int_0^\infty \frac{(\lambda t)^k}{k!} e^{-\lambda t} dV_i(t) \qquad (3)$$

In what follows we denote the column vectors of the LSTs of the service times and the vacation times by

$$\overline{\boldsymbol{S}}^*(\theta) = \left(S_1^*(\vartheta),\ S_2^*(\vartheta), \cdots,\ S_{j_{max}}^*(\vartheta) \right)' \qquad (4)$$

$$\overline{\boldsymbol{V}}^*(\theta) = \left(V_1^*(\vartheta),\ V_2^*(\vartheta), \cdots,\ V_{j_{max}}^*(\vartheta) \right)' \qquad (5)$$

and the column vectors of the averages of the service times and the vacation times by

$$\overline{\boldsymbol{E}[\boldsymbol{S}]} = (E[S_1], E[S_2], \cdots, E[S_{j_{max}}])' \qquad (6)$$

$$\overline{\boldsymbol{E}[\boldsymbol{V}]} = (E[V_1], E[V_2], \cdots, E[V_{j_{max}}])' \qquad (7)$$

4.2. The Embedded Markov Chain

In principle, such a queueing system is studied by observing the system at departure epochs (see e.g. [17], [19]). However, due to the fact that the variable E needs to be taken into account and that the length of a vacation and a service depend on the value of E at the start of the vacation and the service, we will observe the system at the queue inspection moments $t_n, n = 1,2, \ldots$ instead of the departure epochs.

We consider the stochastic process (Q_n, E_n) at these inspection instants t_n. Referring to the system described in Section 2, Q_n is the number of packets in the system at time instant t_n and E_n is the energy capacity available at time instant t_n. It is easy to check that this stochastic process is a discrete-time finite Markov Chain with state space size equal to $(N+1)\cdot j_{max}$ with transition probabilities shown in Figure 1. We denote the limiting probability distribution vector by $\overline{\boldsymbol{p}} = (\overline{\boldsymbol{p}}_{\boldsymbol{0}}, \overline{\boldsymbol{p}}_{\boldsymbol{1}}, \dots, \overline{\boldsymbol{p}}_{\boldsymbol{N-1}}, \overline{\boldsymbol{p}}_{\boldsymbol{N}})$, where

$$(\overline{\boldsymbol{p}}_{\boldsymbol{k}})_i = \lim_{n\to\infty} Prob\{Q_n = k, E_n = i\},\ 0 \le k \le N, i = 1, \dots, i_{max} \tag{8}$$

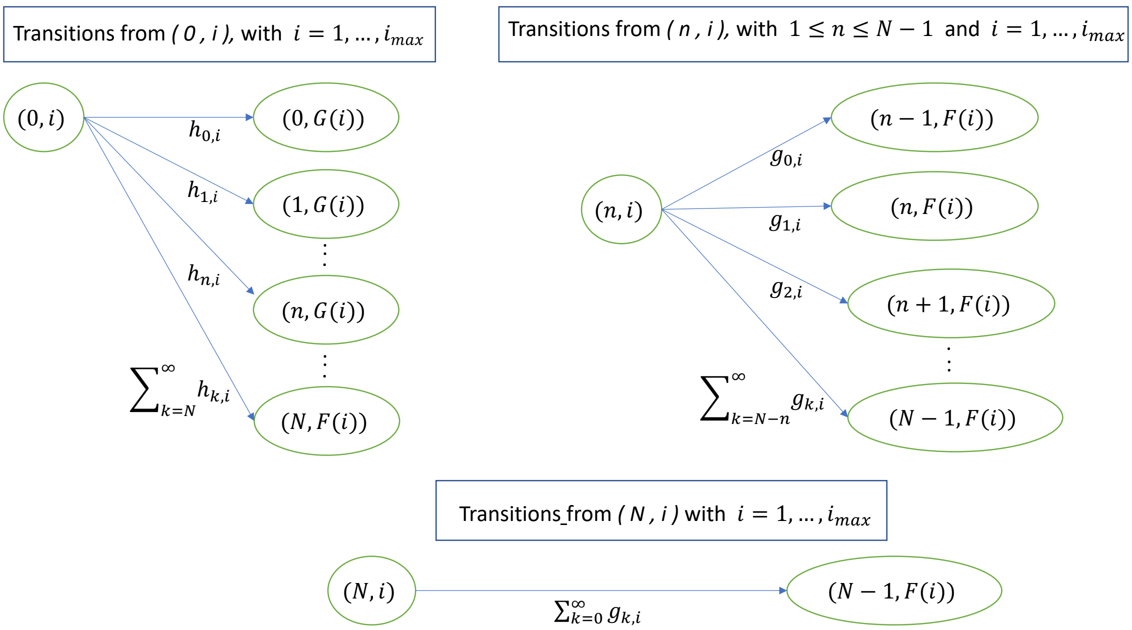

Figure 1: Markov Chain Transitions

The transition matrix of this Markov Chain (see Figure 1) is given by

$$\boldsymbol{P} = \begin{pmatrix} \boldsymbol{B_0} & \boldsymbol{B_1} & \boldsymbol{B_2} \cdots \boldsymbol{B_{N-2}} & \boldsymbol{B_{N-1}} & \sum_{n=N}^{\infty} \boldsymbol{B_n} \\ \boldsymbol{A_0} & \boldsymbol{A_1} & \boldsymbol{A_2} \cdots \boldsymbol{A_{N-2}} & \sum_{n=N-1}^{\infty} \boldsymbol{A_n} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A_0} & \boldsymbol{A_1} \cdots \boldsymbol{A_{N-3}} & \sum_{n=N-2}^{\infty} \boldsymbol{A_n} & \boldsymbol{0} \\ \ddots & \ddots & \ddots \quad \ddots & \ddots & \ddots \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \cdots \boldsymbol{A_0} & \sum_{n=1}^{\infty} \boldsymbol{A_n} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \cdots \boldsymbol{0} & \sum_{n=0}^{\infty} \boldsymbol{A_n} & \boldsymbol{0} \end{pmatrix} \tag{9}$$

The $i_{max} * i_{max}$ matrices $\boldsymbol{B_n}$ and $\boldsymbol{A_n}$ are given by

$$(\boldsymbol{B_n})_{i,j} = \begin{cases} \int_0^\infty \frac{(\lambda t)^n}{n!} e^{-\lambda t}\, dV_i(t) = h_{n,i}\,, & j = G(i), i = 1, \cdots, i_{max} \\ 0 \quad \text{elsewhere} \end{cases} \tag{10}$$

and

$$(\boldsymbol{A_n})_{i,j} = \begin{cases} \int_0^\infty \frac{(\lambda t)^n}{n!} e^{-\lambda t}\, dS_i(t) = g_{n,i}\,, & j = F(i), i = 1, \cdots, i_{max} \\ 0 \quad \text{elsewhere} \end{cases} \tag{11}$$

The distribution vector satisfies the following equations

$$\begin{aligned} \overline{\boldsymbol{p}} &= \overline{\boldsymbol{p}} \cdot \boldsymbol{P} \\ \overline{\boldsymbol{p}} \cdot \overline{\boldsymbol{e}} &= 1 \end{aligned} \tag{12}$$

where $\overline{\boldsymbol{e}}$ is the $(N+1) \cdot i_{max}$ unit vector.
The system occupancy distribution at inspection epochs is then given by

$$\nu_n = \overline{\boldsymbol{p}}_{\boldsymbol{n}} \cdot \overline{\boldsymbol{e}}, \qquad 0 \le n \le N \tag{13}$$

4.3. System Occupancy Probability at Arbitrary Instants

Once the probability vector $\overline{\boldsymbol{p}}$ is determined, it is possible to compute the steady state probability of the system occupancy at arbitrary time instants. To obtain these probabilities, we follow a reasoning similar to the one in [18].
Let t be an arbitrary time instant. Let $(\overline{\boldsymbol{\omega}}_{\boldsymbol{n}})_i(t)dt$ the probability that t falls in a vacation that starts with the energy level equal to *i*, that at time t there are *n* packets in the queue and the remaining time of the vacation $\widetilde{V}_i$ satisfies $t < \widetilde{V}_i \le t + dt$. Similarly, let $(\overline{\boldsymbol{\pi}}_{\boldsymbol{n}})_i(t)$ be the probability that t falls in a service, that at the start of this service the energy level was *i*, that at time t there are *n* packets in the queue and the remaining time of the service $\widetilde{S}_i$ satisfies $t < \widetilde{S}_i \le t + dt$. We denote the corresponding LST vectors by

$$\overline{\boldsymbol{\omega}}_{\boldsymbol{n}}^*(\theta) = \int_0^\infty e^{-\theta t}\ \overline{\boldsymbol{\omega}}_{\boldsymbol{n}}(t)dt \tag{14}$$

$$\overline{\boldsymbol{\pi}}_{\boldsymbol{n}}^*(\theta) = \int_0^\infty e^{-\theta t}\ \overline{\boldsymbol{\pi}}_{\boldsymbol{n}}(t)dt \tag{15}$$

Clearly $\overline{\boldsymbol{\omega}}_{\boldsymbol{n}}^*(0) \cdot \overline{\boldsymbol{e}}$, resp. $\overline{\boldsymbol{\pi}}_{\boldsymbol{n}}^*(0) \cdot \overline{\boldsymbol{e}}$, is the steady state vector that an arbitrary time instant falls in a vacation, resp. service, and the number of packets in the system is n. The distribution of the number of packets in the system at an arbitrary time instant is given by

$$\eta_0 = \overline{\boldsymbol{\omega}}_{\boldsymbol{0}}^*(0) \cdot \bar{e} \tag{16}$$

$$\eta_n = \overline{\boldsymbol{\omega}}_{\boldsymbol{n}}^*(0) \cdot \overline{\boldsymbol{e}} + \overline{\boldsymbol{\pi}}_{\boldsymbol{n}}^*(0) \cdot \overline{\boldsymbol{e}}, \quad n = 1, \dots, N \tag{17}$$

Let us now compute $\overline{\boldsymbol{\omega}}_{\boldsymbol{n}}^*(\theta) \cdot \overline{\boldsymbol{e}}$ and $\overline{\boldsymbol{\pi}}_{\boldsymbol{n}}^*(\theta) \cdot \overline{\boldsymbol{e}}$.
Let ρ_i, resp. σ_i be the probability that an arbitrary time instant falls in a service, resp. a vacation, that started with energy level *i*.
Clearly

$$\rho_i = \frac{\sum_{n=1}^{N} (\overline{\boldsymbol{p}}_{\boldsymbol{n}})_i * E[S_i]}{\sum_{j=1}^{i_{max}} [\sum_{n=1}^{N} (\overline{\boldsymbol{p}}_{\boldsymbol{n}})_j * E[S_j] + (\overline{\boldsymbol{p}}_{\boldsymbol{0}})_j * E[V_j]]} \quad (18)$$

$$\sigma_i = \frac{(\overline{\boldsymbol{p}}_{\boldsymbol{0}})_i * E[V_i]}{\sum_{j=1}^{i_{max}} [\sum_{n=1}^{N} (\overline{\boldsymbol{p}}_{\boldsymbol{n}})_j * E[S_j] + (\overline{\boldsymbol{p}}_{\boldsymbol{0}})_j * E[V_j]]} \quad (19)$$

The denominator of the above probabilities can be simplified as follows.
The overall average service time (i.e. the average over all possible energy levels) is given by

$$E[S] = \frac{\sum_{n=1}^{N} \overline{\boldsymbol{p}}_{\boldsymbol{n}} \cdot \overline{\boldsymbol{E[S]}}}{1 - \overline{\boldsymbol{p}}_{\boldsymbol{0}} \cdot \overline{\boldsymbol{e}}} \quad (20)$$

and the overall average vacation time (i.e. the average over all possible energy levels) is given by

$$E[V] = \frac{\overline{\boldsymbol{p}}_{\boldsymbol{0}} \cdot \overline{\boldsymbol{E[V]}}}{\overline{\boldsymbol{p}}_{\boldsymbol{0}} \cdot \overline{\boldsymbol{e}}} \quad (21)$$

Hence,

$$\rho_i = \frac{\sum_{n=1}^{N} (\overline{\boldsymbol{p}}_{\boldsymbol{n}})_i * E[S_i]}{E[S] \cdot [1 - \overline{\boldsymbol{p}}_{\boldsymbol{0}} \cdot \overline{\boldsymbol{e}}] + E[V] \cdot \overline{\boldsymbol{p}}_{\boldsymbol{0}} \cdot \overline{\boldsymbol{e}}} \quad (22)$$

and

$$\sigma_i = \frac{(\overline{\boldsymbol{p}}_{\boldsymbol{0}})_i * \mathrm{E}[V_i]}{E[S] \cdot [1 - \overline{\boldsymbol{p}}_{\boldsymbol{0}} \cdot \overline{\boldsymbol{e}}] + E[V] \cdot \overline{\boldsymbol{p}}_{\boldsymbol{0}} \cdot \overline{\boldsymbol{e}}} \quad (23)$$

Remark that the denominator $D = E[S] \cdot [1 - \overline{\boldsymbol{p}}_{\boldsymbol{0}} \cdot \overline{\boldsymbol{e}}] + E[V] \cdot \overline{\boldsymbol{p}}_{\boldsymbol{0}} \cdot \overline{\boldsymbol{e}}$ is nothing else but the average time between two inspection instants.
The carried load, defined as the fraction of time the server is busy is then given by

$$\rho = \frac{\sum_{n=1}^{N} \overline{\boldsymbol{p}}_{\boldsymbol{n}} \cdot \overline{\boldsymbol{E[S]}}}{D} \quad (24)$$

In Appendix 1 we derive the following explicit formulas for $(\overline{\boldsymbol{\omega}}_{\boldsymbol{n}}^*)(0) \cdot \overline{\boldsymbol{e}}$ and $(\overline{\boldsymbol{\pi}}_{\boldsymbol{n}}^*)(0) \cdot \overline{\boldsymbol{e}}$:

$$(\overline{\boldsymbol{\omega}}_{\boldsymbol{n}}^*)(0) \cdot \overline{\boldsymbol{e}} = \frac{1}{\lambda \cdot D} \cdot \overline{\boldsymbol{p}}_{\boldsymbol{0}} \cdot \left[\overline{\boldsymbol{e}} - \sum_{k=0}^{n} \boldsymbol{B}_{\boldsymbol{k}} \cdot \overline{\boldsymbol{e}} \right], n = 0, \dots, N-1 \quad (25)$$

$$(\overline{\boldsymbol{\omega}}_{\boldsymbol{N}}^*)(0) \cdot \overline{\boldsymbol{e}} = \frac{1}{D} \cdot \overline{\boldsymbol{p}}_{\boldsymbol{0}} \cdot \overline{\boldsymbol{E[V]}} - \sum_{n=0}^{N-1} (\overline{\boldsymbol{\omega}}_{\boldsymbol{n}}^*)(0) \cdot \overline{\boldsymbol{e}} \quad (26)$$

$$(\overline{\boldsymbol{\pi}}_{\boldsymbol{n}}^*)(0) \cdot \overline{\boldsymbol{e}} = \frac{1}{\lambda \cdot D} \cdot \left[\overline{\boldsymbol{p}}_{\boldsymbol{n}} \cdot \overline{\boldsymbol{e}} - \overline{\boldsymbol{p}}_{\boldsymbol{0}} \cdot \overline{\boldsymbol{e}} + \sum_{k=0}^{n-1} \overline{\boldsymbol{p}}_{\boldsymbol{0}} \cdot \boldsymbol{B}_{\boldsymbol{k}} \cdot \overline{\boldsymbol{e}} \right], n = 0, \dots, N-1 \quad (27)$$

$$(\overline{\boldsymbol{\pi}}_{\boldsymbol{N}}^*)(0) \cdot \overline{\boldsymbol{e}} = \frac{1}{D} \cdot \sum_{n=1}^{N-1} \overline{\boldsymbol{p}}_{\boldsymbol{k}} \cdot \overline{\boldsymbol{E[S]}} - \sum_{n=0}^{N-1} (\overline{\boldsymbol{\pi}}_{\boldsymbol{n}}^*)(0) \cdot \overline{\boldsymbol{e}} \quad (28)$$

The system occupancy distribution at an arbitrary time instant is then given by

$$\eta_n = \overline{\boldsymbol{\omega}}_{\boldsymbol{n}}^*(0) \cdot \overline{\boldsymbol{e}} + \overline{\boldsymbol{\pi}}_{\boldsymbol{n}}^*(0) \cdot \overline{\boldsymbol{e}} = \frac{1}{\lambda \cdot D} \cdot [\overline{\boldsymbol{p}}_{\boldsymbol{n}} \cdot \overline{\boldsymbol{e}} - \overline{\boldsymbol{p}}_{\boldsymbol{0}} \cdot \boldsymbol{B}_{\boldsymbol{n}} \cdot \overline{\boldsymbol{e}}], \qquad n = 0, \dots, N-1 \quad (29)$$

$$\eta_N = \overline{\boldsymbol{\omega}}_{\boldsymbol{N}}^*(0) \cdot \overline{\boldsymbol{e}} + \overline{\boldsymbol{\pi}}_{\boldsymbol{N}}^*(0) \cdot \overline{\boldsymbol{e}} = 1 - \frac{1}{\lambda \cdot D} \cdot [1 - \overline{\boldsymbol{p}}_{\boldsymbol{0}} \cdot \overline{\boldsymbol{e}}] \quad (30)$$

The probability that a customer finds the system full upon arrival (and hence is dropped) is given by

$$P_b = \eta_N \tag{31}$$

and the actual arrival rate is given by

$$(1 - P_b) \cdot \lambda = \frac{1}{D} \cdot [1 - \overline{\boldsymbol{p}}_{\boldsymbol{0}} \cdot \overline{\boldsymbol{e}}] \tag{32}$$

The average system occupancy is given by

$$E[\eta] = \sum_{n=1}^{N} n \cdot \eta_n \tag{33}$$

Remarks

Remark that the results for η_n, $0 \leq n \leq N-1$, in (29) correspond with the Formulas (9) in [18]. Indeed, $\overline{\boldsymbol{p}}_{\boldsymbol{n}} \cdot \overline{\boldsymbol{e}}$ is the probability that at an inspection instant the system occupancy equals n (be it the end of a service or a vacation interval) and $\overline{\boldsymbol{p}}_{\boldsymbol{0}} \cdot \boldsymbol{B}_{\boldsymbol{n}} \cdot \overline{\boldsymbol{e}}$ is the probability that at the end of a vacation the system occupancy equals n. Hence $\overline{\boldsymbol{p}}_{\boldsymbol{n}} \cdot \overline{\boldsymbol{e}} - \overline{\boldsymbol{p}}_{\boldsymbol{0}} \cdot \boldsymbol{B}_{\boldsymbol{n}} \cdot \overline{\boldsymbol{e}}$ is the probability that at the end of a service the system occupancy equals n, which is exactly in agreement with Formula (9) in [18]. It would also have been possible to follow exactly the same approach of [18] to define the embedded Markov Chain by making a distinction between end points of a service and end points of a vacation. Then we would have obtained the same formulas as in [18], however, the complexity to compute the steady state distribution at those embedded instants would have been larger since the size of the transition matrix $\boldsymbol{P}$ would have been twice as large, namely 2$\cdot$ $(N+1) \cdot i_{max}$.
Moreover, the results (29) and (30) are also in agreement with Formulas (5) in [17], where the distribution of the system occupancy at an arbitrary time instant is obtained from the distribution of the system occupancy at an embedded time instant divided by the customer arrival rate times the average time between embedded time instants.

4.4. Waiting Time Distribution

The LST of waiting time of a customer arriving at an arbitrary time instant at the system when the energy at the previous inspection instant was i, is given by

$$(\overline{\boldsymbol{W}}^*)_i(\theta) = \frac{1}{1-P_b}[\sum_{n=1}^{N-1} (\overline{\boldsymbol{\pi}}_{\boldsymbol{n}}^*)_i(\theta) \cdot S_{F(i)}^*(\theta) \cdots S_{F^n(i)}^*(\theta)$$
$$+ \sum_{n=0}^{N-1} \left(\overline{\boldsymbol{\omega}}_{\boldsymbol{n}}^* \right)_i(\theta) \cdot S_{F(G(i))}^*(\theta) \cdots S_{F^{n+1}(G(i))}^*(\theta) \Big] \tag{34}$$

To avoid complex notations, we have used $F(i)$, resp. $G(i)$ instead of $F(i, S_i)$, resp. $G(i, V_i)$, and $F^n(i) = F \circ F \circ \cdots \circ F(i)$ (n times).

Let $\overline{\boldsymbol{W}}^*(\theta)$ =$((\overline{\boldsymbol{W}}^*)_1(\theta), (\overline{\boldsymbol{W}}^*)_2(\theta), \cdots, (\overline{\boldsymbol{W}}^*)_{i_{max}}(\theta))$, then

$$\overline{\boldsymbol{W}}^*(\theta) = \frac{1}{1-P_b}[\sum_{n=1}^{N-1} \overline{\boldsymbol{\pi}}_{\boldsymbol{n}}^*(\theta) \cdot Diag(\prod_{k=1}^{n} S_{F^k(1)}^*(\theta), \cdots, \prod_{k=1}^{n} S_{F^k(i_{max})}^*(\theta))$$
$$+ \sum_{n=0}^{N-1} \overline{\boldsymbol{\omega}}_{\boldsymbol{n}}^*(\theta) \cdot Diag(\prod_{k=1}^{n+1} S_{F^k(G(1))}^*(\theta), \cdots, \prod_{k=1}^{n+1} S_{F^k(G(i_{max}))}^*(\theta))] \tag{35}$$

The average waiting time is then given by

$$E[W] = -\frac{d\overline{\boldsymbol{W}}^*(\theta)}{d\theta}|_{\theta=0} \cdot \overline{\boldsymbol{e}} \quad (36)$$

In Appendix 2, we show that this results in Little's well-known formula, namely

$$E[W] = \frac{1}{1-P_b} \cdot \frac{1}{\lambda} \cdot E[\eta] \quad (37)$$

5. Numerical Examples

In this section we apply the above model to the transmission system described in Section 2. The model has been implemented in MATLAB. The purpose of these numerical examples is not to investigate in detail the performance of a communication node using energy harvesting, but rather illustrate the potential use of the model to evaluate such systems.

In order to model the energy harvesting and consumption we use the function that was introduced in [3]. Remark however that this is to be considered as an example and that other functions could be used as well.

Assume that the device is in state X, (where X may by SLEEP or packet transmit (TX)) during an interval I. The evolution in time of the available energy during the interval I depends on this state X and the available energy at time t=0. More formally, it is described by the following function (see Formula (1)):

$$f(X, i_0, t) = c(X) \cdot \left(1 - e^{-\frac{t}{a(X)}}\right) + i_0 \cdot e^{-\frac{t}{a(X)}}, \qquad t \in I \quad (38)$$

where $c(X)$ and $a(X)$ are parameters depending on the state X, and i_0 is the energy available at time $t = 0$ and t is a time instant of the interval I. Using the notations of the system description in Section 2, $F(i_0, S_{i_0}) = f(\text{TX}, i_0, S_{i_0})$ and $G(i_0, V_{i_0}) = f(\text{SLEEP}, i_0, V_{i_0})$.

Remarks

- The values of the function f are real numbers. In the model, we have assumed that the variable E take integer values $1, \cdots, i_{max}$. When applying the model, the values of f are multiplied by 100 or 1000 and E is the integer part of the result of this multiplication.
- Depending on the device (capacitor, harvesting function) and the state X (TX or SLEEP), the slope of f at time t may be positive, zero or negative, depending on the values of the parameters $c(X)$ and $a(X)$. Clearly when the state is SLEEP, the function f is increasing while during packet transmission, it is decreasing.
- If the device remains in state X, then the energy level tends to $c(X)$ for increasing t. In the next examples, the maximum energy level is equal to $i_{max} = \max_{X=\text{SLEEP,TX}} c(X)$.

In Table 1 the values for $c(X)$ and $a(X)$ used in the examples are given. Clearly $i_{max} = 3.24$.

	SLEEP	TX
c	3.24	0.03
a	50.27	0.20

Table 1: parameters of the energy function f

The graphs in Figure 2 and Figure 3 show the increase of f for *X*=SLEEP, i.e., for energy harvesting, and the decrease of f for *X*=TX, i.e., for energy consumption.
In Figure 2, the function f for *X*=SLEEP is increasing to the value $c(SLEEP) = 3.24$, while in Figure 3, the function f for *X*=TX decreases to $c(TX) = 0.03$.

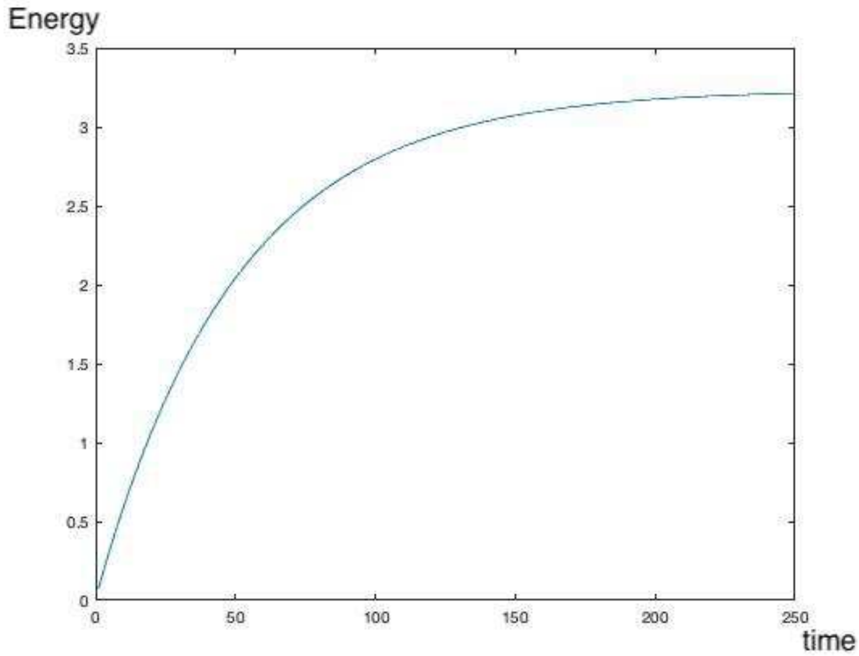

Figure 2: the energy function f for *X*=SLEEP

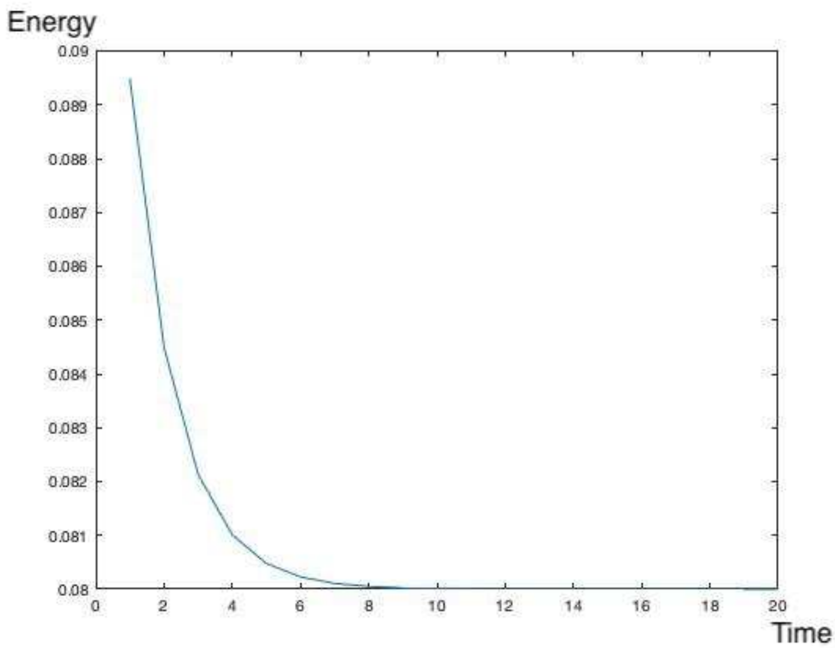

Figure 3: the energy function f for *X*=TX

In the following examples the node may contain *N=5* packets. We also assume that the distributions of the time needed to transmit a packet and duration of the SLEEP state both are deterministic, hence have constant length. Remark that this constant length depends on the available energy at the start of the state.

5.1. Example 1

In this example we study the influence of the length of the duration of the SLEEP state when the queue is empty (i.e., the duration of the vacation time) on the average system occupancy and the average packet waiting time for different values of the input rate.

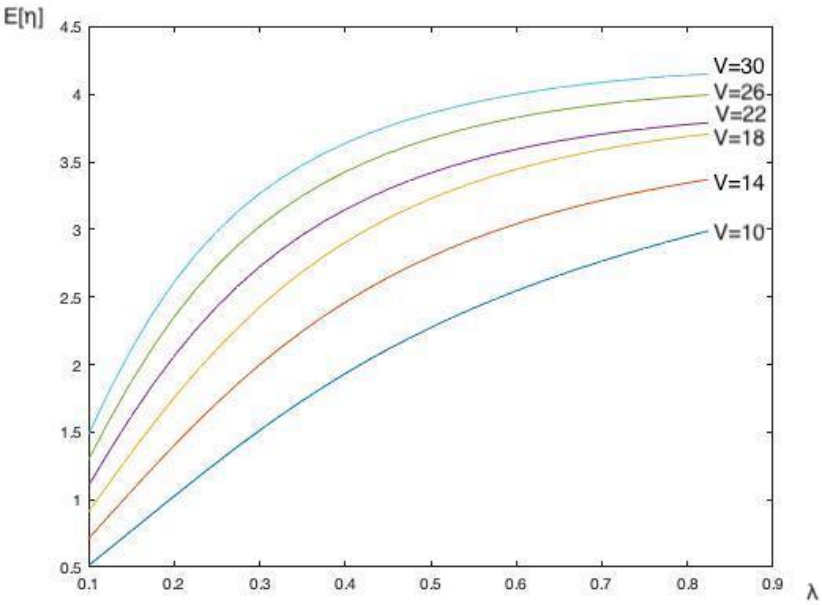

Figure 4: Average System Occupancy as a function of the input rate for variable SLEEP duration

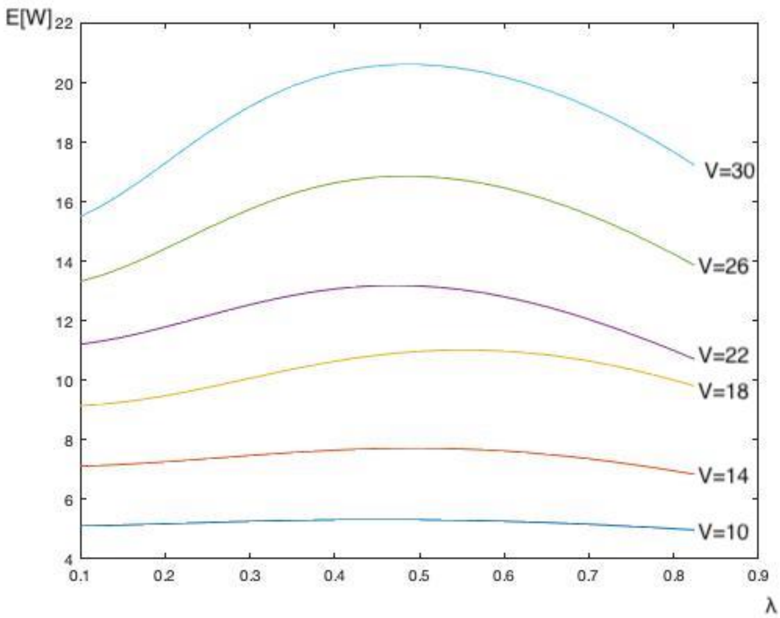

Figure 5: Average waiting time as a function of the input rate for variable SLEEP duration

The time needed to transmit a packet is 0.1 time units. We evaluate the average system occupancy (Figure 4) and the average waiting time (Figure 5) for variable input rate, λ varying between 0.10 and 0.85, for variable length of the SLEEP state, V varying between 10 and 30 time units.
Clearly both the average system occupancy and the average waiting time increase for increasing vacation time V and fixed arrival rate. Indeed, the longer the vacation time, the more packets are stored in the queue during such a vacation. This leads to a higher system occupancy and longer waiting times.
An increase of the average waiting time can be observed when the arrival rate of packets increases for lower values of the arrival rate. However, for higher arrival rates, the probability that a vacation occurs decreases and hence the average waiting time decreases. This phenomenon is more distinct in case of higher values of V as shown in Figure 5.

5.2. Example 2

In this example we investigate the influence of the energy threshold i_{th}. The system parameters are the same as in Example 1: N=5, The time needed to transmit a packet equals 0.1 time units and the length of the SLEEP state is 20 time units. The threshold i_{th} was defined in Section 2 as the minimal available energy required to allow a packet to be transmitted successfully. We let i_{th} very between 1.5 and 2.7; clearly for all these values packet transmissions are successful. We investigate the influence of i_{th} for the input rate λ varying between 0.1and 0.85. From Figure 6 and Figure 7 it is clear that the average system occupancy and the average waiting time increase for increasing energy threshold i_{th} due to the fact that packets have to wait extra until the energy threshold is reached.

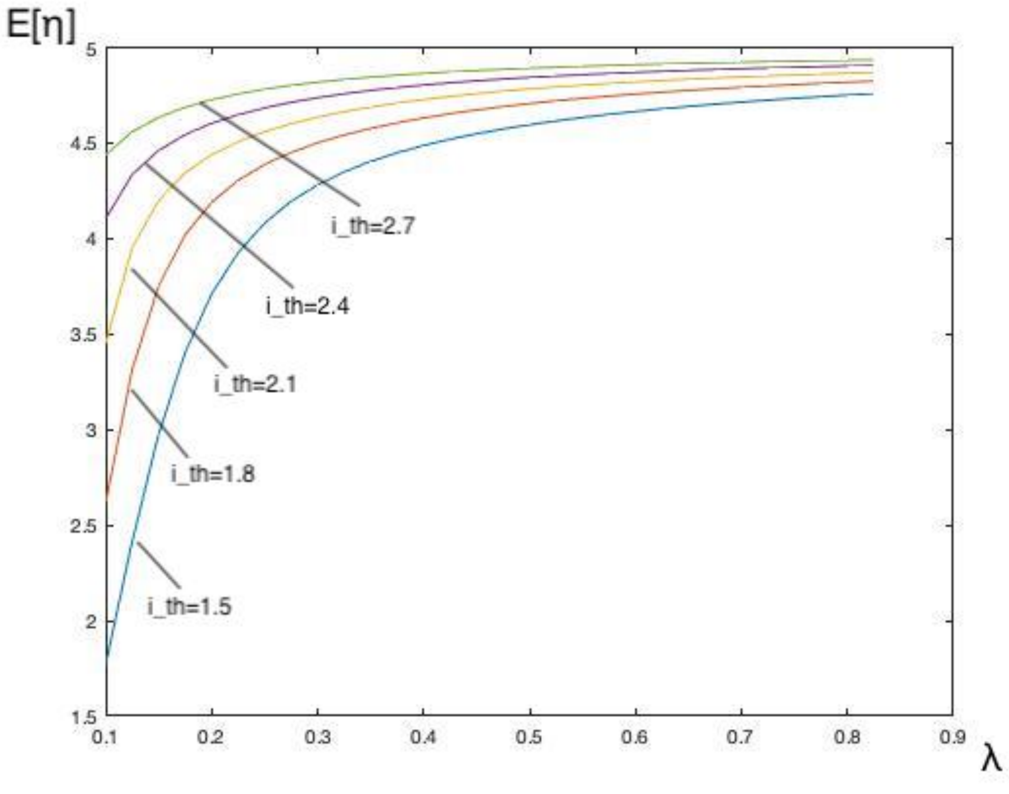

Figure 6: Average System Occupancy as a function of the input rate for variable energy threshold

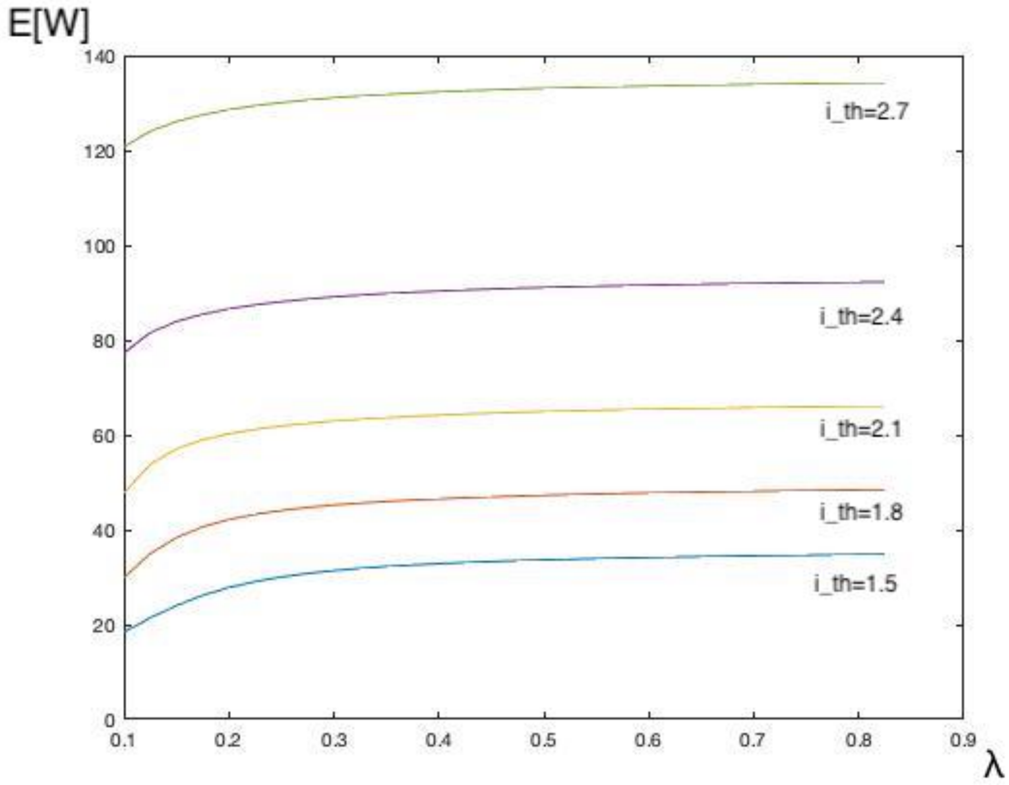

Figure 7: Average waiting time as a function of the input rate for variable energy threshold

5.3. Example 3

In this example we let the duration of the SLEEP phase be such that during a SLEEP phase the energy level increases by a value equal to Δ. This implies that the duration of the SLEEP phase is no longer constant but depends on the value of the energy at the start of the SLEEP phase. Again, we let N=5 and the time needed to transmit a packet 0.1 time units.

If the energy at the start of the SLEEP phase is i_0, then the time T needed to reach the energy level $i_0 + \Delta$ while in SLEEP state is given by

$$T = -a(SLEEP) \cdot \ln\left[\frac{i_0 + \Delta - c(SLEEP)}{i_0 - c(SLEEP)}\right] \tag{39}$$

We evaluate the average system occupancy (Figure 5) and the average waiting time (Figure 6) for variable input rate, λ varying between 0.05 and 0.80, for variable energy increase values Δ, varying between 0.3 and 1.50.

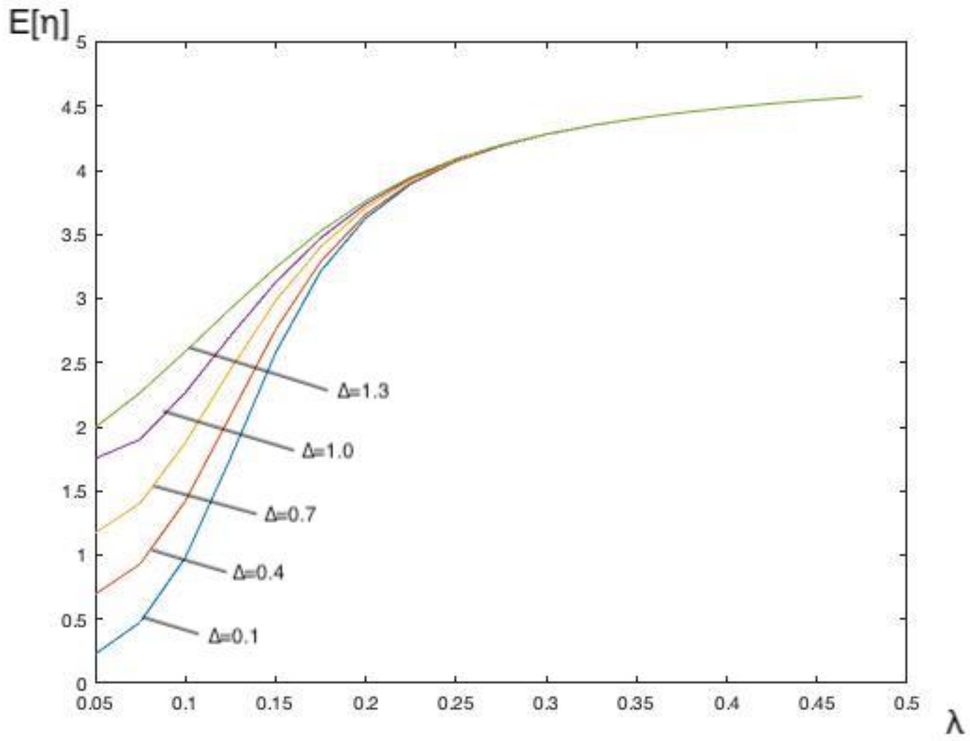

Figure 8: Average System Occupancy as a function of the input rate for variable energy increase while in SLEEP state

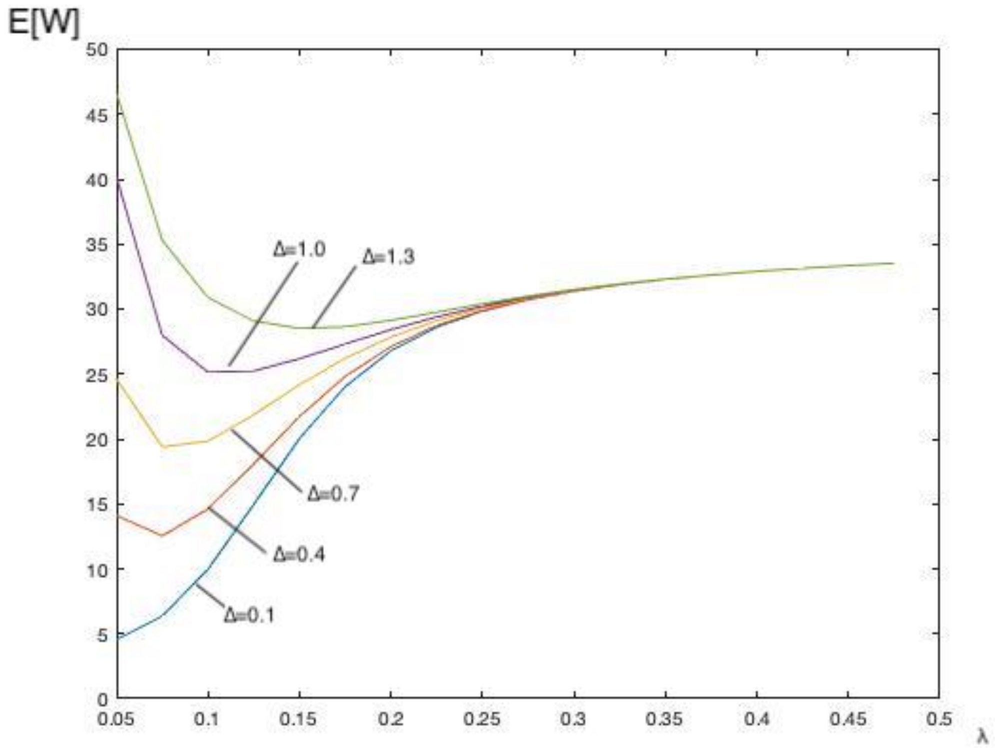

Figure 9: Average waiting time as a function of the input rate for variable energy increase while in SLEEP state

Clearly the average system occupancy increases for increasing values of the input rate. The higher the value of Δ, the higher the average system occupancy, but as the input rate increases the results for the average system occupancy converge. For low values of the input rate and higher values of Δ, Figure 9 shows a decrease of the average waiting time for increasing input rate. This is due to the fact that when the input rate is low, the system is often empty and therefore vacations take place frequently. When the input rate increases, vacations start to take place less frequently and the average waiting time decreases. As the input rate continues to increase, less vacations occur and the average waiting time increases towards a value that is independent from Δ, due to the fact that more packets arrive.

5.4. Example 4

In this example we illustrate the generality of the model. In many WSN protocols, complex link layer schemes are avoided and therefore, to increase the reliability, packets are sent several times. In this example we assume that each packet needs to be transmitted K times, $K > 1$. If between two consecutive transmissions of the same packet, the available energy drops below the threshold i_{th}, then the radio will switch to the sleep mode until the threshold is reached and the next transmission can take place. The length S_i of the TRANSMIT state, in case the energy level equals i at the moment the queue is inspected, is then the sum of K transmissions with possible sleep periods in between the K consecutive transmissions of the packet. Hence, given the energy level i, we need to compute the time till the next inspection instant, together with the energy level at this next inspection instant, i.e., after the K^{th} transmission of the packet. Once the values for S_i and the energy levels at inspection instants are known, the application of the model leads to the average system occupancy and average waiting time.
We let $N = 5$ and the length of the SLEEP state be 20 time units.

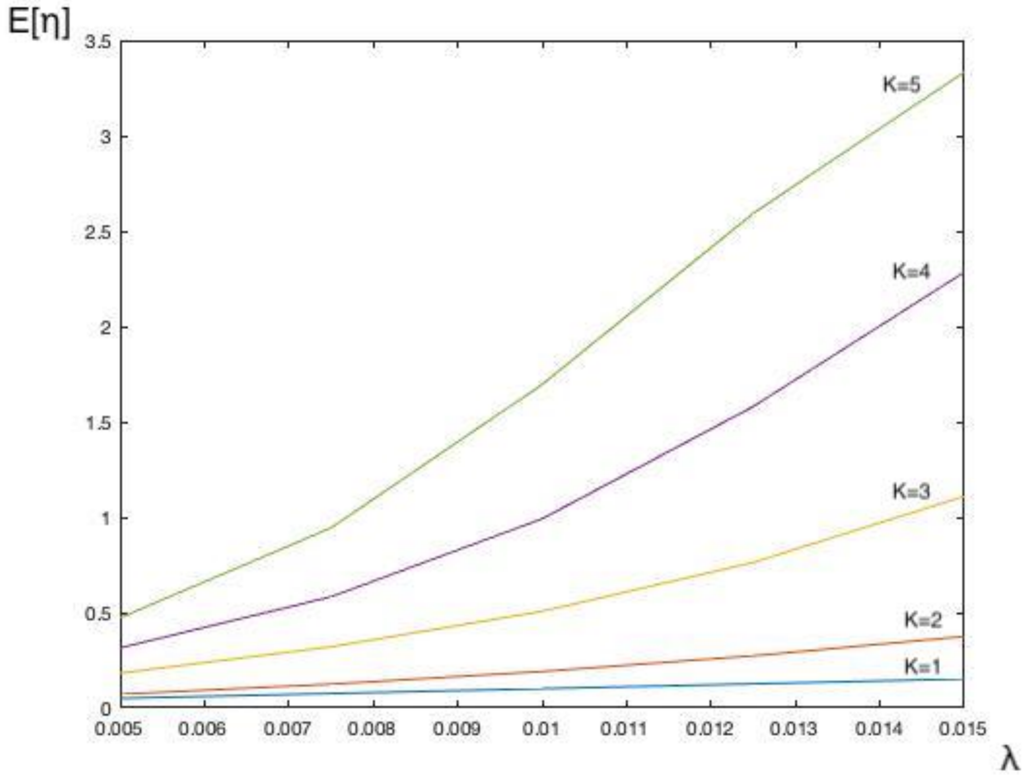

Figure 10: Average System Occupancy as a function of the input rate for variable packet repetitions

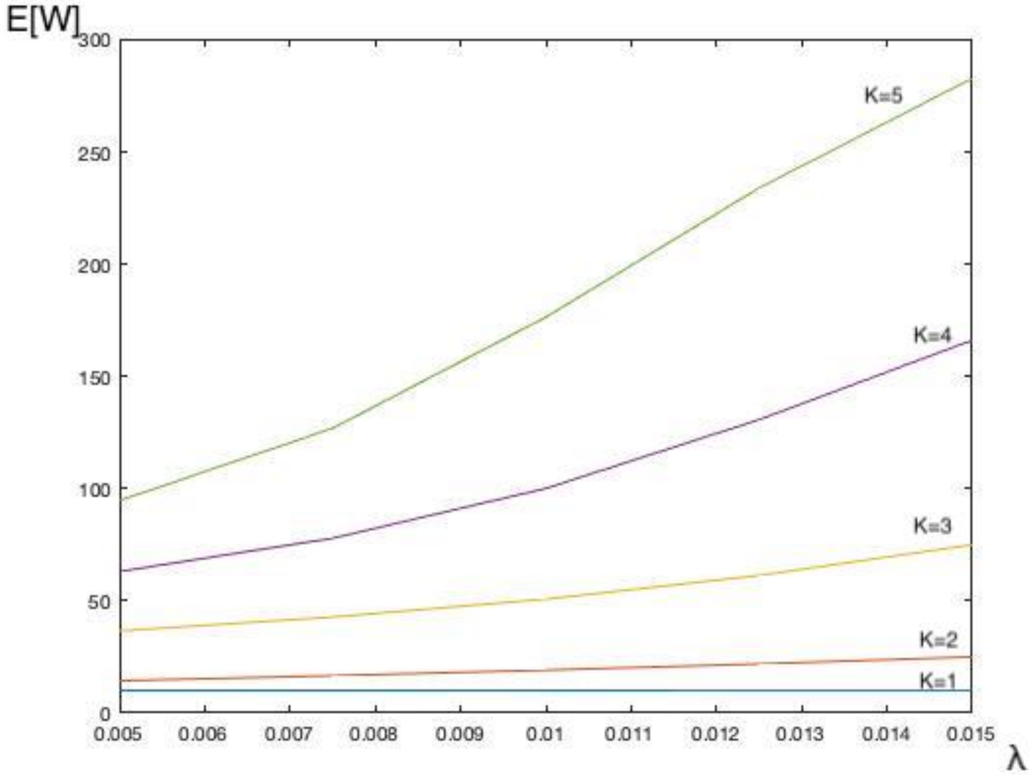

Figure 11: Average waiting time as a function of the input rate variable packet repetitions

Consider the system in which the duration of transmitting a single packet equals 0.1 time units. We evaluate the average system occupancy (Figure 10) and the average waiting time (Figure 11) for variable input rate, λ varying between 0.005 and 0.015, for variable number of repetitions of packet transmissions, namely $K = 1, \cdots, 5$. From these results we see that both the average system occupancy and average waiting time drastically increase for increasing K, due to the fact that the energy often drops below the threshold and the time it takes to reach the threshold again heavily contributes to these performance measures.

6. Conclusion

In this paper we have proposed a model for a wireless sensor node in a WSN that uses energy harvesting. This model is a finite capacity queue with repeated server vacations, where the length of a service and the length of a vacation depends on a variable the value of which is determined by the result of a function applied to the previous value of that variable and the length of the service, resp. the vacation. For this model we obtain the distribution of the system occupancy at inspection instants and at arbitrary instants and the LST of the waiting time distribution. We illustrate the use of the model through a number of examples. The model can be used for more complex protocols as shown by the last example. In the future, possible extensions of the model e.g., where the reception of packets is considered, could be envisaged.

Appendix 1

We derive explicit formulas for $(\overline{\boldsymbol{\omega}}_{\boldsymbol{n}}^{*})(0)\cdot\overline{\boldsymbol{e}}$ and $(\overline{\boldsymbol{\pi}}_{\boldsymbol{n}}^{*})(0)\cdot\overline{\boldsymbol{e}}$.
In a similar way as in [18], it is possible to show that for $n=0,\cdots,N-1$

$$(\overline{\boldsymbol{\omega}}_{\boldsymbol{n}}^{*})_i(\theta)=\sigma_i\cdot\{\frac{1}{(\lambda-\theta)V_i}[V_i^*(\vartheta)(\frac{\lambda}{\lambda-\theta})^n-\sum_{k=0}^{n}h_{k,i}\cdot(\frac{\lambda}{\lambda-\theta})^{n-k}\}$$
$$=\frac{1}{(\lambda-\theta)D}\cdot(\bar{p}_0)_i\cdot\{V_i^*(\vartheta)\cdot(\frac{\lambda}{\lambda-\theta})^n-\sum_{k=0}^{n}h_{k,i}\cdot(\frac{\lambda}{\lambda-\theta})^{n-k}\},$$

and hence

$$(\overline{\boldsymbol{\omega}}_{\boldsymbol{n}}^{*})(\theta)\cdot\overline{\boldsymbol{e}}=\frac{1}{(\lambda-\theta)D}\cdot\{\overline{\boldsymbol{p}}_{\boldsymbol{0}}\cdot\overline{\boldsymbol{V}}^*(\vartheta)\cdot(\frac{\lambda}{\lambda-\theta})^n-\sum_{k=0}^{n}\overline{\boldsymbol{p}}_{\boldsymbol{0}}\cdot\boldsymbol{B}_{\boldsymbol{k}}\cdot\overline{\boldsymbol{e}}\cdot(\frac{\lambda}{\lambda-\theta})^{n-k}\tag{40}$$

and the probability of having a system occupancy equal to n at an arbitrary instant of a vacation interval is given by

$$(\overline{\boldsymbol{\omega}}_{\boldsymbol{n}}^{*})(0)\cdot\overline{\boldsymbol{e}}=\frac{1}{\lambda\cdot D}\cdot\overline{\boldsymbol{p}}_{\boldsymbol{0}}\cdot\left[\overline{\boldsymbol{e}}-\sum_{k=0}^{n}\boldsymbol{B}_{\boldsymbol{k}}\cdot\overline{\boldsymbol{e}}\right]\tag{41}$$

For n=N, we obtain

$$(\overline{\boldsymbol{\omega}}_{\boldsymbol{N}}^{*})_i(\theta)=\sigma_i\cdot\{\frac{1}{(\lambda-\theta)V_i}\sum_{n=N}^{\infty}[V_i^*(\vartheta)(\frac{\lambda}{\lambda-\theta})^n-\sum_{k=0}^{n}h_{k,i}\cdot(\frac{\lambda}{\lambda-\theta})^{n-k}]\}$$

which leads to

$$(\overline{\boldsymbol{\omega}}_{\boldsymbol{N}}^{*})(0)\cdot\overline{\boldsymbol{e}}=\frac{1}{D}\cdot\overline{\boldsymbol{p}}_{\boldsymbol{0}}\cdot\overline{\boldsymbol{E[V]}}-\sum_{n=0}^{N-1}(\overline{\boldsymbol{\omega}}_{\boldsymbol{n}}^{*})(0)\cdot\overline{\boldsymbol{e}}\tag{42}$$

In the same way, for $n=1,\cdots,N-1$

$$(\overline{\boldsymbol{\pi}}_{\boldsymbol{n}}^{*})_i(\theta)=\rho_i\cdot\{\sum_{k=1}^{n}\frac{(\overline{\boldsymbol{p}}_{\boldsymbol{k}})_i}{\sum_{l=1}^{N}(\overline{\boldsymbol{p}}_{\boldsymbol{l}})_i}\cdot\frac{1}{(\lambda-\theta)S_i}\cdot[S_i^*(\vartheta)\cdot(\frac{\lambda}{\lambda-\theta})^{n-k}-\sum_{l=0}^{n-k}g_{l,i}\cdot(\frac{\lambda}{\lambda-\theta})^{n-k-l}]\},$$
$$=\frac{1}{(\lambda-\theta)D}\cdot\{\sum_{k=1}^{n}(\overline{\boldsymbol{p}}_{\boldsymbol{k}})_j\cdot[S_i^*(\vartheta)\cdot(\frac{\lambda}{\lambda-\theta})^{n-k}-\sum_{l=0}^{n-k}g_{l,i}\cdot(\frac{\lambda}{\lambda-\theta})^{n-k-l}\}$$

and hence

$$(\overline{\boldsymbol{\pi}}_{\boldsymbol{n}}^{*})(\theta)\cdot\overline{\boldsymbol{e}}=\frac{1}{(\lambda-\theta)D}\cdot\{\sum_{k=1}^{n}\overline{\boldsymbol{p}}_{\boldsymbol{k}}\cdot\overline{\boldsymbol{S}}^*(\vartheta)\cdot(\frac{\lambda}{\lambda-\theta})^{n-k}-\sum_{k=1}^{n}\sum_{l=1}^{k}\overline{\boldsymbol{p}}_{\boldsymbol{l}}\cdot\boldsymbol{A}_{\boldsymbol{k-l}}\cdot\overline{\boldsymbol{e}}\cdot(\frac{\lambda}{\lambda-\theta})^{n-k}\}$$
$$=\frac{1}{(\lambda-\theta)D}\cdot\{\sum_{k=1}^{n}\overline{\boldsymbol{p}}_{\boldsymbol{k}}\cdot\overline{\boldsymbol{S}}^*(\vartheta)\cdot(\frac{\lambda}{\lambda-\theta})^{n-k}-\sum_{k=1}^{n}(\overline{\boldsymbol{p}}_{\boldsymbol{k-1}}-\overline{\boldsymbol{p}}_{\boldsymbol{0}}\cdot\boldsymbol{B}_{\boldsymbol{k-1}})\cdot\overline{\boldsymbol{e}}\cdot\left(\frac{\lambda}{\lambda-\theta}\right)^{n-k}\right\}\tag{43}$$

The probability of having a system occupancy equal to n at an arbitrary instant of a service interval is given by

$$(\overline{\boldsymbol{\pi}}_{\boldsymbol{n}}^{*})(0)\cdot\overline{\boldsymbol{e}}=\frac{1}{\lambda\cdot D}\cdot\left[\overline{\boldsymbol{p}}_{\boldsymbol{n}}\cdot\overline{\boldsymbol{e}}-\overline{\boldsymbol{p}}_{\boldsymbol{0}}\cdot\overline{\boldsymbol{e}}+\sum_{k=0}^{n-1}\overline{\boldsymbol{p}}_{\boldsymbol{0}}\cdot\boldsymbol{B}_{\boldsymbol{k}}\cdot\overline{\boldsymbol{e}}\right] \tag{44}$$

For n=N, we obtain

$$(\overline{\boldsymbol{\pi}}_{\boldsymbol{N}}^{*})(0)\cdot\overline{\boldsymbol{e}}=\frac{1}{D}\cdot\sum_{n=1}^{N-1}\overline{\boldsymbol{p}}_{\boldsymbol{k}}\cdot\overline{\boldsymbol{E}[\boldsymbol{S}]}-\sum_{n=0}^{N-1}(\overline{\boldsymbol{\pi}}_{\boldsymbol{n}}^{*})(0)\cdot\overline{\boldsymbol{e}} \tag{45}$$

Appendix 2

Since

$$\overline{\boldsymbol{W}}^{*}(\theta)=\frac{1}{1-P_b}[\sum_{n=1}^{N-1}\overline{\boldsymbol{\pi}}_{\boldsymbol{n}}^{*}(\theta)\cdot Diag(\prod\nolimits_{k=1}^{n}S_{F^{k}(1)}^{*}(\theta),\cdots,\prod\nolimits_{k=1}^{n}S_{F^{k}(E_{max})}^{*}(\theta))$$
$$+\sum_{n=0}^{N-1}\overline{\boldsymbol{\omega}}_{\boldsymbol{n}}^{*}(\theta)\cdot Diag(\prod\nolimits_{k=1}^{n+1}S_{F^{k}(G(1))}^{*}(\theta),\cdots,\prod\nolimits_{k=1}^{n+1}S_{F^{k}(G(E_{\max}))}^{*}(\theta))]$$

by taking the derivative

$$E[W]=-\frac{d\overline{\boldsymbol{W}}^{*}(\theta)}{d\theta}|_{\theta=0}\cdot\overline{\boldsymbol{e}}$$

we obtain

$$E[W]=\frac{1}{1-P_b}\{-\sum_{n=1}^{N-1}\frac{d\overline{\boldsymbol{\pi}}_{\boldsymbol{n}}^{*}(\theta)}{d\theta}|_{\theta=0}\cdot\overline{\boldsymbol{e}}-\sum_{n=0}^{N-1}\frac{d\overline{\boldsymbol{\omega}}_{\boldsymbol{n}}^{*}(\theta)}{d\theta}|_{\theta=0}\cdot\overline{\boldsymbol{e}}$$
$$+\sum_{n=1}^{N-1}\overline{\boldsymbol{\pi}}_{\boldsymbol{n}}^{*}(0)\cdot(\sum\nolimits_{l=1}^{n}E[S_{F^{l}(1)}],\cdots,\sum\nolimits_{l=1}^{n}E[S_{F^{l}(E_{max})}])'$$
$$+\sum_{n=0}^{N-1}\overline{\boldsymbol{\omega}}_{\boldsymbol{n}}^{*}(0)\cdot(\sum\nolimits_{l=1}^{n+1}E\left[S_{F^{l}(G(1))}\right],\cdots,\sum\nolimits_{l=1}^{n+1}E[S_{F^{l}(G(E_{max}))}])']\} \tag{46}$$

Using the results for $(\overline{\boldsymbol{\pi}}_{\boldsymbol{n}}^{*})(\theta)\cdot\overline{\boldsymbol{e}}$ and $(\overline{\boldsymbol{\omega}}_{\boldsymbol{n}}^{*})(\theta)\cdot\overline{\boldsymbol{e}}$ in (41) and (44) and the fact that

$$\sum_{n=1}^{N-1}\overline{\boldsymbol{\pi}}_{\boldsymbol{n}}^{*}(0)\cdot(\sum\nolimits_{l=1}^{n}E[S_{F^{l}(1)}],\cdots,\sum\nolimits_{l=1}^{n}E[S_{F^{l}(E_{max})}])'$$
$$=\sum_{n=1}^{N-1}\overline{\boldsymbol{\pi}}_{\boldsymbol{n}}^{*}(0)\cdot(n\cdot E[S_1],\cdots,n\cdot E[S_{E_{max}}])'$$

and

$$\sum_{n=0}^{N-1}\overline{\boldsymbol{\omega}}_{\boldsymbol{n}}^{*}(0)\cdot(\sum\nolimits_{l=1}^{n+1}E\left[S_{F^{l}(G(1))}\right],\cdots,\sum\nolimits_{l=1}^{n+1}E[S_{F^{l}(G(E_{max}))}])'$$
$$=\sum_{n=0}^{N-1}\overline{\boldsymbol{\omega}}_{\boldsymbol{n}}^{*}(0)\cdot((n+1)\cdot E[S_1],\cdots,(n+1)\cdot E[S_{E_{max}}])'$$

we obtain Little's well known result

$$E[W] = \frac{1}{1 - P_b} \cdot \frac{1}{\lambda} \cdot E[\eta] \tag{47}$$

References

[1] F. K. Shaikh and S. Zeadally (2016), Energy harvesting in wireless sensor networks: A comprehensive review*, Renewable and Sustainable Energy Reviews*, 55, 1041 – 1054.

[2] I. Ahmed, M.M. Butt, C. Psomas, A. Mohammed, I. Krikidis and M. Guizani (2015), Survey on energy harvesting wireless communications: Challenges and opportunities for radio resource allocation*, Computer Networks, 88*, 234-248.

[3] C. Delgado, J.M. Sanz, C. Blondia, J. Famaey (2020), Battery-Less LoRaWAN Communications using Energy Harvesting: Modeling and Characterization, to appear in *IEEE Internet of Things Journal.*

[4] K. Patil and D. Fiems (2018), The value of information in energy harvesting sensor networks*, Operations Research Letters, 46(3)*, 362-366.

[5] A. Mouapi and N. Hakem (2016), Performance evaluation of wireless sensor node powered by RF energy harvesting*, 16th Mediterranean Microwave Symposium (MMS), Abu Dhabi*, 1-4.

[6] S. Kosunalp, "*MAC protocols for energy harvesting wireless sensor networks: a survey",* ETRI Journal Voll. 37 (4), PP. 804-812, 2015

[7] H.H.R. Sherazi, L. A. Grieco and G. Boggia (2018), A comprehensive review on energy harvesting MAC protocols in WSNs: Challenges and Tradeoffs*, Ad Hoc Networks, 17*, 117-134.

[8] M.A. Mitici, J. Goseling, M. de Graaf and J. Boucherie (2016), Energy-efficient data collection in wireless sensor networks with time constraints*, Performance Evaluation, 102*, 34-52.

[9] Li J., Zhou H.Y., Zuo D.C., Hou K.M., Xie H.P. (2014), Zhou P. Energy consumption evaluation for wireless sensor network nodes based on queuing Petri net*, Int. J. Distrib. Sens. Netw.,* 11 pp.

[10] Ke J.-F., Chen W.-J., Huang D.-C (2015), A life extend approach based on priority queue N strategy for wireless sensor network; *Proceedings of the 11th EAI International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QSHINE)*, Taipei, Taiwan. 19–20 August 2015, 272–279.

[11] Ghosh S., Unnikrishnan S. (2017), Reduced power consumption in wireless sensor networks using queue-based approach; *Proceedings of the 5th IEEE International Conference on Advances in Computing, Communication and Control (ICAC3)*; Mumbai, India. 1–2 December 2017

[12] W.M. Kempa (2019), Analytical model of a wireless sensor network (WSN) node operation with a modified threshold-type energy saving mechanism, *Sensors, 19(14) : 3114*

[13] Y-H. Chen, B. Ng, W.K.G. Seah, A-C. Pang (2016), Modeling and analysis: energy harvesting in the Internet of Things, *Proc. 19th International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems,* 156-165

[14] E. Cuypere, K. De Turck, D. Fiems (2018), A queueing model of an energy harvesting sensor node with data buffering, *Telecommunication Systems,67,* 281-295

[15] A. Seyedi, B. Sikdar (2008), Modeling and analysis of Energy Harvesting Nodes in Wireless Sensor Networks, *2008 46th Annual Allerton Conference on Communication, Control, and Computing*, Urbana-Champaign, IL

[16] J.S. Jornet, I.F. Akyildiz (2012), Joint Energy Harvesting and Communication Analysis for Perpetual Wireless Nanosensor Networks in the Terahertz Band, *IEEE Transactions on nanotechnology 11*, 570–580.

[17] P.-J. Courtois (1980), The M/G/1 finite capacity queue with delays, *IEEE Trans. Comm., COM-28(2)*, 165-172.

[18] T.T. Lee (1984), M/G/1/N queue with vacation time and exhaustive service discipline, *Operations Research*, 32(4), 774-784.

[19] A. Frey and Y. Takahashi (1998), An explicit solution for an M/GI/1/N queue with vacation time and exhaustive service discipline, *Journal of the Operations Research Society of Japan, 41(3)*, 95-100.

[20] C. Blondia (1989), A finite capacity multi-queueing system with priorities and with repeated server vacations, *Queueing Systems, 5(4)*, 313-330.

[21] C. Blondia (1991), Finite Capacity Vacation Models with Non-Renewal Input, *J. Appl. Prob.*, 28, 174-197.

[22] S. R. Mahabhashyam and N. Gautam (2005), On queues with Markov modulated service rates, *Queueing Systems, 51*, 89-113.

[23] G. Horvath, Z. Saffer, M. Telek (2017), Queue length analysis of a markov-modulated vacation queue with dependent arrival and service processes and exhaustive service policy, *Journal of Industrial and Management Optimization, 13 (3)*, 1365-1381.

[24] Abhishek, M.A.A. Boon, O.J. Boxma, R. Nunez-Queija (2017), A single-server queue with batch arrivals and semi-Markov services, *Queueing Systems, 86*, 217-240.