Journal of Computer Science

Original Research Paper

Task Scheduling in Real Time Systems with Energy Harvesting and Energy Minimization

Arvind Kumar and Bashir Alam

Department of Computer Engineering, Faculty of Engineering and Technology, Jamia Millia Islamia, New Delhi, India

Article history
Received: 11-04-2018
Revised: 2-07-2018
Accepted: 31-07-2018

Corresponding Author:
Bashir Alam
Department of Computer Engineering, Faculty of Engineering and Technology, Jamia Millia Islamia, New Delhi, India
Email: babashiralam@gmail.com

Abstract: Energy is a major concern in battery operated real time devices (e.g. sensor nodes situated in remote areas) as battery life is solely responsible for the functioning of such devices. For such systems to work continuously and successively a regular source of energy is required. The lifetime of such systems can be increased by reducing the energy consumption and increasing energy production. Dynamic Voltage Scaling (DVS) scheduling algorithms is an existing approach for minimizing energy consumption. Energy harvesting is used for energy production (as a source of energy) and Earliest Deadline First (EDF) is an existing traditional approach for task scheduling in real time systems. In this paper, we propose a real time task scheduling algorithm using both DVS and EDF with energy harvesting. The proposed approach is designed to reduce energy consumption and enhanced the system lifetime. Our experimental results show that the proposed approach increases the task feasibility and reduces the energy consumption to increase the system lifetime.

Keywords: Real Time System, Energy, Scheduling, Energy Harvesting, Feasibility

Introduction

As the technology trend is changing and enhancing very fast, real time systems are being used very extensively in a large number of systems. These systems also have been used in the daily life of human such as automatic vehicle driver, automatic control system, navigation system, reservation system, medical fields, weather forecasting system, etc. Many real time devices which are situated in the remote area where charging a battery is not possible through usual resources, energy harvesting is required. Energy harvesting is an approach that allows capturing the energy from natural resources. (Moser *et al.*, 2007). An unused ambient energy is converted into electrical energy which can be directly used by the system or stored in batteries for further use. Using this approach we can increase the system's lifetime (Jejurikar and Gupta, 2002). In energy harvesting real time systems, the main issue is to guarantee that the system does not consume more energy than available and harvested energy at a particular moment (Abdeddaïm *et al.*, 2016). There are several energy harvesting approaches such as solar power (sunlight), wind power, ocean waves, etc. Energy harvesting using solar power is a perfect match for real time devices.

Figure 1 shows the energy harvesting model which consist of real time device, real time task set, energy harvester and external energy storage. Energy is being harvested from sunlight and passing to the energy storage device or system directly. A system can perform normal functioning if it is getting the required energy directly from the sun. It performs proper functioning by using a battery as an energy resource (Linjie *et al.* 2012). The system stops working if the battery gets discharged. Dynamic Voltage Scaling (DVS) is the well-known and most effective approach for energy management in real time systems. Energy consumption can be reduced using DVS by scaling the voltage level on the basis of the energy requirement of the system to execute a task (Kim and Kim, 2007). DVS gives the best performance when the energy requirement of a system is less than the available energy, the supply voltage can dynamically be reduced to the lowest possible requirement of energy to operate the system (Lin *et al.*, 2016; Housseyni *et al*, 2016).

Science Publications

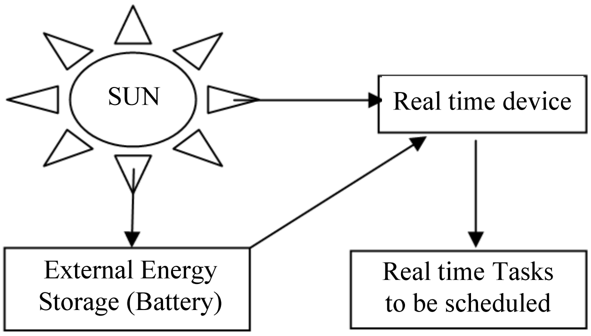
SUN
Real time device
External Energy Storage (Battery)
Real time Tasks to be scheduled

Fig. 1: Energy Harvesting Model (Chetto and Queudet, 2014)

The main issue in today's scenario is to enhance the lifetime of a real time device. Many approaches have been designed to increase the system lifetime using various energy minimization approaches (like DVS) and energy harvesting. The proposed algorithm focuses on enhancing lifetime of the real time system using energy harvesting as well as proper task scheduling using DVS. This scheduling provides an enormous amount of task execution in less span of time as compared to existing algorithms for the same.

In this paper, the aim is to design an algorithm for increasing the lifetime of a battery operated energy harvesting real time system. We used the concept of energy harvesting for ensuring energy availability of a system. DVS is used for reducing energy consumption. The concept of EDF is also used for task scheduling. In EDF task scheduling algorithm, tasks are scheduled on the basis of task deadline. We designed an algorithm for minimizing power consumption in energy harvesting real time systems that results in the increase in its lifetime.

Related Work

For past many years, most of the research in real time energy harvesting system focuses on either minimizing energy consumption or enhancing lifetime of a system (Kwon and Kim. 2005). Feasibility of task in real time systems depends on its deadline as timing constraint (Qin and Xie, 2008). Unavailability of energy in these real time systems (working in remote areas) is a big issue. A lot of research has been done to minimize the energy consumption. Dynamic Power Management (DPM) and Dynamic Voltage and Frequency Scaling (DVFS) has been investigated for minimizing energy consumption in battery operated system (Liu *et al*, 2006; 2008). These approaches work efficiently with available energy resources. In battery operated systems, a battery has its lifetime. It can't work after discharge. Batteries need to be charged or replaced for continuing the functioning. Replacement of battery is not possible for those systems which are working in remote areas (such as mountains, forests, etc). It is very costly and impractical to charge or replace batteries in such systems.

Energy harvesting approach has been developed to overcome this problem (Rusu *et al*., 2004; Guohui *et al*, 2009; Ghor *et al*., 2011). The lifetime of such real time system can be increased using energy harvesting and the limit of battery charging is also removed. Energy harvesting from sunlight using Photovoltaic (PV) cell (solar system) can provide uninterrupted power supply. Energy harvesting from sunlight have various characteristics such as the output of the solar panel varies in the morning, noon and evening time. The magnitude of energy is very high at noon as compared to morning and evening time. The outputs of such systems also depend upon weather conditions. Good weather condition leads to maximize energy output. To continue operating properly in bad weather condition, morning or evening time, or even in the night, energy storage element such as a rechargeable battery is required (Chen*et al.,* 2014; Shashank and Daniel, 2009). Many researchers also have presented their work for energy harvesting in real time systems which work differently from DVFS/DPM. The task scheduling in such systems is also based on EDF (Chetto and Queudet, 2014).

In earlier research (Francesco Quaglia *at al*., 2001), it was considered that energy is not required for scheduling jobs in single processor computing systems. In EDF, the feasibility of each task depends on its deadline. A task which has a minimum deadline at that instant is scheduled first. Energy harvesting EDF scheduling algorithms also have been presented for an aperiodic task, which does not consider future energy requirement, energy production and task arrival (Zhang and Chakrabarty, 2006). Energy shortage should imperatively be anticipated to avoid missing the deadline of tasks. ED-H, a semi-online scheduling algorithm has been presented by Chetto (2011) for optimal scheduling in the energy harvesting computing system in which feasibility test is applied to check optimality (Lu and Qiu, 2011).

System Models and Terminology

This section presents the models to formalize energy harvesting real time systems under energy and timing constraint. Application model, task model, and energy model are included in the system model. Some basic terminologies used in this paper are also discussed in this section.

Application Model

We consider real time critical applications connected to an energy harvesting system. Real time critical systems contain various tasks such as periodic task, sporadic task, or aperiodic task. Here we consider the aperiodic task which contains its arrival time, execution time and deadline. An energy harvesting device collects energy from its environment at some rate. That energy is stored in an energy storage device for future use by real time system.

Task Model

We considered a set T = {T1, T2, …, Tn} of n independent, non-preemptive aperiodic tasks scheduled on the uni-processor system. Each task Ti is a collection of 4 tuples <*ai*, *ci*, *di*, *Ei*>, where *ai* represents arrival time, *ci* represents execution time, di is a deadline of a task and Ei is energy consumption. We considered a single processing unit to schedule the real time task that supports only operating voltage. Its energy consumption is only at the time of operation performing on a task. Another parameter in real time system is utilization U_i. The utilization of the task can be defined as $U_i = ci/di$ such that $0 \le U_i \le 1$. The total utilization of task set can be defined as the sum of all task utilizations and it can be expressed as:

$$Total\,Utilization U = \sum_{i=1}^{n} U_i$$

The utilization of task and total utilization of n task set should be less than 1. When the utilization U is closed to "1" then the execution of task can't be postponed by the processor.

Energy Model

Energy model can be expressed in three different ways such as energy production model, energy consumption model, and energy storage model (Yasmina and Masson, 2012).

Energy Production Model

The production of energy is not controllable from source (sunlight). The energy can be produced by an instantaneous charging rate that incorporates all losses. The energy production from a source between time interval [*t*1 and *t*2] can be expressed as follows:

$$Ep(t1,t2) = \int_{t1}^{t2} Rp(t)\,\mathrm{dt}$$

where, *Ep*(*t*1, *t*2) = Energy produced between *t*1 and *t*2.

Rp(*t*) = Rate of energy production

The energy produced can be used directly by the system or it can be stored in a storage device for future use.

Energy Consumption Model

We assume that energy consumption of a real time device can be modeled as the sum of static energy consumption and dynamic energy consumption. Real time systems are DVS enabled which can perform at different frequency level. The frequency level of a system is directly proportional to supplied voltage. The frequency level is maximum (fmax) at maximum voltage (Vmax) and frequency level is minimum (fmin) at the minimum voltage (Vmin). Frequency level increases on increasing the voltage and decreases on decreasing the supplied voltage. The energy consumption *Ec*(*t*) at maximum voltage and frequency for a task *T* can be defined as below:

$$Ec(t) = Ec(stat) + h\{Ec(dep) + Ec(Indp)\}$$

where *Ec*(*stat*) is static energy consumption which can be saved by turning off the whole system, *Ec*(*dep*) and *Ec*(*Indp*) both are dynamic power consumption which is frequency dependent and independent. The coefficient *h*=1, *If* the task executes in the system otherwise *h=0.*

Energy Storage Model

Normally battery (energy storage device) is used for energy storage in a system. We also assume a battery (with maximum capacity EC), is used in the real time system. Harvested energy cannot be more than its capacity. Energy is wasted on continuous charging if the battery is fully charged at time t. A battery can be considered as fully charged if *Ep*(*t*1, *t*2) ≈ *EC*.

Basic Terminology

Some of the basic terminologies has been defined which are needful throughout this paper.

Definition 1

Time starvation: when a task misses its deadline before completing its execution at time t, a deadline violation occurs due to missing the deadline with sufficient available energy.

Definition 2

Energy Starvation: Execution of task is incomplete and misses the deadline due to unavailability of required energy, a deadline violation occurs due to insufficient available energy.

Definition 3

Valid Schedule: A schedule of a task in a task set T is valid if the deadline of the task met with available energy.

Definition 4

Feasible Schedule: Schedulability of a task is feasible if there exists a valid schedule with available energy otherwise it is infeasible.

Definition 5

Static Schedule: It can pre-compute the optimal schedule of a static task at design time.

Definition 6

Dynamic Schedule: It does not require any prior information of task at design time. It handles the scheduling of task at run time only. Aperiodic tasks are scheduled at run time.

Scheduling Policies

Static Scheduling

Task set is pre-defined and initially computed for checking the feasible schedule. A processor demand for the task set at starting the level. A predefined set of periodic tasks are generated between time interval *t*1 and *t*2. The total processor demand of n tasks can be calculated as the sum of the execution time of all tasks are expressed by the following equation:

$$pd(t1,t2) = \sum_{t_1}^{t_2} c_i \;|\{i=1,2,...,n\}$$

where, pd(*t*1, *t*2) = Processor Demand between *t*1 and *t*2 and c_i = Execution Time.

Feasibility of task can be checked by comparing processor demand with the deadline. Feasibility of a task Ti is guaranteed if and only if the processor demand (pd) of that task is less than its deadline and slack time of the task is greater than 0. For a task *Ti*, its deadline *di* and slack time *Si* are:

$$di = D - \sum_{j=i+1\, c_j}^{k}, 1 \le i \le k$$

$$Si = di - \sum_{j=i+1}^{k} c_j, 1 \le i \le k$$

where *D* is the duration of a frame in $1 \le i \le k$. we can easily say that necessary and sufficient condition for a task $Ti \in T$ to be feasible is $Si \ge 0$.

Dynamic Scheduling

In dynamic scheduling, the arrival of the task is not defined and the feasibility of the task is decided after its arrival at run time. If a task *Ti* arrives at *ai* with execution time *ci* and deadline *di*. Slack time *Si*(*d*) for dynamic task *Ti* at arrival time *ai* is computing by the following equation:

$$Si(d) = di - ai - ci - RTi$$

where *RTi* is total remaining execution time of higher priority tasks.

In real time energy harvesting systems, the feasibility of a task depends upon slack time and the slack time depends upon available energy. If the system has sufficient energy then it can be run at the maximum voltage which gives maximum slack time. We reduced the voltage level for saving energy to increase the lifetime of a system. The system also harvests energy if it is situated in remote areas where harvesting is only the source of energy (Allavena and Mosse, 2001).

Proposed Approach

Energy Harvesting Earliest Deadline First Scheduling Algorithm

EDF is well known and optimal scheduling algorithm based on earliest deadline first scheduling policy. Energy efficient EDF scheduling also has been proposed to enhance the task feasibility. The task is authorized to execute only if there is sufficient power supply otherwise the low priority task which requires less energy than available energy is scheduled first. The idea behind this paper is to harvest the energy in that interval and execute the awaiting task.

Theorem 1

There exists a valid schedule for a task if and only if Energy Demand (Edi) < Energy Available (EAi).

Proof

If Edi is energy demand for a particular task at a time instant esti (Execution start time) and E*Ai* is available energy at the same time instant, E*di* < E*Ai* directly shows that sufficient energy is available to execute the task and a valid schedule exist.

Theorem 2

If Task *Ti* missed the deadline due to Edi > EAi, there exist no schedule in EDF for a task set *T*.

Proof

If E*d* > E*Ai*, directly shows that energy demand for a task is greater than the available energy. The task will not schedule due to less amount of available energy. If the task is not schedulable then there exists no schedule for a task set T.

Energy Harvesting EDF Algorithm

Theorem 1 shows that there exists no schedule if available energy is less than the demand energy:

```
If $Ep(t1, t2)$ is energy produced between interval $t1$ and $t2$:
Then $EAi = EAi(t1) + Ep(t1, t2)$
    If $Edi < EAi$
        There exists a valid schedule for task $Ti$.
```

The energy demand of a task set T between the time interval *t*1 and *t*2 can be defined as:

$$Edi(t1,t2) = \sum_{t_1}^{t_2} E_i$$

where Ei is energy demand for a task Ti.

Available energy of a task set T between the time interval $t1$ and $t2$ is computing using the following:

$$EAi(t1, t2) = EC + Ep(t1, t2) - Edi(t1, t2)$$

where, $EAi(t1, t2)$ = Available energy after executing T tasks between time interval t1 and t2 and EC = the capacity of energy storage device

Algorithm 1

Static Energy Harvesting Earliest Deadline First Scheduling Algorithm
{Task set T = {T1, T2, …, Tn},
ReadyQueue = { }, est = execution start time, AD= Absolute deadline}

```
1.  ReadyQueue = empty
2.  Feasibility = TRUE
3.  At time t = $a_i$,
4.  ADi = $a_i$ + di, Ui = $\frac{c_i}{d_i}$
5.  If Ui ≤ 1
        Then ReadyQueue = ReadyQueue + {Ti}
6.  While ReadyQueue ≠ empty
7.  {
8.  $Select\ Ti \mid AD_i \leftarrow \min_{i=1 to n} AD_i$
9.  If WCET (esti + ci ) ≥ ADi THEN
            Feasibility = FALSE
10. Else if WCET (esti + ci ) ≤ ADi THEN
        If Edi ≥ EA THEN
            Feasibility = FALSE
        Else
            Schedule the Ti on empty core
            TAG core as Busy with task Ti
            ReadyQueue = ReadyQueue – {Ti}
11. }
12. Exit
```

In proposed static energy harvesting earliest deadline first (SHE-EDF) scheduling algorithm EDF is applied with energy harvesting to schedule the task set. The feasibility of tasks depends on two factors, absolute deadline ADi and available energy. If a task has a smaller execution time than its absolute deadline then the task may be feasible. It then checks the available energy at the time of execution and if the available energy is less than the required energy to execute the task then the task scheduling is feasible.

Now consider an example, suppose there are four tasks scheduled on the dual-core processor as shown in Table 1 Each task consists of its arrival time ai, execution time ci, deadline di and energy demand Edi.

Table 1 shows the feasibility of four tasks. Each task contains its arrival time, execution time, deadline and initially available energy. Now suppose EA is available energy at time instant t = 0, we compute the utilization of that task. Utilization of the task is smaller than 1, which means the task scheduling is feasible. We again checked the worst case execution time of the task and if it is less than the absolute deadline, then again task scheduling is feasible. Now we initially consider the battery is full, assume EC = EA = 100, energy is drained 1 unit for each interval of execution time and energy produced is 0.5 unit in the same time interval. We compared the drained energy with available energy. If drained energy is less than available energy then the task scheduling is feasible. Each task is scheduled on the basis of a smaller deadline.

As we have seen Static Energy Harvesting Earliest Deadline First Scheduling Algorithm have increased the lifetime of the system. We then proposed another approach which uses DVS to increase lifetime at run time. We reduced the voltage level of the system to schedule a task. The processing time of a task is increased by decreasing the voltage. If a task has sufficient deadline then we can apply DVS and decrease energy consumption.

Algorithm 2

Dynamic Energy Harvesting Earliest Deadline First Scheduling Algorithm with DVS

{Task set T = {T1, T2, …, Tn}, ReadyQueue = { }, est = execution start time, AD= Absolute deadline}

```
1.  ReadyQueue = empty
2.  Feasibility = TRUE
3.  At time t = $a_i$,
4.  $AD_i$ = $a_i$ + di , Ui = $\frac{c_i}{d_i}$
5.  If Ui ≤ 1
        Then ReadyQueue = ReadyQueue + {Ti}
6.  While ReadyQueue ≠ empty
7.  {
8.  $Select Ti \mid AD_i \leftarrow \min_{i=1 to n} AD_i$
9.  If WCET (esti + ci ) ≥ ADi THEN
        Feasibility = FALSE
10. Else if WCET (esti + ci ) ≤ ADi THEN
        If Edi ≥ EA THEN
            Feasibility = FALSE
        Else
            Apply DVS to reduce voltage level
            GOTO step 10
11. }
12. IF Feasibility ==TRUE
13.     Schedule the Ti on empty core
14.     TAG core as Busy with task Ti
15.     ReadyQueue = ReadyQueue – {Ti}
16. Exit
```

Table 1: Feasibility test for 4 tasks

Task	ai	ci	di	WCET	EA	ADi	Ui	Edi	Epi	Feasibility
Ti	0	10	40	10	100.0	40	0.25	10	5.0	TRUE
T2	5	15	30	25	95.0	35	0.43	15	7.5	TRUE
T3	10	16	60	51	82.5	70	0.23	16	8.0	TRUE
T4	10	10	55	35	87.5	65	0.15	10	5.0	TRUE

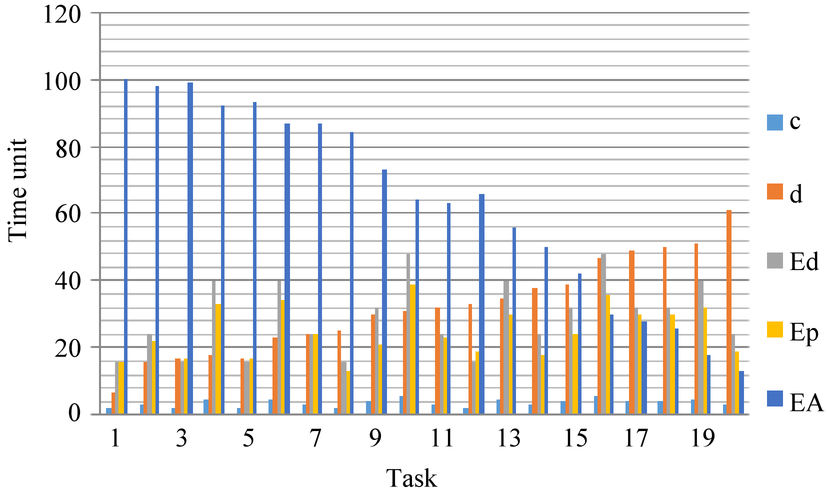

Fig. 2: Task Feasibility with energy harvesting

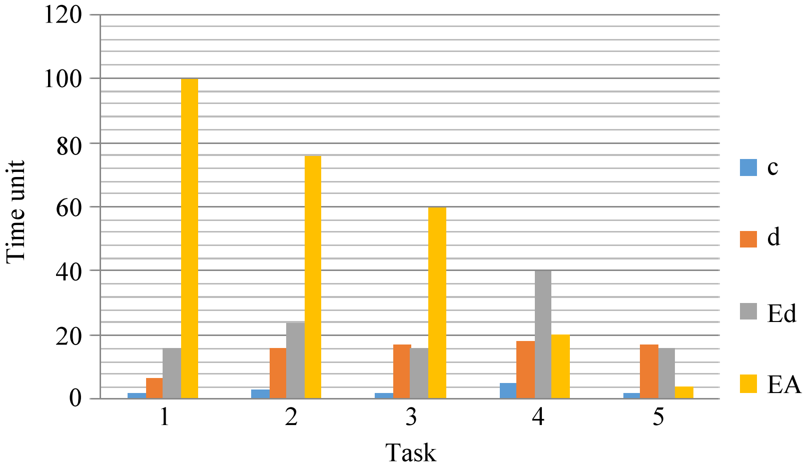

Fig. 3: Task feasibility without energy harvesting

Dynamic scheduling algorithm also works similar to the static scheduling algorithm, the only difference is that we can dynamically increase or decrease the voltage level as per system requirement which saves the system energy and enhance the lifetime of a system.

Experiment and Results

To evaluate the proposed algorithm, we implemented the proposed approaches in C++ and used Jetson TK1 development kit to check the task scheduling feasibility. We assumed initially battery is full with the capacity of 100units (time) and tasks are ready for execution. We also assumed that the energy is drained in multiples of 8 units with the execution time of each task. Figure 2 shows that the tasks are feasible if energy drained is less than the available energy and energy harvested.

As shown in Fig. 2, initially battery is full of 100 units of time. Available Energy decreases on increasing the execution of tasks.

Figure 3 shows the relation between the tasks and time taken for their execution. As there is no source of energy available for the charging of the battery, so the battery is drained after the execution of each task. After a point of time, it stops the execution of the tasks because of the unavailability of the energy, which is a dead end.

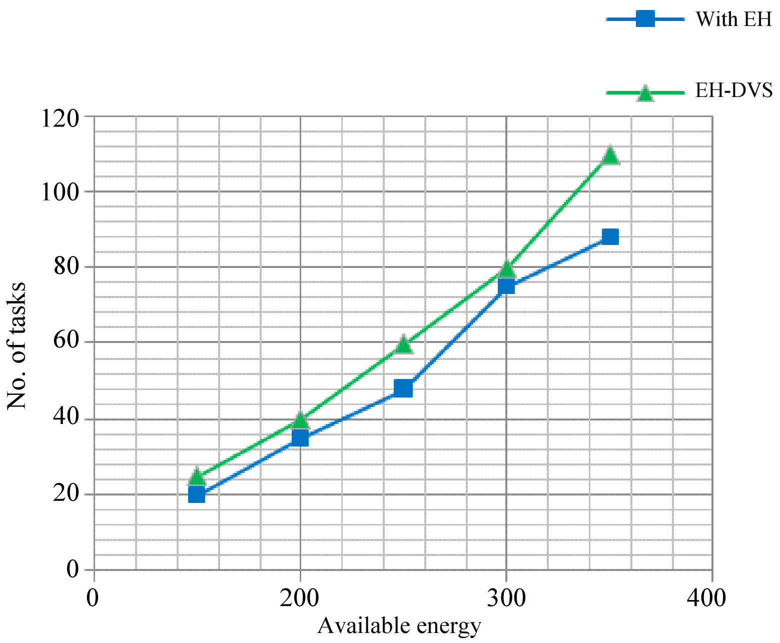

Fig. 4: Comparison of task scheduling with EH

Figure 4 portrays the comparison of the relationship between the number of tasks executed and available energy for two approaches which are real time task scheduling with energy harvesting (existing approach) and real time scheduling with energy harvesting and DVS (proposed approach). The number of tasks executed increases within the same available energy as compared to the existing approach.

In case of no energy harvesting, the feasibility of the task is very low. The task scheduling in energy harvesting approach increased as compared to existing approaches in which energy harvesting approach has not been applied. The feasibility of task has been increased by using energy harvesting with DVS simultaneously.

Conclusion

As the need for digital applications is increasing, there is a need for energy harvesting approach in most of the battery operated real time applications such as surveillance, military, health, forecasting, etc. To enhance the lifetime of such a system, energy harvesting with dynamic voltage scaling (DVS) is applied. Energy harvesting is the best approach for charging a battery in remote areas. In this paper, we developed an approach which increased the task feasibility and enhanced the lifetime of a system using energy harvesting with DVS. We applied both static and dynamic scheduling and compared the results with existing approaches. Experiments and results had shown that the proposed approach is more efficient for tasks scheduling in energy harvesting environment as compared to the existing approaches.

Acknowledgment

The Authors would like to thanks to Prof. M. N. Doja, Professor and Head, Department of Computer Engineering, Faculty of Engineering and Technology, Jamia Millia Islamia, New Delhi, India for providing us the laboratory and other facilities for smooth conduct of our research. We are also thankful to Mr. Chandra Cheij, Academic Programs Manager, NVIDIA Corporation, Lakeline Blvd. Bldg 2, Suite 100 , Austin for providing us JetsonTK1 Development Kits.

Ethics

This Article is original and unpublished. The authors declare that there are no ethical issues that may arise after the publication of this manuscript.

References

Abdeddaïm, Y., C. Younès, R.I. Davis and D. Masson, 2016. Response time analysis for fixed priority real-time systems with energy-harvesting, Real-Time Syst., 2: 125-160. DOI: 10.1007/s11241-015-9239-7

Allavena, A. and M. Daniel, 2001. Scheduling of Frame-based Embedded Systems with Rechargeable Batteries. University of Pittsburgh, USA.

Chen, J., T. Wei and J. Liang, 2014. State-Aware Dynamic Frequency Selection Scheme for Energy-Harvesting Real-Time Systems. IEEE Trans. Very Large Scale Integration Syst., 22: 1679-1692. DOI: 10.1109/TVLSI.2013.2278315

Chetto, M. and A. Queudet, 2014. A Note on EDF Schedulingfor Real-Time Energy Harvesting Systems. IEEE Trans. Comput., 63: 1037-1040. DOI: 10.1109/TC.2013.21

Francesco, Q., 2001. A Cost Model for Selecting Checkpoint Positions in Time Warp Parallel Simulation. IEEE Trans. Parallel Distri. Syst., 12: 346-362. DOI: 10.1109/71.920586

Ghor, H.E., M. Chetto and R.H. Chehade, 2011. A real-time scheduling framework for embedded systems with environmental energy harvesting. Comput. Electrical Eng., 37: 498-510. DOI: 10.1016/j.compeleceng.2011.05.003

Guohui, L., H. Fangxiao and Y. Ling, 2009. An energy-efficient scheduling scheme for aperiodic tasks in embedded systems. Proceedings of the IEEE Multimedia and Ubiquitous Engineering, Jun. 4-6, pp: 369-376.

Housseyni, W., O. Mosbahi, M. Khalgui and M. Chetto, 2016. Real-time scheduling of sporadic tasks in energy harvesting distributed reconfigurable embedded systems. Proceedings of the 13th International Conference on Computer Systems and Applications, 29 Nov.-2 Dec, IEEE Xplore press, Morocco, pp: 1-8. DOI: 10.1109/AICCSA.2016.7945682

Jejurikar, R. and G. Rajesh, 2002. Energy aware task scheduling with task synchronization for embedded real time systems. Proceedings of the International Conference on Compilers, Architecture and Synthesis for Embedded Systems, Oct. 08-11, ACM, France, pp: 164-169. DOI: 10.1145/581630.581655

Kim, K.H. and J. Kim, 2007. An Adaptive DVS Checkpointing Scheme for Fixed-Priority Tasks with Reliability Constraints in Dependable Real-Time Embedded Systems. In: Embedded Software and Systems, ser. Lecture Notes Comput. Sci., 4523: 560-571.

Kwon, W.C. and T. Kim, 2005. Optimal voltage allocation techniques for dynamically variable voltage processors. J. ACM Trans. Embedded Computing Syst., 4: 211-230. DOI: 10.1145/1053271.1053280

Lin, X., Y. Wang, N. Chang and M. Pedram, 2016. Concurrent task scheduling and dynamic voltage and frequency scaling in a real-time embedded system with energy harvesting. IEEE Trans. Computer-Aided Design Integr. Circ. Syst., 35: 1890-1902. DOI: 10.1109/TCAD.2016.2523450

Linjie, Z., W. Tongquan, C. Xiaodao G. Yonghe and H. Shiyan, 2012. Adaptive fault-tolerant task scheduling for real-time energy harvesting systems. J. Circuits Syst. Comput., 21: 01. DOI: 10.1142/S0218126612500041

Liu, S., Q. Qiu and Q. Wu, 2006. Task Merging for Dynamic Power Management of Cyclic Applications in Real-Time Multiprocessor System. Proceedings of the International Conference on Computer Design, Oct. 1-4, IEEE Xplore press, USA, DOI: 10.1109/ICCD.2006.4380847

Liu, S., Q. Qiu and Q. Wu, 2008. Energy aware dynamic voltage and frequency selection for real-time systems with energy harvesting. Proceedings of the Proceedings of the conference on Design, automation and test in Europe, Mar. 10-14, ACM, Germany, pp: 236-241. DOI: 10.1145/1403375.1403432

Lu, J. and Q. Qiu, 2011. Scheduling and Mapping of Periodic Tasks on Multi-core Embedded Systems with Energy Harvesting. Proceedings of the International Green Computing Conference and Workshops, July, 25-28, IEEE Xplore press, USA, DOI: 10.1109/IGCC.2011.6008595

Moser, C., D. Brunelli, L. Thiele and L. Benini, 2007. Real-time scheduling for energy harvesting sensor nodes. Real-Time Syst., 37: 233-260. DOI: 10.1007/s11241-007-9027-0

Qin, X. and T. Xie, 2008. An availability-aware task scheduling strategy for heterogeneous systems. IEEE Trans. Comp., 57: 188-199. DOI: 10.1109/TC.2007.70738

Rusu, C., R. Melhem and D. Mosse, 2005. Multi-version scheduling in rechargeable energy-aware real-time systems. J. Embedded Comp., 1: 271-283.

Shashank, P. and I.J. Daniel, 2009. Energy Harvesting Technologies. 1st Edn., Springer-Verlag, New York, USA, ISBN-10: 9780387764641, pp: 524.

Yasmina, A. and D. Masson, 2012. Real-time scheduling of energy harvesting embedded systems with timed automata. Proceedings of the IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, Aug. 19-22, IEEE Xplore press, DOI: 10.1109/RTCSA.2012.21

Zhang, Y. and K. Chakrabarty, 2006. A unified approach for fault tolerance and dynamic power management in fixed-priority real-time embedded systems. IEEE Trans. Computer-Aided Design Integrated Circuits Syst., 25: 111-125. DOI: 10.1109/TCAD.2005.852657