



Advanced Research in Real-Time Scheduling for Energy Autonomous Sensor Nodes

M Chetto, M Elghor, H Elghor

► To cite this version:

M Chetto, M Elghor, H Elghor. Advanced Research in Real-Time Scheduling for Energy Autonomous Sensor Nodes. 10th International Conference on Sensors Engineering and Electronics Instrumentation Advances (SEIA' 2024), Sep 2024, Ibiza, Spain. hal-04700764

HAL Id: hal-04700764

<https://hal.science/hal-04700764v1>

Submitted on 17 Sep 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Advanced Research in Real-Time Scheduling for Energy Autonomous Sensor Nodes

M. Chetto ¹, M. Elghor ¹ and H. Elghor ²

¹ LS2N Laboratory, Nantes University, Nantes, France

² LENS Laboratory, Lebanese University, Saida, Lebanon

Tel.: + 33 228092109, e-mail: maryline.chetto@ls2n.fr

Summary: Energy-aware scheduling policies must be integrated into a sophisticated design process in order to design Energy Harvesting (EH) sensor nodes with real-time capabilities. These systems should achieve energy neutrality by efficiently managing energy resources, ensuring continuous operation even in situations with intermittent energy availability while respecting timeliness constraints. An overview of the ineffectiveness of classical real-time scheduling techniques is given. This motivates more advanced research studies to propose and validate new energy harvesting aware schedulers. The paper presents EED-H, a scheduler which efficiently specifies how processing time and energy should be allocated to the software tasks in order to ensure energy neutrality whenever feasible. The paper finally draws attention to a few technical issues with the scheduler's actual implementation, mainly in relation to measuring energy production, energy consumption and energy prediction.

Keywords: Autonomous sensor, real-time computing, energy harvesting, energy neutrality, dynamic priority scheduling.

1. Introduction

The Internet of Things (IoT) and sensor nodes are two interconnected technologies that are essential to the creation of contemporary smart systems. A sensor node in the context of the IoT refers to a device equipped with sensors that can continuously collect data from the environment and transmit it in real-time over a network, typically the internet, for analysis, monitoring, or control purposes. Sensor nodes have uses in many different fields, such as industry, transportation, agriculture, smart homes, and healthcare. Computational and communication components are integrated with physical elements [1]. Since bound responses to changes in the physical environment are required, this integration must enable real-time monitoring.

1.1. Energy Harvesting for sensor nodes

To extend battery life, the majority of sensor nodes are made to run at low power. Energy Harvesting (EH) offers a safer option in situations where maintenance or battery replacement could be dangerous. The process of gathering and storing environmental energy to power electronic devices is referred to as this technology. Energy harvesting encourages sustainability and has the potential to save money over time by lowering dependency on external power sources and minimizing the need for battery

replacement. EH sensor nodes powered by energy harvesting can be installed anywhere maintenance is expensive or difficult to access (see Fig.1.).

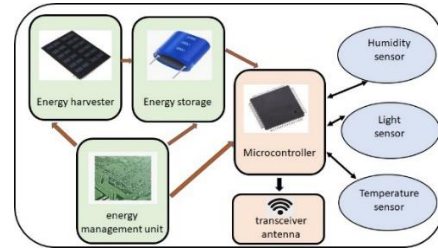


Fig. 1. Framework of an EH sensor node

It is possible to use a variety of renewable sources, including ambient RF signals, vibration, thermal gradients, and sunlight. The particular environment influences the choice of energy source. For instance, in a factory, vibration-based machinery harvesting might be feasible, but in an outdoor environment, solar energy might be more useful. Other examples are air and water quality monitoring, wildlife tracking, disaster detection. Transducers are used to transform energy that is captured from the environment such as light, motion, and heat into electrical energy [2]. After that, this energy is kept for later use in batteries, supercapacitors, or capacitors.

1.2 Challenges

In order to develop effective and dependable systems, engineers and researchers must overcome the challenges posed by EH sensor nodes. A multidisciplinary strategy involving knowledge of mechanical, electrical, computer, and system integration engineering is needed to meet these challenges. Power management techniques should be specifically included for EH small electronic devices in order to make them energy neutral, or to dynamically modify the activity of the main component (processor) in accordance with energy levels that are available. The system is said to be energy-neutral when the energy extracted from the environment equals or surpasses the energy used by the system. Stated differently, the system can operate continuously without requiring external power and without exhausting its energy supply.

The processing unit is the brain of the sensor node, responsible for processing the data collected by the sensors from their environment by running tasks at regular intervals (e.g., every second, every minute). These tasks may preprocess data to filter noise, aggregate information, or perform initial analysis to reduce the amount of data transmitted to other nodes or a central hub. Such tasks are said to be real-time since they should be executed timely before specified deadlines. An RTOS (Real Time Operating System) is used to manage them efficiently through a specific scheduling algorithm. The challenge in a so-called RTEH (Real Time Energy Harvesting) system is to design a smart energy aware scheduler that can effectively balance the need for timely task execution with the constraints of limited and variable energy availability

1.3 Contributions of the paper

The paper's primary goal is to highlight the key problems with creating an RTEH sensor node. The second goal is to present EED-H, a new scheduler that advances the state of the art. The remainder of the paper's sections are arranged as follows. A quick overview of scheduling issues for RTEH systems is done in Section 2. The details of the new energy harvesting aware scheduler (EED-H) for uniprocessing nodes are provided in Section 3. Section 4 presents a discussion on the applicability of this scheduler, and Section 5 concludes.

2. Background materials

2.1. Classical real-time scheduling

Most of the research on real-time scheduling has focused on models where the tasks have to be completed before a deadline and the processing requirements are only expressed in terms of Worst Case Execution times (WCET). Since Liu and Layland's seminal work five decades ago, many results have been obtained for both fixed-priority and dynamic-priority task scheduling under this

assumption. The Earliest Deadline First (EDF) scheduling algorithm is an optimal dynamic priority scheduling method. In EDF, the priority of a job is determined by its deadline: the job with closest deadline is given the highest priority [4].

2.2. Energy harvesting aware scheduling

As seen in the example on Fig. 2., the famous real-time scheduler EDF is ineffective in energy autonomous devices with intermittent energy supply. This is a result of its work-conserving nature which ensures that jobs are completed as soon as possible and never leaves the processor idle while there are still jobs waiting to be completed. By looking at the possible outcomes of job execution that involve time and energy consumption, we should be able to comprehend the specific architecture of an efficient energy harvesting aware scheduling algorithm.

Let's look more closely at the detrimental effects of energy variability using two jobs as an example (see Fig. 2.). The deadlines and release times for each job are represented by the upper and lower arrows. $E(t)$ is the stored energy at any given time t .

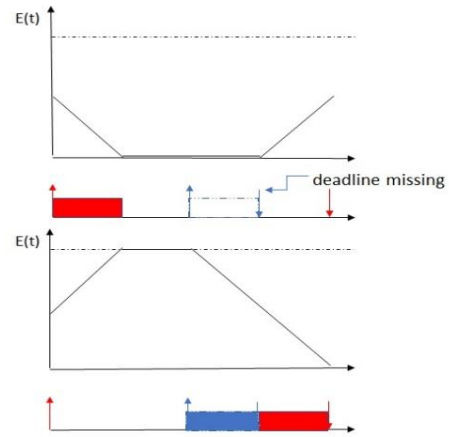


Fig. 2. Classical v.s. energy aware real-time scheduling

In order to determine whether to keep the processor busy or idle at a given time instant, we need to achieve the online prediction of environmental energy production in the near future and the online monitoring of the energy available in the storage unit. Let us examine in more detail the undesirable effect of energy variability. A job with deadline at time 4 executes from time 0 to time 1 and consumes energy. As a result, when another job releases at time 2, there is no more sufficient energy to complete it by its deadline at time 3 whatever its priority. Our intuition might tell us that, we should consider future arrivals of jobs as well as their energy requirements to make dispatching decision i.e. to decide whether or not putting the processor in the sleep mode. The second Gantt chart in Fig. 2. illustrates this fact. A good scheduler should prevent from energy starvation whenever possible. Consequently, we have to examine not only the energy that is currently stored in the storage unit at every time

instant but also the energy that will be produced by the environmental source and consumed by jobs in future.

In 2014, we thus proved that optimality requires clairvoyance on the near future and more precisely lookahead-D, where D is the application's longest relative job deadline. Put another way, any optimal scheduler that makes decisions during runtime must predict the pattern of energy inflow and task arrival over at least D time units [3].

3. EED-H: a new energy aware scheduler

3.1. Assumptions

We consider a sensor node that is equipped with a single processor with constant frequency, an energy harvester and an energy storage unit (see Fig. 3.). The application software is a set of independent periodic tasks represented by their worst case execution time, worst case energy consumption, deadline and period. Thus, each task generates an infinite set of deadline constrained jobs. The tasks can be categorized in two types: discharging and recharging. By definition, the energy level in the storage unit decreases whenever a discharging task is running whereas a recharging task contributes to recharge the energy storage unit (i.e. energy consumed is lower than energy produced by the source).

The energy storage unit is typically a battery or a capacitor. The maximum capacity of the energy storage unit is denoted by C. The energy level of the energy storage unit at time t is denoted as E(t). The energy harvesting unit can define its variable power output as a function of time, say $P_p(t)$.

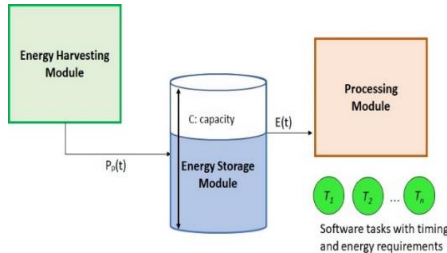


Fig. 3. Description of the model

3.2. Main concepts

An addition to classical EDF which orders the jobs according to deadlines, the EED-H scheduler provides a dynamic power management strategy sometimes known as dynamic duty cycling. In that objective, the main concepts behind EED-H are the so-called *slack time* and *preemption slack energy*:

- The longest period of time, beginning at current time t, during which the processor may remain idle without causing deadline violations is known as the slack time at t.

- The maximum amount of energy that can be consumed by the highest priority job at current time t without causing future energy starvation is known as the preemption slack energy. Stated differently, if at any given time the preemption slack energy is zero the processor has to immediately enter the idle state so as to avoid energy starvation and consequently deadline missing in future. Obviously, calculation of the preemption slack energy needs the knowledge of energy consumed and energy produced in near future. Mathematical formulae for calculation of slack time and slack energy can be found in [3].

3.3 Pseudo-code of EED-H

Energy availability, preemption slack energy and slack time at current time t are respectively denoted by E(t), PSE(t) and ST(t) in the pseudo-code reported on Fig. 4.

Algorithm 1 Extended Earliest Deadline First with Harvesting (EED-H) Scheduler

Input: $L(t)$ = list of ready jobs at time t in the EDF order
 $L_r(t)$ = list of ready recharging jobs at time t in the EDF order
 $L_d(t)$ = list of ready discharging jobs at time t in the EDF order

Output: Processor management in $[t, t + 1)$.

```

1: while (1) do
2:   if  $L(t) = \phi \parallel PSE(t) = 0 \parallel (E(t) = 0 \ \&\& \ L_r(t) = \phi)$  then
3:     Put the processor idle in  $[t, t + 1)$ 
4:   else
5:     if  $L_d(t) \neq \phi \ \&\& \ E(t) = C$  then
6:       Execute the first job of  $L_d(t)$  in  $[t, t + 1)$ 
7:     else
8:       if  $L_r(t) \neq \phi \ \&\& \ E(t) = 0$  then
9:         Execute the first job of  $L_r(t)$  in  $[t, t + 1)$ 
10:      else
11:        if  $ST(t) = 0$  then
12:          Execute the first job of  $L(t)$  in  $[t, t + 1)$ 
13:        else
14:          Put the processor idle or execute the first job of  $L(t)$  indifferently
15:        end if
16:      end if
17:    end if
18:  end if
19: end while

```

Fig. 4. EED-H: a new energy harvesting aware real-time scheduler

The algorithm operates within a continuous loop, handling, at each time t, one of the following situations:

- there is no job ready for execution, or the preemption slack energy (PSE) is equal to zero, or the level in the energy reservoir is zero while there is no recharging job ready for execution: the processor has to be put into the idle state imperatively. Then, the energy harvester produces energy which serves to recharge the storage unit (line 2).

- there is at least one discharging job ready for execution and the energy storage unit is fully replenished: the processor has to execute the highest priority discharging job so as to discharge the energy unit, wasting no energy and optimizing processor utilization (line 5).
- the level in the energy storage unit is zero and there is at least one ready recharging job: the processor has to execute the highest priority recharging job so as to recharge the energy unit and optimize processor utilization (line 8).
- Whenever the slack time is zero, the processor has to be active imperatively and execute the highest priority job so as to avoid deadline missing (line 11).
- In any other situation, indifferently the processor may be idle or active executing the highest priority job (line 14).

5. Implementation considerations

Accurate estimations of the energy generated by the source and accurate assessment of the energy consumed by the tasks are the two simplified assumptions of the EED-H scheduling framework. Estimating the worst-case energy consumption of any task can be done simply when the only impact of the processor is taken into consideration. However, the amount of energy used along task execution can depend on other energy users, such as, memory, timers, transceivers, etc.

The majority of studies found no significance in the processor's energy consumption when it was idle. Actually, processor's energy consumption while it is idle depends on a variety of factors. The scheduling algorithm EED-H regarded the processor's idle state energy consumption as insignificant. Processor design must make an effort to minimize idle power consumption that depend on a number of variables, such as the processor's unique architecture and the amount of energy used by the parts that are kept active while the machine is idle. Maintaining a security stock of energy in the energy storage to guarantee the survival of the system in any situation—that is, regardless of the processor's state—is an efficient way to account for this phenomenon.

The most challenging when designing an EH platform comes from the possible highly unstable nature of the energy source. Various environmental sources can be employed, such as sunlight, vibration, temperature differences, etc. Sunlight is certainly the most frequently used since it produces sufficient energy to supply small electronic devices such as wireless sensor nodes. Unstability of sunlight originates from varying weather conditions, seasons, etc. Prediction schemes can be found in literature, including Exponentially Weighted Moving Average (EWMA), Weather Conditioned Moving Average (WCMA), leveraging weather forecasts [6] [7]. They

provide accurate solutions with time horizons of the order of minute.

6. Conclusion and future works

The most promising technology to extend maintenance-free lifetime of wireless sensors is energy harvesting i.e. the use of ambient energy sources. For over twenty years, EH has been the focus of extensive research. But there are still a lot of unanswered research questions. In this paper, we addressed the real-time scheduling one where timeliness requirements are expressed in terms of deadlines.

The paper introduced EED-H, a new dynamic priority based scheduler that cleverly calculates the busy and sleep periods of the processor to provide energy neutrality. From a short-term viewpoint, EED-H efficiently specifies how processing time and energy should be allocated to the real-time software tasks in order to ensure energy neutrality. Simulations are currently achieved to put in light the performance gain compared to the traditional scheduler EDF. A theoretical study has proved that EED-H allows for energy-neutral operation with no energy wastage, no energy starvation, and no missed deadlines whenever possible which makes it an optimal scheduler.

As the demand for sustainable and autonomous sensor nodes grows, the EED-H scheduler is an essential step toward providing energy harvesting aware RTOSes.

References

- [1]. Ma, D.; Lan, G.; Hassan, M.; Hu, W.; Das, S. K. Sensing, Computing, and Communications for Energy Harvesting IoTs: A Survey. *IEEE Communications Surveys and Tutorials*, 22(2), 1222-1250, 2020.
- [2]. Newell, D.; Du_y, M. Review of Power Conversion and Energy Management for Low-Power, Low-Voltage Energy Harvesting Powered Wireless Sensors. *IEEE Transactions on Power Electronics*, 34(10), 9794-9805, 2019.
- [3]. Chetto, M. Optimal Scheduling for Real-Time Jobs in Energy Harvesting Computing Systems. *IEEE Transactions on Emerging Topics in Computing*, 2(2), 122-133, 2014.
- [4]. Liu, C.I.; Layland, J.w. Scheduling algorithms for multiprogramming in a hard real-time environment. *Journal of the Association for Computing Machinery*, 20(1), 46-61, 1973.
- [5]. Chetto, M.; El Ghor, H.; Scheduling and power management in energy harvesting computing systems with real-time constraints, *Journal of Systems Architecture*, 98, Pages 243-248, 2019.
- [6]. N. Sharma, J. Gummeson, D. Irwin, and P. Shenoy, Cloudy computing: Leveraging weather forecasts in energy harvesting sensor systems, in *Proceedings of 7th Annual IEEE Communications Society Conference on Sensor Mesh and Ad Hoc Communications and Networks*, pp. 1–9, 2010.
- [7]. Y. Li, Z. Jia and X. Li, Task Scheduling Based on Weather Forecast in Energy Harvesting Sensor Systems, *IEEE Sensors Journal*, 14 (11), 2014.