Energy Management for Energy Harvesting Real Time System with Dynamic Voltage Scaling

Ranvijay[1], Rama Shankar Yadav[1], Arvind Kumar[1], and Smriti Agrawal[2]

[1] Department of Computer Science and Engineering,
Motilal Nehru National Institute of Technology, Allahabad. India
[2] Department of Computer Science and Engineering,
Jaypee University of Information Technology, Solan, India
ranvijaymnnit@gmail.com, rsy@mnnit.ac.in,
arvinddagur@gmail.com, smriti.agrawal@juit.ac.in

Abstract. Energy harvesting has recently emerged as a feasible option to increase the operating time of battery based real time embedded systems. In this paper, we propose a scheduling algorithm that offers lesser energy consumption for battery powered dynamic real time system modeled with aperiodic tasks and energy harvesting constraints. As the harvested energy is highly dependent on the environment thus, available power/energy of storage changes over the time. The proposed approach has to decide which speed or voltage level is to be to select leading to reduction in energy overhead as well as timing overhead due to the speed switching. We further, improve the quality of service to accept more number of aperiodic tasks and improve the system performance in terms of remaining energy. Theorem is being derived to show the effectiveness our approach having lesser energy consumption as compared to existing one. The simulation results and examples illustrate that our approach can effectively reduce the overall system energy consumption and improve the system performance in terms of remaining energy as well as reduce the rejection ratio of aperiodic tasks.

Keywords: Real time systems, energy aware scheduling, harvested energy, dynamic voltage scaling, and quality of service (QoS).

1 Introduction

Energy minimization is a key issue for designing of real time embedded systems. This is especially important for battery powered systems. As the advancement of technology chip area reduces. Thus, less energy is storable on board. If deployed bigger battery on a chip as a result leading to their size as well as cost still severely limits the system's lifespan. The emerging technology of energy harvesting has earned much interest recently to provide a means for sustainable embedded systems [1] one of the important domain is wireless sensor network. Wireless sensor networks consisting of numerous minuscule sensors that are unobtrusively embedded in their environment. That sensor nodes scavenging ambient energy may operate perpetually, that has been constrained by their limited power supply. Most of the time in sensor network, recharging or replacing nodes' batteries is not practical due to inaccessibility

D.C. Wyld et al. (Eds.): NeCoM/WeST/WiMoN 2011, CCIS 197, pp. 536–548, 2011.

and sheer number of the sensor nodes. In order to solve the energy problem and increase the system operating time, environmental energy harvesting is deemed a promising approach. There are various source of harvesting energy such as solar, thermal, kinetic or vibrational energy, etc. most of the environmental energy sources varied over time [4]. For example, the harvested energy of a solar cell at a sunny noon is much higher than that at dawn. The feasibility of battery based real time system depends upon timing constraint as well as energy constraint. Energy constraint depends upon the stored energy as well as harvested energy from environment. The author [3, 5] is to maximize the utilization of solar energy, in this paper the authors sagest an algorithm, when the scavenged energy is low decrease the duty cycle in time (e.g.at night) and increase the duty cycle when scavenged energy is high (e.g. during the day). Author [2, 7] has been presented which demonstrate both feasibility and usefulness of sensors nodes which are powered by solar or vibrational energy. Kansal et al. [5] explore how to maximize the utilization of solar energy by minimizing the roundtrip losses of the battery. Moser et al. [6] develop lazy-scheduling to avoid deadline violation in energy-harvesting systems.

Existing strategy for energy minimization in support of aperiodic tasks [10, 11, 16] do not solve all this issues completely. For example, uncontrolled occasional deadline misses are possible by using the slacked EDF [13]. W. Yuan et al.[17] proposed energy aware algorithm for dynamic soft real-time multimedia applications. Author [19] provides offline energy aware scheduling approach for mixed task set. The algorithm presented by Shin and Choi in [18] also sets the initial voltage level using Static Voltage Scaling. Then they lower the voltage level further whenever a single task is eligible for execution. Lee et al. [20] developed their DVS algorithms using only two voltage levels and distributing the tasks into two sets, each corresponding to one of the voltage levels: High and Low.

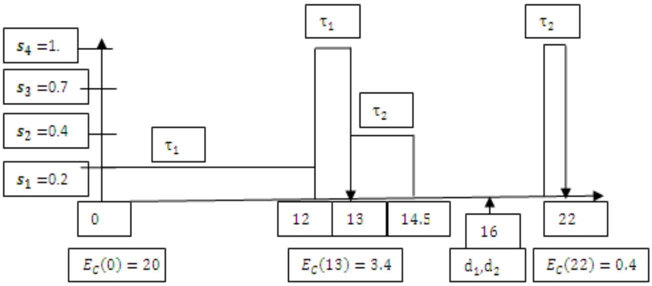


Fig. 1. Schedule of existing approach(EA-DVFSA)[]

Liu et al. [21] extend the results in [6] to taking into account dynamic voltage scaling processor to reduce the rate of deadline misses due to the shortage of energy. In this paper author propose energy aware dynamic voltage and frequency selection algorithm (EA-DVFA). According to EA-DVFA [21] when system have not enough energy execute some portion of the computation (execution) time on slower speed and some portion are scheduled at maximum speed level. This leading to more energy consumption due to the speed switching overhead and execute some portion of task at maximum speed level. As a consequence, future tasks have to violet their deadlines due to the shortage of energy.

In this paper, we aim to improve the quality of service (QoS) by accepting more number of aperiodic task as well minimize the energy consumption leading to elongate the operating time of battery. So, we will find the tradeoff between timing conflicts and energy conflicts for that we proposed the modified energy aware dynamic voltage and speed selection algorithm (MEA-DVSA). In this paper we reduce the energy overhead as well as timing overhead by utilizing the speed in such a way that response time of task is less than or equal to the existing approach even though on the cost of lesser energy consumption. The rest of the paper is organized as follows: in section 2, we describe our preliminary, followed by motivational example and system model. Sections 3 discuss our contribution. Section 4 and section 5 elaborates our proposed approach followed by results and analysis in section 6. Finally, paper concludes with section 7.

2 Preliminaries

Here, we provide the system model followed by the motivations for our approach.

2.1 Motivational Example

Consider a real time system with two aperiodic tasks, τ_1 (0, 16, 4), τ_2 (5, 16, 1.5), where the triplet of attributes denotes arrival time(a_i), deadline (d_i) and worst case execution time at maximum speed $e_i(s_{max})$. Assuming at time instance $= 0$, the stored energy $E_C(t) = 20.$ The harvested power from time interval 0 to 25 is set to 0.5. System having four normalized speed level say ($s_1 = 0.25$, $s_2 = 0.40$, $s_3 = 0.70$ and $s_4 = 1.0$). the power consumption at corresponding speed level are 1joule, 1.75 joule, 3.8 joule and 7joule per unit time Energy overhead due to the speed/voltage switching are as follows: $\beta_1^2 = 1.8\ \mu\, joule,\ \beta_1^3 = 2.1\ \mu\, joule,\ \beta_1^3 = 2.6\ \mu\, joule,\ \beta_1^4 = 3.1\ \mu\, joule,\ \beta_2^3 = 1.5\ \mu\, joule$, $\beta_2^4 = 1.9\ \mu\, joule$ and $\beta_3^4 = 1.2\ \mu\, joule.$

Energy Aware Dynamic Voltage and Frequency Selection (Existing approach)
Here, at time t=0 the total stored energy is $E_C(t) = 20.$ When first task τ_1 arrive at time t=0 it required 4 unit computation time at s_{max} speed level. The energy consumption of task τ_1 at s_{max} speed is 32 joule which is more than the available energy. Existing approach compute execute the some portion at lower speed and some portion run at maximum speed level to provide the time opportunity for future tasks. We can observe from the schedule system starts running task τ_1 at speed level s_1 up to time $t = 12$ and remaining computation time i.e. 1 unit at s_{max} from time $t = 12$ to time $t = 13$ and finish at time $t = 13$. Hence the energy consumption for the task τ_1. $E_{consumption}(\tau_1)$ = task processing energy consumption plus energy overhead due to energy switching. $E_{consumption}(\tau_1) = 12 * 1 + 1 * 7.5 + 3.1 = 23.1$ joule. Thus response time of task τ_1 is 13. The harvested energy $E_{Source}(0{,}13) = 6.5$ and $E_C(0) = 20$. Hence energy available at time t=13: $E_C(t) = E_C(0) + E_{Source}(0{,}13) - E_D(0{,}13) - \beta_1^4 E_C(13) = 20 + 6.5 - 19.5 - 3.1 = 3.4\, joule.$

The time t=13 to time t=16 the harvested energy $E_{Source}(13{,}16) = 1.5$ so the total available energy from time t= 13 to time t=16 is 4.9 joule leading to miss the deadline of task τ_2 because existing approach starts τ_1 at time t=13 at speed evel s_2 up to time t=14.5, while remaining computation time (0.4) unit executed at maximum speed levels_4 . For that it is required $\mathrm{E_{consumption}}(\tau_2) = 1.5 * 1.75 + 0.4 * 7.5 + 1.9 = 7.525$ joule of energy where as only 4.9 joule of energy is available up to time t=16. So, deadline of τ_2 is violated due to the shortage of energy.

On the other hand, if we compute a single speed in such away response time of task is same or less without going on maximum speed level. The power consumption increases exponentially at higher speed level as compare to lower speed level as well as energy save due to reducing the speed switching. This provides the better opportunity to insure the timing as well energy constraint of tasks. Thus, we proposed the modified dynamic voltage and speed algorithm that improve the acceptability domain by accepting more aperiodic tasks and minimize the energy consumption.

Before discussing the proposed modified energy aware dynamic voltage and frequency selection algorithm, we describe the system model and assumptions in the next section.

2.2 System Model

This system deals with energy minimization of random arrival pattern aperiodic tasks and is able to operate at different speed level. System modelled with energy source, energy storage, energy drain, DVS processor and real time aperiodic tasks.

Energy Source:
Harvesting source of energy is dependent on environmental factors. Such as solar, wind etc. they are highly varying with time, if (r) is the rate of harvesting power per unit time in any interval [t1, t2]. So total energy harvested in time interval [t1, t2] is as follows:

$$E_{Source}(t1, t2) = (t2 - t1) * r \tag{1}$$

Energy Storage:
Here, we assume a limited energy storage that may be charged up to its capacity C. If no tasks are executed and the stored energy has reaches its capacity leading to energy overflow.

$$0 \le EC(t) \le C \qquad \forall t \tag{2}$$

For executing the task, power $P_D\,(t)$ and the respective energy$E_D(t1, t2)$) is drained from the storage to execute tasks. We have the following relation:

$$E_C(t2) \le E_C(t1) + E_{source}(t1, t2) - E_D(t1, t2) \quad \forall t2 > t1$$

Therefore,

$$E_D(t1, t2) \le E_C(t1) + E_{source}(t1, t2) \; \forall t2 > t1 \tag{3}$$

Energy Drain:
Energy is the function of speed level s_i where, $(s_{low} \le s_i < s_{max})$. The energy drain in time interval $(t1, t2)$ is given as:

$$E_D(t1, t2) = \int_{t1}^{t2} P_D(s_i)(t)\, dt \tag{4}$$

Where $P_D(s_i)$ is the power drain at speed level s_i.

The other considerations are as follows:

1. Real time system modeled with aperiodic tasks τ_1, τ_2, τ_3 ... τ_n. Each task τ_i has the attributes, (a_i) is the arrival time, $e_i(s_{max})$ is the worst-case execution time at maximum speed level (s_{max}), and deadline (d_i) are known only after its release.
2. We assume that uni-processorr system.
3. We consider Dynamic priority Scheduling algorithm Earliest deadline first (EDF).

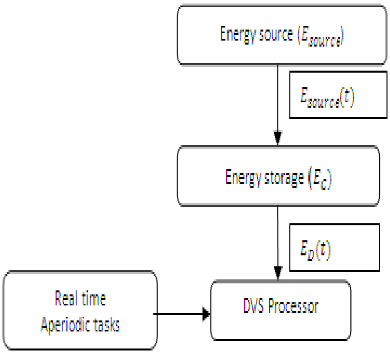
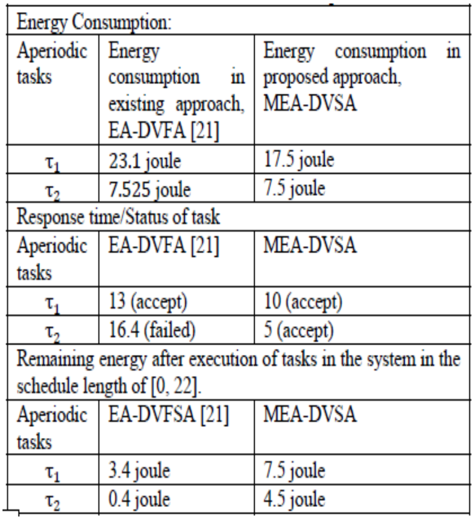


Fig. 2. Energy aware real time scheduling scenario

Table 1. Performance measurement of example 1



Energy Consumption:		
Aperiodic tasks	Energy consumption in existing approach, EA-DVFA [21]	Energy consumption in proposed approach, MEA-DVSA
τ_1	23.1 joule	17.5 joule
τ_2	7.525 joule	7.5 joule
Response time/Status of task		
Aperiodic tasks	EA-DVFA [21]	MEA-DVSA
τ_1	13 (accept)	10 (accept)
τ_2	16.4 (failed)	5 (accept)
Remaining energy after execution of tasks in the system in the schedule length of [0, 22].		
Aperiodic tasks	EA-DVFSA [21]	MEA-DVSA
τ_1	3.4 joule	7.5 joule
τ_2	0.4 joule	4.5 joule

4. DVS processor can operate at $\mathcal{N}$ discrete voltage levels, i.e., $V = \{v_{low}, v_2, v_3 \ldots v_{max}\}$ where each voltage level is associated with a corresponding speed from

the set $S = \{s_{low},\ s_2,\ s_3 \ldots\ s_{max}\}$.The speed s_{low} is the lowest operating speed level whereas s_{max} is the maximum speed level.

In this figure 2 the energy source harvest the energy from the environment and convert into electrical signal at particular rate of energy. This energy accumulated in the energy storage up to their capacity C; the stored energy at time t is denoted by $E_C(t)$. When the processor executes the real-time task, it has drawn the energy from energy storage. If the energy storage is empty, the processor stops functioning.

3 Contribution

- We proposed energy constrained scheduling algorithm for aperiodic time critical tasks whose energy storage is recharged by environmental source.
- We provide modified dynamic voltage and frequency algorithm to accept more number of tasks while elongate the operating time of battery.
- We present theorem that ensure our existing algorithm for aperiodic task leads to the same or lesser amount of energy consumption as compared to DVFSA.

In this paper we proposed the modified energy aware dynamic voltage and speed selection algorithm (MEA-DVSA). We reduce the energy overhead as well as timing overhead due to the speed switching by utilizing the speed in such a way that response time of task is less than or equal to the existing approach even though on the cost of lesser energy consumption.

4 Energy and Time Overhead

Every time the processor's speed level / supply voltage are switched from one speed/voltage level to other speed/voltage level. The change requires a certain amount of extra energy and time. That extra amount of energy and time is called energy overhead and time overhead respectively [13]. The energy overhead due to the speed /voltage switching form speed level (s_k) to speed level (s_j) is denoted by β_i^j where as α_i^j indicate the timing overhead.

$$\beta_k^j = Cr \cdot |Vdd_k - Vdd_j|^2 + Cs \cdot |Vbs_k - Vbs_j|^2 \tag{5}$$

$$\alpha_k^j = max(pVdd \cdot |Vdd_k - Vdd_j|, pVbs \cdot ||Vbs_k - Vbs_j|) \tag{6}$$

Where, *Cr* denotes power rail capacitance, and *Cs* the total substrate and well capacitance. Since transition times for Vdd and Vbs are different, the two constants p Vdd and p Vbs are used to calculate both time overheads independently. Vdd indicate the supply voltage whereas, Vbs indicate the body bias voltage. The energy overhead(β_k^j) due to speed switching from speed level (s_k) to speed level (s_j) is as same as for the speed switching from speed level (s_j) to speed level (s_k) i.e.

$$\beta_k^j = \beta_j^k \tag{7}$$

5 Modified Energy Aware Dynamic Voltage and Speed Selection

In this section we proposed a voltage and speed selection algorithm that reduces the energy as well as timing overhead incurred due to switching between speed levels.

As from example 1 existing approach schedule the some portion of task on lower speed and some portion on maximum speed level to prevent stealing excessive time from future tasks. Due to going on maximum speed level energy consumption increases exponentially (energy consumption is a function of cubic of speed) as well as energy consumption due to the speed switching that leading to miss the deadline due of future task due to the shortage of energy. So we will propose modified dynamic voltage and frequency selection algorithm that compute the speed of task with same or less response time with lesser energy consumption.

Run tasks at maximum speed: when any task τ_k arrived at any time t. its energy requirement at maximum speed level is less than the total available energy $E_C(t)$.

Energy Minimization

Run task at lower speed: when any task τ_k arrived at any time t. its energy requirement at maximum speed level is greater than the total available energy $E_C(t)$. If we start execution at maximum speed that leading to operable Vs non operable time of processor. Tasks start execution at maximum speed until the available energy is zero. When the available energy is zero the system has to stop running the task and delay the task execution until it has a scavenged energy. That's leading to miss the deadline of a task even though it may be feasible at lower speed level.

Assume task τ_k arrives with arrival time a_k , worst case execution time at $\boldsymbol{s_{max}}$ and deadline d_k. The available stored energy is $E_C(a_m)$ and harvested energy from a_k to d_k is $E_{Source}(a_k, a_k + d_k)$. We can calculate the system running time $(Srun_{s_i})$ at any speed level $(\boldsymbol{s_i})$ as same as the existing author such that all available energy is completed depleted at power P_n until time instance $a_k + d_k$

$$Srun_{s_i}=E_c(a_k) + \frac{E_{Source}(a_k, a_k + d_k)}{P_n} \tag{8}$$

When we slow down the speed of task τ_k, the execution time at any speed level s_i may be less than equal to its deadline and the energy consumption of task must be less than or equal to sum of available energy $E_c(a_k)$ as well as harvested energy $E_{Source}(a_k, a_k + d_k)$. Mathematically

$$e_i/s_i \leq d_i - a_i) \tag{9}$$

$$E_{consumption}(a_k(s_i)) \leq E_c(a_k) + E_{Source}(a_k, a_k + d_k) \tag{10}$$

So we can find the feasible speed s_i $(s_{low} \leq s_i \leq s_{max})$ under the constraint of equation 9 and equation 10.

We compute the earliest possible starting running time of task τ_k and as late as possible starting running time under constraint of equation 9 and equation 10 by equation 11 and 12.

$$S_{earlrast}(\tau_k) = \max(a_k, a_k + d_k - Srun_i) \tag{11}$$

$$\mathrm{S}_{\mathrm{late}} = \max(\mathrm{a}_{\mathrm{k}}, \mathrm{a}_{\mathrm{k}} + \mathrm{d}_{\mathrm{k}} - \mathrm{Srun}_{\mathrm{max}}) \tag{12}$$

Where $\mathrm{Srun}_{\mathrm{max}}$ is computed by equation 13.

$$\mathrm{Srun}_{\mathrm{s}_{\mathrm{max}}} = \mathrm{E}_{\mathrm{c}}(\mathrm{a}_{\mathrm{k}}) + \frac{\mathrm{E}_{\mathrm{Source}}(\mathrm{a}_{\mathrm{k}}, \mathrm{a}_{\mathrm{k}} + \mathrm{d}_{\mathrm{k}})}{\mathrm{P}_{\mathrm{max}}} \tag{13}$$

if $S_{earlrast}$ and S_{late} are same it indicates both are equal to a_k . i.e.

$$Srun_i \geq Srun_{max} \geq d_k \tag{14}$$

The equation 13 indicates system running time at maximum power is larger than d_k . Thus, the system has enough energy to run the task at maximum speed level.

If $S_{earlrast}$ less than S_{late} it means the system have not enough energy between the time instances a_k *and* $a_k + d_k$ to schedule the task τ_k at maximum speed level. In existing approach [21] they execute the task initially at lower speed level (s_i), if task is not finish up to S_{late} the remaining computation time executed at maximum speed level (s_{max}). In our approach we slow down the speed of task and compute the speed level $s_i$for task τ_k and execute the whole computation time at same computed speed level ($s_{compute}$).

$s_{compute}$ Can be computed as follows:

$$\left({}^{e_i}/_{s_i}\right) > (S_{late} - S_{earlrast}) \ \ where \ \ (s_{low} \leq s_i < s_{max}) \tag{15}$$

$$Z_i = \left(e_i - \left({}^{S_{late} - S_{earlrast}}/_{s_i}\right)\right) \tag{16}$$

Where Z_i is the remaining computation time.

$$Y_i = Z_i + (S_{late} - S_{earlrast}) \tag{17}$$

Where Y_i is the total required time to complete the task τ_k at speed level s_i as well as speed level s_{max}. Hence, ($s_{compute}$) is computed by equation 11.

$$s_{compute}\,(\tau_i) = \left({}^{e_i}/_{Y_i}\right) \tag{18}$$

Choose the speed level $S_{assigned}$ if it is available otherwise chose the nearest speed level where $\left({}^{e_i}/_{s_{compute}}\right) \leq d_i$ and energy drain at speed $S_{assigned}$ in any interval (t1, t2) is less than or equal to $E_D(t1, t2) \leq E_C(t1) + E_{source}(t1, t2)$.

Proposed modified energy aware dynamic voltage and speed selection algorithm (MEA-DVSA) is summarized as below. The effectiveness of proposed modified dynamic voltage and speed selection algorithm (MDVSA) compared to existing dynamic voltage and frequency selection (DVFSA) approach can be seen in example 1. Here, same example 1 is scheduled by the proposed approach. We can observe from the schedule (figure 4) when τ_1 arrives at time $t = 0$ and it required 4 unit of computation time at maximum speed. The power consumption at maximum speed level is 8 joule per unit time. Thus, energy consumption of task τ_1 at maximum speed level is 32 joule while energy available at time instance t=0 is only 20. So it is not process at maximum speed level. Thus, execute the task at lower speed level. We compute the speed ($\boldsymbol{s_{compute}}$) for task τ_1 by equation 18. We get, $s_{compute}\,(\tau_1) = 0.30$, but this speed level is not available in the system thus, we select the nearest speed level under the constraint of equation 9 and equation 10. So select speed level $s_2 = 0.40$ to run the task τ_1. Thus, task τ_1 finish at time t=10. The energy consumption of task τ_1 $E_{consumption}(\tau_1) = 10 * 1.75 = 17.5$ joule. the total stored energy at time $t = 10$ is $E_C(10) = 20 + 5 - 17.5 = 7.5$ joule. Thus, task τ_2 run at maximum speed and finish at time $t = 14$ hence, remaining energy in the system at time $t = 16$ is 1.5 joule.

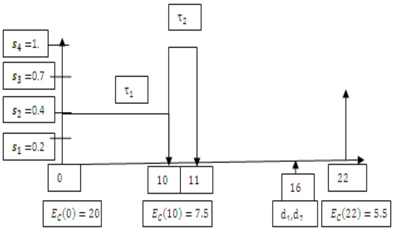


Fig. 3. schedual of existing approach (EAM-DVSA)

Whereas, only task τ_1 is feasible through existing approach [21]. The modified approach not only improves the acceptability domain by accepting more number of tasks it also reduces the response time.The effectiveness of proposed approach has been summarized in table 1.

Theorem: For DVS enable processor, the modified energy aware dynamic voltage frequency scaling (MEA-DVSA) algorithm for aperiodic task leads to the same or lesser amount of energy consumption as compared to EA-DVFA.

Proof: In both the approach we calculate earliest possible start time ($\boldsymbol{S_{earlrast}}$) and as possible as late start time ($\boldsymbol{S_{late}}$) of task. Case1: when $\boldsymbol{S_{earlrast}} = \boldsymbol{S_{late}}$

Both the approaches execute the task at maximum speed level (s_{max}). Hence, the energy consumption for both approaches is same.

Case2: when ($\boldsymbol{S_{earlrast}} < \boldsymbol{S_{late}}$), scale down the speed.

Start with lowest speed level $s_i \leftarrow s_{low}$ where $s_{low} \leq s_i < s_{max}$ When $\left({}^{e_i}/_{s_i}\right) \leq \boldsymbol{S_{late}} - \boldsymbol{S_{earlrast}}) \leq (d_i - a_i)$

Both the approach is able to execute the task at same reduced speed level s_i.

Case3: when ($\boldsymbol{S_{earlrast}} < \boldsymbol{S_{late}}$), scale down the speed Start with lowest speed level $S_i \leftarrow S_{low}$ where, $S_{low} \leq S_i \leq S_{max}$. When $\left({}^{e_i}/_{s_i}\right) \geq \boldsymbol{S_{late}} - \boldsymbol{S_{earlrast}}) \leq (d_i - a_i)$ Existing DVFSA start the execution of task at speed level s_i up to s_2 and the remaining computation time are executed at maximum speed.

So there is a speed switching even though there is no higher priority tasks. So the energy consumption of task is the amount of time task running at speed level s_i, remaining computation time executed at maximum speed level (s_{max}) and the energy consumption due to the speed/voltage switching. In proposed MEA-DVSA task executed at $S_{assigned}$ throughout the execution. Energy consumption of task is the amount of time task running at $S_{assigned}$. i.e. only due to the speed level $S_{assigned}$. Energy consumption is a function of cubic of speed level. So energy consumption of task by MEA-DVSA is lesser than existing EA-DVFA. In view of all the three cases we can say MEA-DVSA algorithm for aperiodic task leads to the same or lesser amount of energy consumption as compared to EA-DVFA. Hence, prove. In the view of above theorem we hence have the following corollary.

```
Algorithm
    1. Maintain a ready task queue Q
    3 $Z_i \leftarrow 0$
    4. $Y_i \leftarrow 0$
    5.$t \leftarrow current\ time$
    3. While (true)
    4.$d_{i\leftarrow}\ min\{d_j: T_j \in Q\}$
    5. Calculate $S_{earlrast}$and $S_{late}$ for task $\tau_i$
    6. if $S_{earlrast} = S_{late})$ then
    7. Execute task $\tau_i$ at maximum speed of processor or voltage.
    9 end if
    8. if ( $S_{earlrast} < S_{late})$
    9. Slow down the speed
    10. $s_i \leftarrow s_{low}$
    11. While $({}^{e_i}/_{s_i} \leq d_i - a_i)$
      {
          $if\ \left(({}^{e_i}/_{s_i}) > (S_{late} - S_{earlrast})\right)$
          $Z_i = (e_i - ({}^{S_{late} - S_{earlrast}}/_{s_i}))$
            $Y_i = Z_i + (S_{late} - S_{earlrast})$
                $s_{compute} = \left({}^{e_i}/_{Y_i}\right)$
      End if
      }
                    $s_i \leftarrow s_{i+1}$
      Go to level 11
              if
                    t=ak
                    then add task k to Q
                    Goto step 1
                      if t=fk
                  then remove the task from Q
              End if

End
```

Corollary 1: let us consider a system is characterized by energy capacity C, r is the rate of harvesting energy through source Es an interval $[t_1, t_2]$. If our approach can not schedule the given task due to the shortage of energy then existing EA-DVFA is not able to schedule it.

The next section deals with performance measurement of multi budget bandwidth preserving server through simulations.

6 Simulation and Result Discussion

In this section simulation of synthesized tasks are performed to evaluate the performance of the proposed window based lazy scheduling followed by the speed stretching approach to save energy. The processor is capable of voltage and frequency scaling. With five speed level 100 MHz, 200 MHz, 500MHz, 750MHz, and 1000 MHz. We assumed Power consumption at corresponding speed level is 60mW, 180Mw, 750Mw, 1800mW and 3200mW. Aperiodic tasks were generated by the exponential distribution using with inter arrival time (1/λ) and service time (1/μ) with parameters λ and μ.simulation is run for 10000 .Here, we compare the performance of proposed modified energy aware Dynamic voltage and speed algorithm refer MEA-DVSA with existing energy aware Dynamic voltage and frequency algorithm referred as EA-DVFA [21]. The key parameters for performance measurment are remaning energy and acceptence ratio.

In the following section we measure the effect of variation in aperiodic load on the average energy consumption and acceptense ratio of aperiodic task.

Effect of load on average remaning energy

The effect of load on the remaning energy consumption can be seen from the figure 4. This compare the performance of EA-DVFA and MEA-DVSA. In this we set the storage capacity is to 2000. We observe from the figure 4 when the aperiodic load increses the remaining energy will decreases. When we varies aperiodic load from 10% to 90% we observe from the figure as load increase remaining energy of the system decreases. At lower aperiodic load (10% to 40%) our modified approach have significant saving in energy almost store 15% more energy ac compared to existing EA-DVFA [21] approach. This is due to the at lower aperiodic load our approch run the whole compation of task most of the time at slower speed and same speed level however, existing approach execute some portion at lower speed and remaning comptation time at maximum speed level even there is no any higher priority tasks.

Effect of load on acceptence ratio

The effect of load on the acceptence ratio of aperiodic tasks can be seen from the figure 5. Figure 5 compare the performance of EA-DVFA and MEA-DVSA. In this we set the storage capacity is to 2000. We observe from the figure 6 when the aperiodic load increses the acceptence ratio will decreases. At lower aperiodic load (10% to 40%) our modified approach have accept 10% more aperiodic tasks as compared to existing EA-DVFA [21].

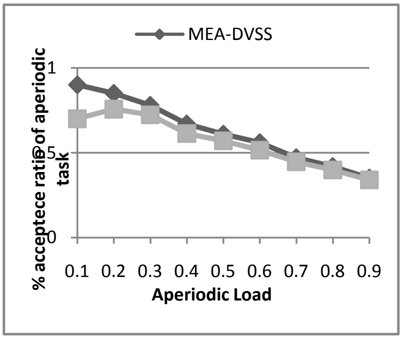


Fig. 4. % Acceptence of aperiodic task

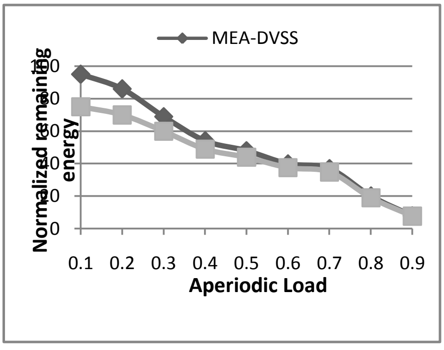


Fig. 5. Average remaining energy

7 Conclusion

In this paper we have presented a general scheduling algorithm that maximize the utility of harvested energy for real time embedded system with voltage scalable processor. The proposed approach judiciously decides operating speed that reducing the energy overhead as well as timing overhead due to the speed switching. The existing EA_DVFA switch the speed level of the task even though there is no any aperiodic task.

The examples and simulation studies shows that the proposed scheduling algorithm (MEA-DVSA) improves the overall average remaining stored energy. The average remaining stored energy of the system is approximately 5 % at aperiodic load varied from 70% to 90% while 20% more energy will be stored at lower aperiodic load varied from 10% to 50%. When the aperiodic load is low say 10% to 50% our praposed approach accept 8 % more task than existing approach. However, at higher aperiodc load both approach performa almost same. Thus, extensive simulation and illustrative example shows that our proposed approach is capable of performing better in terms of average stored remaining energy of the system as well as acceptance ratio of aperiodic tasks.

References

1. Li, D., Chou, P.H.: Application/architecture power co-optimization for embedded systems powered by renewable sources. In: DAC 2005: Proceedings of the 42nd Annual Conference on Design Automation, pp. 618–623. ACM, New York (2005)
2. Ammar, Y., Buhrig, A., Marzencki, M., Charlot, B., Basrour, S., Matou, K., Renaudin, M.: Wireless sensor network node with asynchronous architecture and vibration harvesting micro power generator. In: sOc-EUSAI 2005: Proceedings of the 2005 Joint Conference on Smart Objects and Ambient Intelligence, pp. 287–292. ACM Press, New York (2005)
3. Hsu, J., Zahedi, S., Kansal, A., Srivastava, M.: Raghunathan. Adaptive duty cycling for energy harvesting systems. In: ISLPED 2006: Proceedings of the 2006 International Symposium on Low Power Electronics and Design, pp. 180–185. ACM Press, New York (2006)
4. Roundy, S., Steingart, D., Frechette, L., Wright, P., Rabaey, J.M.: Power sources for wireless sensor networks. In: Karl, H., Wolisz, A., Willig, A. (eds.) EWSN 2004. LNCS, vol. 2920, pp. 1–17. Springer, Heidelberg (2004)
5. Kansal, A., Hsu, J., Zahedi, S., Srivastava, M.B.: Power management in energy harvesting sensor networks. ACM Transactions on Embedded Computing Systems (in revision) (May 2006); also available from: NESL Technical Report Number: TR-UCLA-NESL-200605-01
6. Moser, C., Brunelli, D., Thiele, L., Benini, L.: Real-time scheduling for energy harvesting sensor nodes. Real-Time Systems 37, 233–260 (2007)
7. Jiang, X., Polastre, J., Culler, D.E.: Perpetual environmentally powered sensor networks. In: Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks, UCLA, USA, April 25-27, pp. 463–468 (2005)
8. Mejía-Alvarez, P., Levner, E., Mossé, D.: Adaptive Scheduling Server for Power-Aware Real-Time Tasks. ACM Trans. Embedded Computing Systems 3(2), 284–306 (2004)
9. Aydin, H., Melhem, R.G., Mossé, D., Mejía-Alvarez, P.: Power- Aware Scheduling for Periodic Real-Time Tasks. IEEE Trans. Computers 53(5), 584–600 (2004)

10. Hong, I., Potkonjak, M., Srivastava, M.B.: On-Line Scheduling of Hard Real-Time Tasks on Variable Voltage Voltage Processor. In: Proc. Int'l Conf. Computer-Aided Design, pp. 653–656 (1999)
11. Sharma, V., Thomas, A., Abdelzaher, T., Skadron, K., Lu, Z.: Power-Aware QoS Management in Web Servers. In: Proc. IEEE Real-Time Systems Symp., pp. 63–72 (2003)
12. Pillai, P., Shin, K.G.: Real-Time Dynamic Voltage Scaling for Low-Power Embedded Operating Systems. In: Proc. 18th Symp. Operating Systems Principles, pp. 89–102 (2001)
13. Sinha, A., Chandrakasan, A.P.: Energy Efficient Real-Time Scheduling. In: Proc. Int'l Conf. Computer-Aided Design, pp. 458–470 (2001)
14. Zhu, Y., Mueller, F.: Feedback EDF Scheduling Exploiting Dynamic Voltage Scaling. In: Proc. IEEE Real-Time and Embedded Technology and Applications Symp., pp. 203–212 (2004)
15. Shin, Y., Choi, K.: Power Conscious Fixed Priority Scheduling for Hard Real-Time Systems. In: Proc. Design Automation Conf., pp. 134–139 (1999)
16. Qadi, A., Goddard, S., Farritor, S.: A Dynamic Voltage Scaling Algorithm for Sporadic Tasks. In: Proc. IEEE Real-Time Systems Symp., pp. 52–62 (2003)
17. Doh, Y., Kim, D., Lee, Y.-H., Krishna, C.M.: Constrained Energy Allocation for Mixed Hard and Soft Real-Time Tasks. In: Proc. of Int. Conf. on Real-Time and Embedded Computing Systems and Applications, pp. 533–550 (2003)
18. Shin, Y., Choi, K.: Power Conscious Fixed Priority Scheduling for Hard Real-Time Systems. In: Proceedings of the Design Automation Conference, pp. 134–139 (June 1999)
19. Yuan, W., Nahrstedt, K.: Integration of Dynamic Voltage Scaling and Soft Real-Time Scheduling for Open Mobile systems. In: Proc. of Int. Workshop on Network and Operating Systems Support for Digital Audioand Video, pp. 105–114 (2002)
20. Lee, Y.-H., Krishna, C.M.: Voltage-Clock Scaling for Low Energy onsumption in Real-Time Embedded Systems. In: Proceedings of the Sixth Int'l Conf. on Real Time Computing Systems and Applications, pp. 272–279 (1999)
21. Liu, S., Qiu, Q., Wu, Q.: Energy Aware Dynamic Voltage and Frequency election for Real-Time Systems with Energy Harvesting. In: Design, Automation and Test in Europe, DATE 2008 (2008)