

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Power Management in Supercapacitor-Based Wireless Sensor Nodes

Hengzhao Yang and Ying Zhang

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/64987>

Abstract

This chapter studies a power management problem for supercapacitor-based wireless sensor nodes with energy harvesting capabilities. A dependent task scheduling algorithm for nonpreemptable tasks with precedence constraints is developed. The modified first in first out (MFIFO) algorithm takes into account supercapacitor state and energy harvesting. Task precedence constraints are handled by defining a variable called task effective release time. Results show that the MFIFO algorithm improves the energy performance of the first in first out (FIFO) algorithm and maintains the timing performance at the same time.

Keywords: algorithm, energy harvesting, power management, supercapacitor charge redistribution, dependent task scheduling, wireless sensor network

1. Introduction

Wireless sensor networks have been developed for many applications. A wireless sensor network is composed of a large number of spatially distributed wireless sensor nodes. Wireless sensor nodes are usually powered by nonrechargeable batteries with limited capacity. Therefore, energy efficiency is a major concern. To maximize the network lifetime, various power management strategies have been proposed to minimize the energy consumption. In the meantime, numerous energy harvesting technologies have been developed to increase the energy income. Environmentally powered wireless sensor nodes usually need energy storage systems [1] to buffer the harvested energy. Typical energy storage systems include rechargeable batteries [2], supercapacitors [3–5], and hybrid systems [6, 7]. In general, rechargeable batteries have a larger capacity while supercapacitors have a much longer cycle

life. The major drawback of supercapacitors is their high self-discharge rate. Supercapacitor characteristics must be taken into account to develop effective power management solutions. For instance, supercapacitor self-discharge is considered in Refs. [3, 6]. This is because the supercapacitor terminal voltage is a critical parameter in analyzing the power behavior of supercapacitor-based energy storage systems, and self-discharge results in voltage drop. Because of the significance of the voltage drop during self-discharge, this characteristic has been extensively examined [8–14].

While supercapacitor self-discharge leads to voltage drop, this characteristic cannot completely characterize the supercapacitor voltage behavior. In fact, the supercapacitor voltage may increase under the open circuit condition [15], which is due to the charge redistribution characteristic. A mechanism of the low ionic mobility in supercapacitor micropores is identified in Ref. [16]. The impact of charge redistribution on power management is qualitatively investigated in Ref. [17]. A detailed analysis of the voltage change during charge redistribution is performed in Ref. [18]. In Ref. [19], the modified earliest deadline first (MEDF) algorithm is developed for scheduling independent tasks.

This chapter extends the results in Refs. [17–19] and studies a new power management problem. Specifically, this chapter develops a modified first in first out (MFIFO) algorithm for scheduling tasks with precedence constraints in environmentally powered wireless sensor nodes that use supercapacitor-based energy storage systems. The MFIFO algorithm takes into account supercapacitor charge redistribution and energy harvesting. Task precedence constraints are handled by defining a variable called task effective release time. While the first in first out (FIFO) algorithm only considers the timing constraints of tasks, the MFIFO algorithm also considers the energy constraints.

The remainder of this chapter is organized as follows. Section 2 describes a system model for analyzing the power flow in wireless sensor nodes. Section 3 develops the MFIFO algorithm. Section 4 illustrates the implementation setup. A case study and extensive simulations are performed to evaluate the algorithm performance. These qualitative and quantitative results demonstrate that the MFIFO algorithm improves the energy performance of the FIFO algorithm while maintaining its timing performance. Section 5 concludes this chapter.

2. A power model for wireless sensor nodes

2.1. System model

This chapter adopts the wireless sensor node power model used in Ref. [19]. As shown in **Figure 1**, this model is composed of five modules: energy harvester, input power conditioning unit, energy buffer, output power conditioning unit, and energy user. Energy harvesters such as solar cells and piezoelectric films convert energy in other forms to electricity. Typically, an input power conditioning unit is needed to bridge the energy harvester and the energy buffer. For example, a solar-powered wireless sensor node usually includes a maximum power point tracker (MPPT). Energy buffers such as rechargeable batteries and supercapacitors are devices

that store the harvested energy. An output power conditioning unit is often necessary to generate a suitable power supply for the energy user. DC-DC converters are commonly used modules to bridge the energy buffer and the energy user. Energy users are mainly RF transceivers, microcontrollers, and sensors.

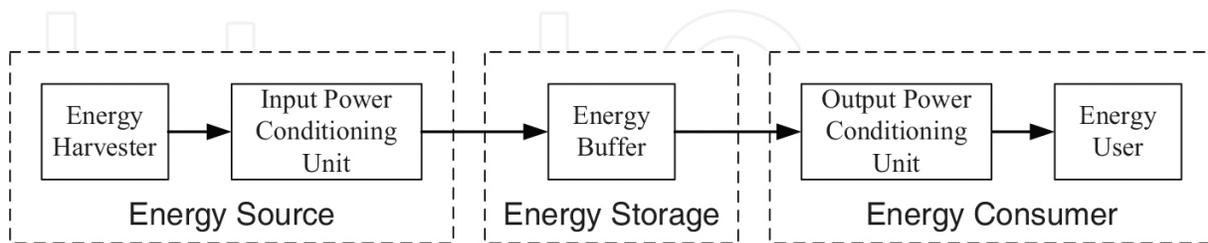


Figure 1. A power model for wireless sensor nodes.

The power model can be further abstracted to facilitate analyzing the power flow in wireless sensor nodes. As shown in **Figure 1**, the power model has three components: energy source, energy storage, and energy consumer. The energy source includes the energy harvester and the input power conditioning unit. For clarity, the energy buffer is referred to as the energy storage in this three-component model. The energy consumer combines the output power conditioning unit and the energy user. This abstracted model introduces two benefits. First, by separating energy buffers and power conditioning units, it is more convenient to study the impact of energy buffer characteristics on power management in wireless sensor nodes. Second, experiments with energy buffers can be readily designed and performed. The effects of input and output power conditioning units are taken into account in the process of designing the experiments.

2.2. Energy source model

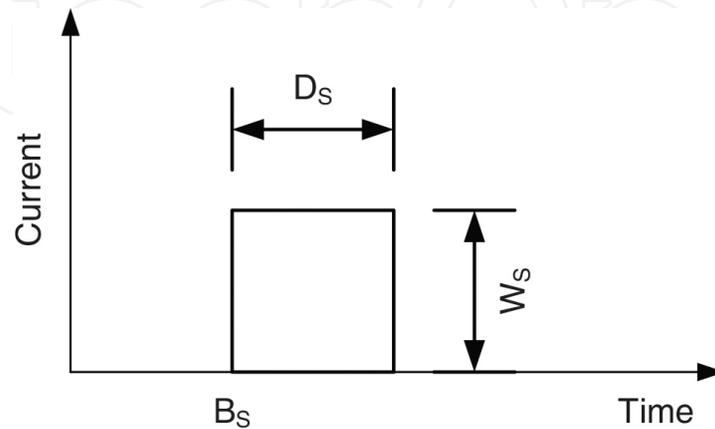
The component models are shown in **Figure 2**. The energy source is modeled as a current pulse train. As shown in **Figure 2(a)**, each current pulse is characterized by three parameters: begin time B_s , duration D_s , and weight W_s , which is the current magnitude. It should be noted that the energy source pulse is the conditioned pulse that is actually injected into the energy storage system. For example, in a solar-powered sensor node, the current pulse conditioned by the MPPT and fed into the energy storage system is the current pulse described in this energy source model. By tuning these three parameters, energy source profiles with different characteristics such as time span and power level can be generated.

2.3. Energy storage model

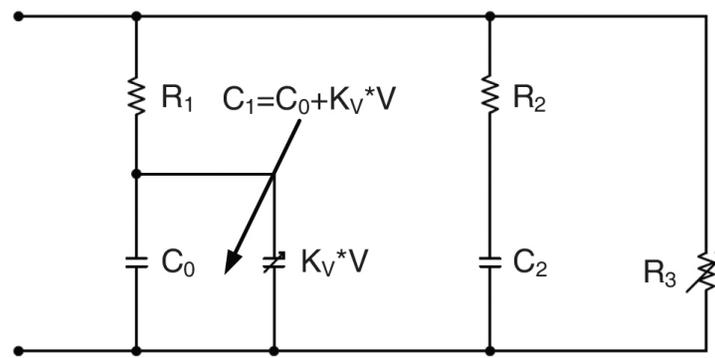
The energy storage system is a single supercapacitor. **Figure 2(b)** shows the variable leakage resistance (VLR) model [10, 11, 17, 18] for supercapacitors, which is a simplified equivalent circuit model. In this model, the first branch has three components: resistor R_1 , constant capacitor C_0 , and voltage-dependent capacitor $K_V * V$. The total capacitance of the first branch is $C_1 = C_0 + K_V * V$. This branch models the voltage dependency of supercapacitor capacitance.

The second branch includes resistor R_2 and capacitor C_2 . This branch models the charge redistribution behavior. The variable leakage resistor R_3 characterizes the time varying self-discharge.

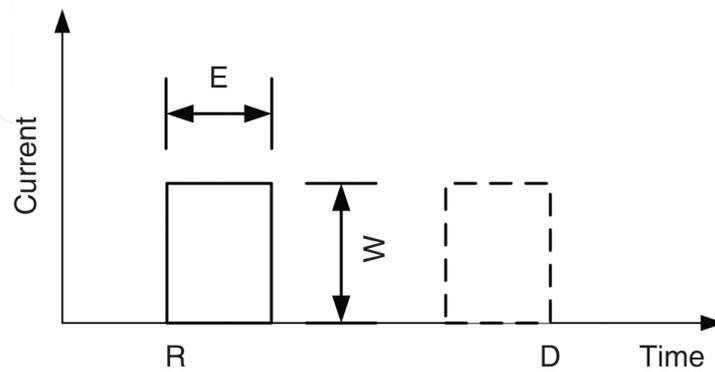
In addition to the model parameters, the voltages across the capacitors in the VLR model are also critical to determine the supercapacitor state. The charge stored in a supercapacitor tends to redistribute among RC branches after a charging or discharging process because each branch



(a)



(b)



(c)

Figure 2. Wireless sensor node component models. (a) Energy source. (b) Energy storage. (c) Energy consumer.

has a different time constant. Charge redistribution is a transient response to the supercapacitor initial state, which is characterized by the initial voltages V_1 and V_2 across the capacitors C_1 and C_2 . For example, if $V_1 > V_2$, the supercapacitor terminal voltage decreases with time because part of the charge stored in C_1 is transferred to C_2 . Therefore, to characterize a supercapacitor, the initial voltages across the capacitors in the VLR model must be specified.

2.4. Energy consumer model

When the energy consumer initiates an event, the energy storage system is assigned a task. The energy consumer is therefore modeled as a current pulse train. As shown in **Figure 2(c)**, each task is defined by four parameters: release time R , execution time E , absolute deadline D , and weight W . The release time is the instant of time at which the task becomes available for execution. The execution time is the amount of time required to complete the execution. The absolute deadline is the instant of time by which the task execution is required to be completed. Release time, execution time, and deadline are temporal parameters defining the timing constraint of a task. The weight of a task is its current magnitude. This parameter defines the energy constraint of a task.

2.5. Task precedence constraint and effective release time

In this chapter, the tasks are assumed to be dependent and nonpreemptable. In addition to the four parameters (release time, execution time, deadline, and weight) used to characterize the task model, a task may also have precedence constraints. If tasks are constrained to execute in some order, they are said to have precedence constraints. The precedence constraints among tasks are specified using precedence relations [20]. A task T_p is a predecessor of another task T_q (and T_q a successor of T_p) if T_q cannot begin the execution until the execution of T_p completes. This fact is usually denoted by $T_p < T_q$. Two tasks are independent when neither $T_p < T_q$ nor $T_p > T_q$. A task with predecessors is ready for execution when the time is at or after its release time and executions of all the predecessors are completed. Without loss of generality, it is assumed that in this chapter a task may have no more than one predecessor or successor for simplicity.

The release times of tasks with precedence constraints are sometimes inconsistent with the precedence constraints, which means that the release time of a task may be later than that of its successor. **Figure 3** shows two tasks T_p and T_q using solid lines. If $T_p < T_q$, the release time of task T_q is earlier than the release time of T_p , which is not consistent with the precedence constraint. A parameter called the effective release time of a task is defined to deal with such inconsistency. The effective release time of a task without predecessor is equal to its release time. The effective release time of a task with predecessor is equal to the maximum value between its release time and the release time of its predecessor plus the execution time of its predecessor. For example, the effective release time ER_q of T_q is defined by Eq. 1 depending on whether there is a precedence constraint. As shown in **Figure 3**, the task T_q denoted by dashed lines shows its effective release time if $T_p < T_q$.

$$ER_q = \begin{cases} \max(R_q, R_p + E_p), & T_p < T_q; \\ R_q, & \text{otherwise.} \end{cases} \quad (1)$$

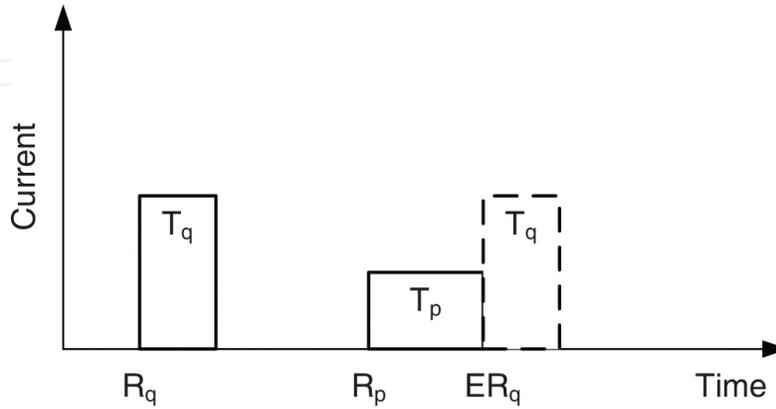


Figure 3. Definition of task effective release time.

3. MFIFO algorithm development

The MFIFO algorithm has three steps. First, create an initial schedule using the FIFO algorithm. This step takes care of the timing constraints and precedence constraints of tasks. Second, calculate the ready time adjustment margin based on the initial schedule. This margin determines how much delay is allowed if the ready time of the initial schedule is adjusted. Third, the ready time offset is determined based on supercapacitor state and energy harvesting. The start time of a task is the ready time plus the ready time offset.

3.1. Create an initial schedule using FIFO algorithm

The FIFO algorithm is used to create an initial schedule to ensure that the task precedence constraints are satisfied. The tasks are originally defined by the task set $T = T_i(R_i, E_i, D_i, W_i)$ and the precedence constraints $T_p < T_q$. The precedence constraints are transformed into timing constraints by defining the effective release times. A task is then characterized by four parameters: effective release time ER , execution time E , deadline D , and weight W . The task set is now $T_E = T_i(ER_i, E_i, D_i, W_i)$. The FIFO algorithm sorts the effective release times in the ascending order and determines the ready times of the tasks. The initial schedule is determined using Algorithm 1.

Algorithm 1: Create an initial schedule using FIFO algorithm.

Require: A set of N ready but not scheduled tasks: $T = T_i(R_i, E_i, D_i, W_i)$ and task precedence constraints: $T_p < T_q$.

```

1:   for  $i = 1 : N$  do
2:       if  $T_p < T_q$  then
3:            $ER_q = \max(R_q, R_p + E_p)$ 
4:       else
5:            $ER_q = R_q$ 
6:       end if
7:   end for
8:   Sort  $N$  tasks in the ascending order of their effective release times.
9:   Current Time  $T_C = 0$ .
10:  for  $i = 1 : N$  do
11:      Ready Time  $A_i = \max(T_C, ER_i)$ 
12:      Current Time  $T_C = A_i + E_i$ 
13:  end for
14:  Algorithm output is initial schedule  $T_{FIFO}$  defined by task ready time  $A_i$ :
       $T_{FIFO} = T_i(A_i, E_i, D_i, W_i)$  and modified task set  $T_E = T_i(ER_i, E_i, D_i, W_i)$ .

```

3.2. MFIFO algorithm

Once the initial schedule is determined, the ready time adjustment margin and ready time offset are calculated using the algorithms for the second and third steps in the MEDF algorithm [19], respectively. In particular, the release times used in the MEDF algorithm should be replaced by the effective release times. The complete MFIFO algorithm is summarized in Algorithm 2. The inputs of this algorithm include a set of N ready but not scheduled tasks, $T = T_i(R_i, E_i, D_i, W_i)$; task precedence constraints, $T_p < T_q$; energy source model, $E_S(B_S, D_S, W_S)$; and supercapacitor initial state, $V_1(t = 0)$ and $V_2(t = 0)$. The MFIFO algorithm is a three-step process:

1. **Step 1:** Create an initial schedule using Algorithm 1. The input of this algorithm is the task set $T = T_i(R_i, E_i, D_i, W_i)$ and task precedence constraints $T_p < T_q$. The output is the initial schedule T_{FIFO} defined by task ready time A_i : $T_{FIFO} = T_i(A_i, E_i, D_i, W_i)$ and modified task set $T_E = T_i(ER_i, E_i, D_i, W_i)$.
2. **Step 2:** Calculate ready time adjustment margin of the initial schedule using the second algorithm in Ref. [19]. The inputs are the modified task set $T_E = T_i(ER_i, E_i, D_i, W_i)$ and the initial schedule $T_{FIFO} = T_i(A_i, E_i, D_i, W_i)$. The output is the task ready time adjustment margin M_i .
3. **Step 3:** Determine ready time offset of the initial schedule using the third algorithm in Ref. [19]. The inputs are the modified task set $T_E = T_i(ER_i, E_i, D_i, W_i)$, the initial schedule $T_{FIFO} = T_i(A_i, E_i, D_i, W_i)$, the ready time adjustment margin M_i , the energy source model

$E_s(B_s, D_s, W_s)$, and the supercapacitor initial state $V_1(t=0)$ and $V_2(t=0)$. The output is the modified schedule T_{MFIFO} defined by task start time S_i : $T_{MFIFO} = T_i(S_i, E_i, D_i, W_i)$.

Algorithm 2: MFIFO algorithm

Require: A set of N ready but not scheduled tasks: $T = T_i(R_i, E_i, D_i, W_i)$; task precedence constraints: $T_p < T_q$; energy source model: $E_s(B_s, D_s, W_s)$; and supercapacitor initial state: $V_1(t=0)$ and $V_2(t=0)$.

- 1: Step 1: Create an initial schedule using Algorithm 1.
- 2: Input: task set T and task precedence constraints $T_p < T_q$.
- 3: Output: initial schedule T_{FIFO} and modified task set T_E .
- 4: Step 2: Calculate ready time adjustment margin of the initial schedule using the second algorithm in Ref. [19].
- 5: Input: modified task set T_E and initial schedule T_{FIFO} .
- 6: Output: task ready time adjustment margin M_i .
- 7: Step 3: Determine ready time offset of the initial schedule using the third algorithm in Ref. [19].
- 8: Input: modified task set T_E ; initial schedule T_{FIFO} ; ready time adjustment margin M_i ; energy source model: $E_s(B_s, D_s, W_s)$; and supercapacitor initial state: $V_1(t=0)$ and $V_2(t=0)$.
- 9: Output: modified schedule T_{MFIFO} .
- 10: MFIFO Algorithm Output: modified schedule T_{MFIFO} defined by task start time S_i : $T_{MFIFO} = T_i(S_i, E_i, D_i, W_i)$.

4. MFIFO algorithm implementation and evaluation

4.1. Simulation setup

The MFIFO algorithm is implemented and evaluated using a simulation setup similar to the one used for the MEDF algorithm [19]. The energy source and energy storage models are exactly the same. The energy consumer model is modified. Each task set has six periodic tasks and each periodic task has five jobs. Therefore, each task set is composed of 30 tasks. The timing and energy parameters of a task are defined in the same way as the one used for the MEDF algorithm, too. The precedence constraints are assigned with controlled randomness. The six periodic tasks are partitioned into three groups. Each group consists of two periodic tasks. For convenience, the six periodic tasks are numbered as $\{P_1, P_2, \dots, P_6\}$. The three groups are then $\{P_1, P_2\}$, $\{P_3, P_4\}$, and $\{P_5, P_6\}$. For each group, a job of the first periodic task is randomly selected as the predecessor of a job randomly selected from the second periodic task. Therefore, three pairs of precedence constraints are assigned for each task set. For example, **Figure 4** shows

that in the first group $\{P_1, P_2\}$, the third job T_3 of the first periodic task P_1 is selected as the predecessor of the second job T_7 of the second periodic task P_2 : $T_3 < T_7$. The MFIFO algorithm is evaluated in terms of the two metrics used for the MEDF algorithm: deadline miss rate and energy violation rate.

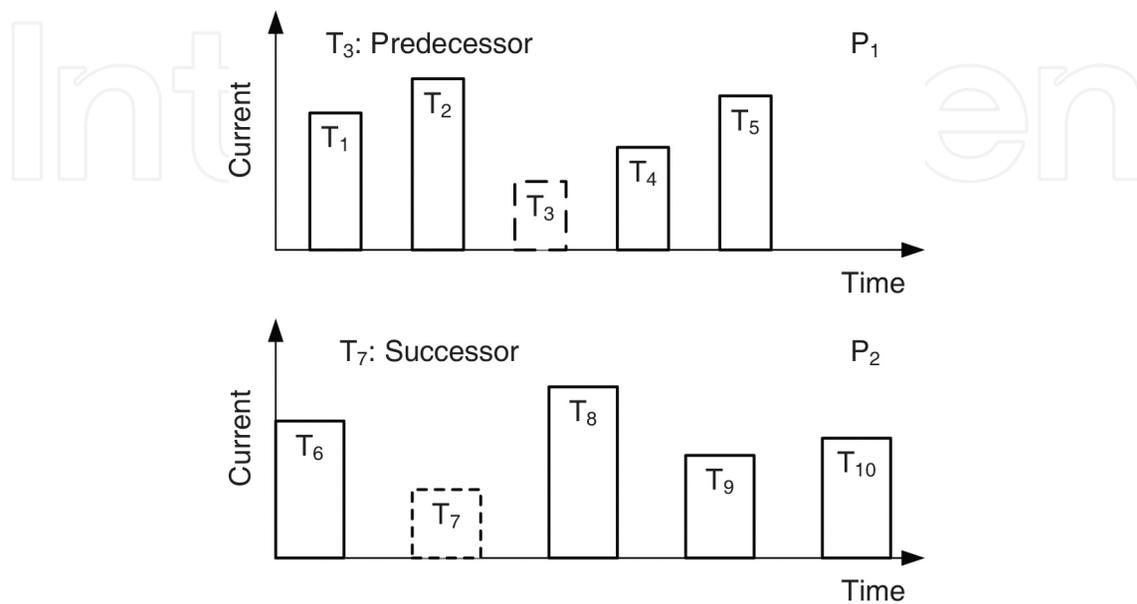


Figure 4. Assignment of precedence constraint.

4.2. An example

An example is used to illustrate the implementation and evaluation of the MFIFO algorithm. The simulation setup is adopted from Ref. [19], which is used to illustrate the MEDF algorithm implementation and evaluation. The supercapacitor initial state is $V_1(t=0) = V_2(t=0) = 1$ V. Two periodic tasks are used to define the task timing and energy constraints. The job T_2 from the first periodic task is selected as the predecessor of the job T_4 from the second periodic task. The precedence constraint is therefore $T_2 < T_4$. The effective release time of task T_4 is $ER_4 = \max(R_4, R_2 + E_2) = 88$ s. The effective release times of the other five tasks are their release times. The task characteristics are listed in Table 1.

	T_1	T_2	T_3	T_4	T_5	T_6
Release time (R_i)	0	80	160	30	130	230
Effective release time (ER_i)	0	80	160	88	130	230
Execution time (E_i)	8	8	8	10	10	10
Deadline (D_i)	80	160	240	130	230	330
Weight (W_i)	35	30	40	42	37	33

Table 1. Tasks with precedence constraints to be scheduled.

The FIFO schedule determined using Algorithm 1 is shown in **Figure 5**. All the tasks are scheduled for execution at their effective release times. The task T_4 begins execution when its predecessor T_2 completes execution. The precedence constraint is satisfied. All the tasks respect their deadlines. The deadline miss rate is therefore $\alpha_{FIFO} = 0$. The supercapacitor terminal voltage profile is shown in **Figure 6**. Two energy violations occur: T_1 and T_5 . The minimum supercapacitor terminal voltages during the executions of the two tasks are 0.9670 and 0.9867 V, respectively. The energy violation rate is therefore $\beta_{FIFO} = 2/6 = 0.333$.

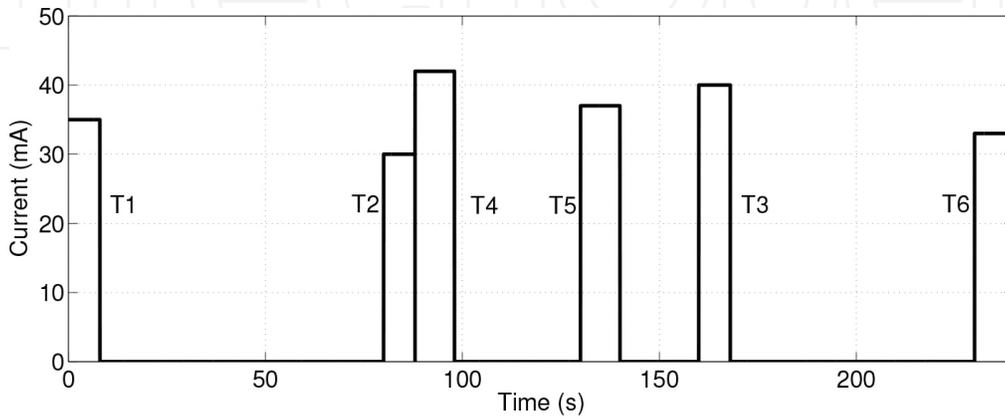


Figure 5. Task schedule determined using FIFO algorithm.

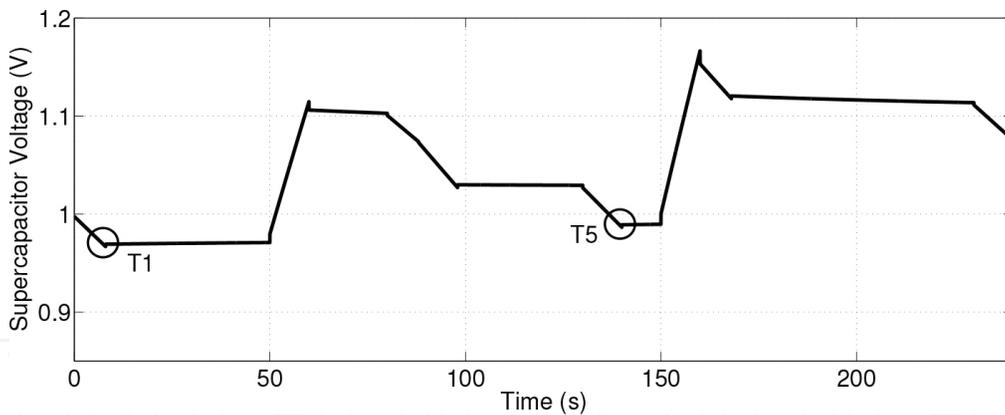


Figure 6. Supercapacitor terminal voltage profile of FIFO schedule.

The task ready time adjustment margin and task ready time offset results are listed in **Tables 2** and **3**, respectively. The task start time is then determined, and the MFIFO schedule is finalized. The MFIFO schedule is shown in **Figure 7**. Tasks T_1 and T_5 are postponed for execution. All the deadlines are respected. Therefore, the deadline miss rate is still $\alpha_{MFIFO} = 0$. The MFIFO and FIFO algorithms have the same timing performance. The supercapacitor terminal voltage profile is shown in **Figure 8**. No energy violation occurs. The energy violation rate is $\beta_{MFIFO} = 0$. This example demonstrates that the MFIFO algorithm is better than the FIFO algorithm in terms of energy performance while maintaining the same timing performance.

FIFO schedule	T ₁	T ₂	T ₄	T ₅	T ₃	T ₆
FIFO schedule ready time (A_i)	0	80	88	130	160	230
Ready time flag ($F_i = A_i - ER_i$)	0	0	0	0	0	0
Maximum delay margin $MM_i = D_i - ER_i - E_i$	72	72	32	90	72	90
Available delay margin ($MA_i = MM_i - F_i$)	72	72	32	90	72	90
End time interval ($I_i = A_{i+1} - (A_i + E_i)$)	72	0	32	20	62	N/A
Ready time adjustment margin $M_i = \min(MA_i, I_i)$	72	0	32	20	62	N/A

Table 2. FIFO schedule and ready time adjustment margin.

FIFO schedule	T ₁	T ₂	T ₄	T ₅	T ₃	T ₆
FIFO schedule ready time (A_i)	0	80	88	130	160	230
Ready time adjustment margin $M_i = \min(MA_i, I_i)$	72	0	32	20	62	N/A
Execution time (E_i)	8	8	10	10	8	10
Latest end time $L_i = A_i + M_i + E_i$	80	88	130	160	230	N/A
$V_1(t = A_i)$	1	1.1005	1.0738	0.8825	1.1539	N/A
$V_2(t = A_i)$	1	1.0247	1.0287	1.0109	1.0352	N/A
If $V_1(t = A_i) > V_2(t = A_i)$	False	True	True	False	True	N/A
If $W_5(A_i < t < L_i) = 0$	False	True	True	False	True	N/A
Ready time offset (Δ_i)	72	0	0	20	0	N/A
MFIFO start time $S_i = A_i + \Delta_i$	72	80	88	150	160	230

Table 3. MFIFO schedule and ready time offset.

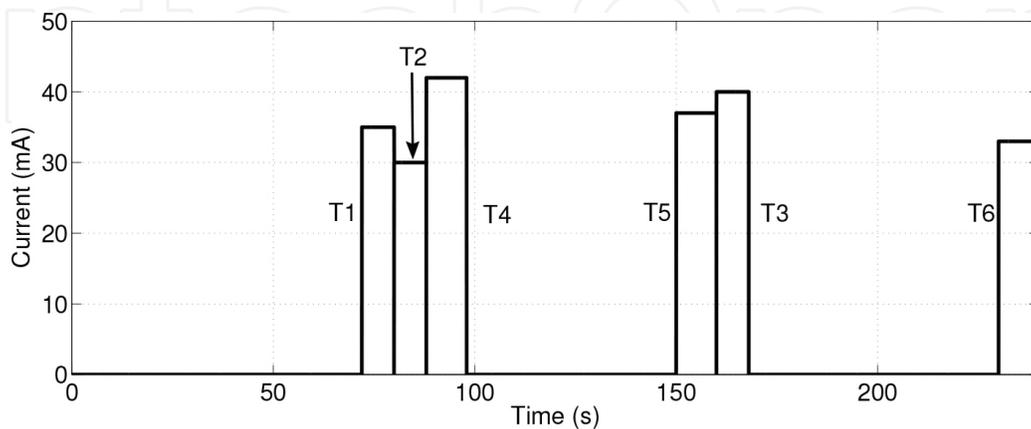


Figure 7. Task schedule determined using MFIFO algorithm.

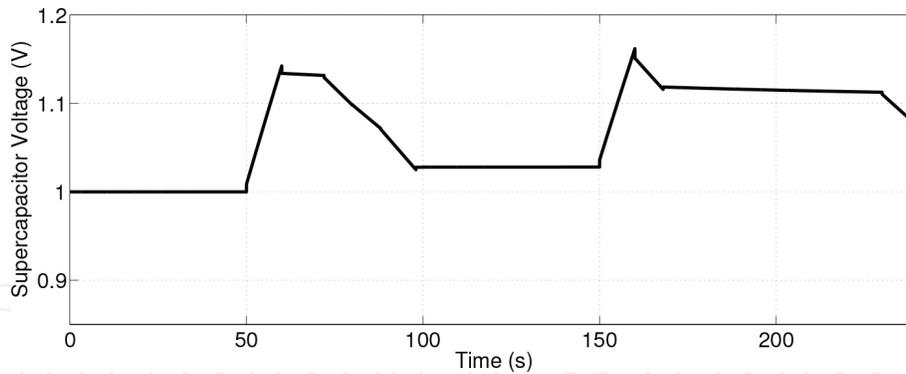


Figure 8. Supercapacitor terminal voltage profile of MFIFO schedule.

4.3. Evaluation results

The simulations are run for 200 times using the setup specified in Ref. [19]. The deadline miss rates and energy violation rates are recorded for the FIFO and MFIFO schedules. The obtained evaluation metrics are sorted in the ascending order and plotted. As shown in **Figure 9**, 35 out of the 200 simulation runs have zero deadline miss rates. The FIFO and MFIFO algorithms always have the same deadline miss rates. The timing and precedence constraints of the FIFO schedules are preserved in the MFIFO schedules. The energy violation rates are shown in **Figure 10**. For the FIFO algorithm, 96 out of the 200 simulation runs have an energy violation rate $\beta_{FIFO} = 1$. Among the 96 runs, the MFIFO schedules have an energy violation rate less than one for seven runs. Among the other 104 runs, the MFIFO algorithm results in an energy violation rate smaller than that of the FIFO algorithm for 81 runs. All together, the MFIFO schedules result in a smaller energy violation rate for 88 runs and a same energy violation rate for 112 runs. The simulation results verify that the MFIFO algorithm improves the energy performance of the FIFO algorithm and maintains the timing performance at the same time.

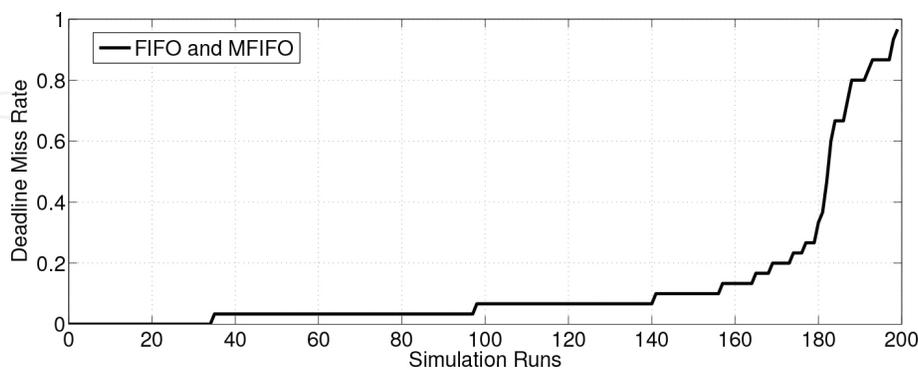


Figure 9. Deadline miss rates of FIFO and MFIFO algorithms.

The simulation setup is slightly modified to quantitatively compare the energy violation rates of the FIFO and MFIFO algorithms. The duty cycles of the six periodic tasks take the same value for each utilization. The utilization is $U = 6 * DC$, where DC is the duty cycle. The duty

cycle increases from 0.02 to 0.1 with a step of 0.02. The utilization is swept from 0.12 to 0.6 with a step of 0.12. The other parameters of the tasks including periods, phases, weights, and precedence constraints are still defined using the setup specified in Ref. [19]. The simulations are run for 30 times for each utilization. **Figure 11** shows the calculated mean absolute percentage error (MAPE) values for the different utilizations. The MAPE decreases as the utilization increases. The average MAPE for the five utilizations is 12.1%. The MFIFO algorithm reduces the average energy violation rate of the FIFO algorithm by 12.1%. The MAPE is 25% for utilization $U = 0.12$. Like the MEDF algorithm, the MFIFO algorithm improves the energy violation rate more significantly if the sensor node operates with a relatively low duty cycle.

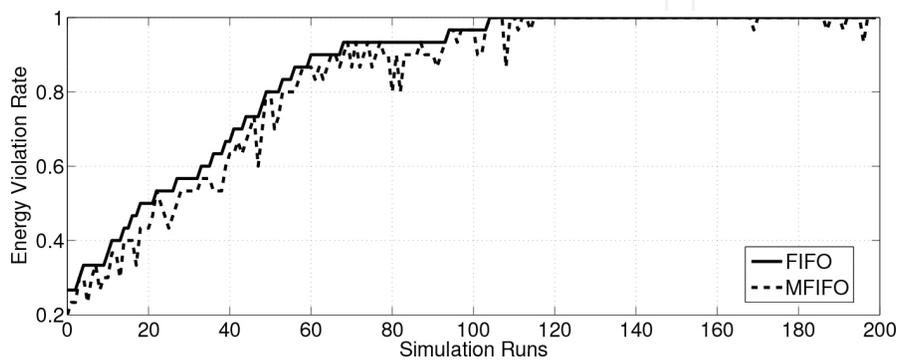


Figure 10. Energy violation rates of FIFO and MFIFO algorithms.

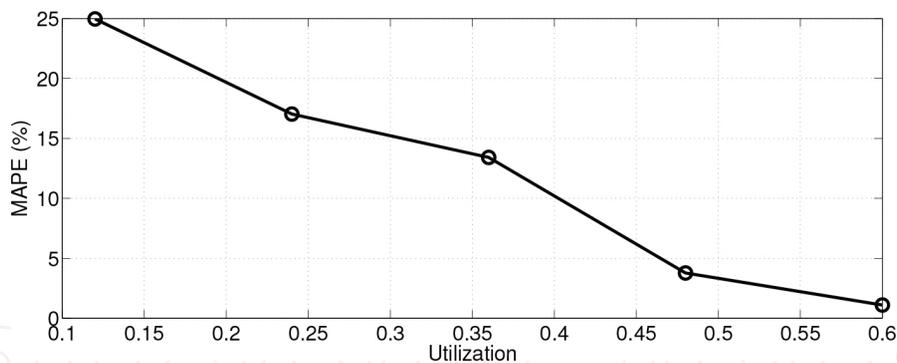


Figure 11. MAPE versus utilization for MFIFO algorithm.

5. Conclusion

This chapter proposes the MFIFO algorithm for nonpreemptable tasks with precedence constraints. The task precedence constraints are transformed into timing constraints by defining the effective release time of a task. The MFIFO algorithm takes into account the energy constraints of tasks in addition to the timing constraints. The MFIFO algorithm is implemented and evaluated. Simulation results show that the MFIFO algorithm improves the energy performance of the FIFO algorithm and maintains the timing performance at the same time.

Author details

Hengzhao Yang^{1*} and Ying Zhang²

*Address all correspondence to: hengzhao.yang@csulb.edu

1 Department of Electrical Engineering, California State University Long Beach, Long Beach, CA, USA

2 School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA

References

- [1] S. Sudevalayam, P. Kulkarni, Energy harvesting sensor nodes: Survey and implications, *IEEE Communications Surveys & Tutorials* 13 (3) (2011) 443–461.
- [2] C. Alippi, C. Galperti, An adaptive system for optimal solar energy harvesting in wireless sensor network nodes, *IEEE Transactions on Circuits and Systems I, Regular Papers* 55 (6) (2008) 1742–1750.
- [3] T. Zhu, Z. Zhong, Y. Gu, T. He, Z.-L. Zhang, Leakage-aware energy synchronization for wireless sensor networks, in: *MobiSys '09*, 2009, pp. 319–332.
- [4] D. Brunelli, C. Moser, L. Thiele, L. Benini, Design of a solar-harvesting circuit for batteryless embedded systems, *IEEE Transactions on Circuits and Systems I: Regular Papers* 56 (11) (2009) 2519–2528.
- [5] F. Simjee, P. H. Chou, Everlast: Long-life, supercapacitor-operated wireless sensor node, in: *ISLPED '06*, 2006, pp. 197–202.
- [6] X. Jiang, J. Polastre, D. Culler, Perpetual environmentally powered sensor networks, in: *IPSN '05*, 2005, pp. 463–468.
- [7] H. Yang, Y. Zhang, Modeling and analysis of hybrid energy storage systems for wireless sensor networks, in: *Proceedings of SPIE*, Vol. 7647, 2010, pp. 76472U:1–76472U:10.
- [8] B. E. Conway, W. Pell, T.-C. Liu, Diagnostic analyses for mechanisms of self-discharge of electrochemical capacitors and batteries, *Journal of Power Sources* 65 (1997) 53–59.
- [9] Y. Diab, P. Venet, H. Gualous, G. Rojat, Self-discharge characterization and modeling of electrochemical capacitor used for power electronics applications, *IEEE Transactions on Power Electronics* 24 (2009) 510–517.
- [10] Y. Zhang, H. Yang, Modeling and characterization of supercapacitors for wireless sensor network applications, *Journal of Power Sources* 196 (8) (2011) 4128–4135.

- [11] H. Yang, Y. Zhang, Self-discharge analysis and characterization of supercapacitors for environmentally powered wireless sensor network applications, *Journal of Power Sources* 196 (20) (2011) 8866–8873.
- [12] H. Yang, Y. Zhang, Evaluation of supercapacitor models for wireless sensor network applications, in: *Proceedings of the 5th International Conference on Signal Processing and Communication Systems*, 2011, pp. 1–6.
- [13] H. Yang, Y. Zhang, Modeling and analysis of a solar powered wireless sensor node, in: *Proceedings of the 2012 International Conference on Computing, Networking and Communications*, 2012, pp. 970–974.
- [14] H. Yang, Task scheduling in supercapacitor based environmentally powered wireless sensor nodes, Ph.D. thesis, Georgia Institute of Technology, 2013 Atlanta, GA, USA.
- [15] M. Kaus, J. Kowal, D. U. Sauer, Modelling the effects of charge redistribution during self-discharge of supercapacitors, *Electrochimica Acta* 55 (25) (2010) 7516–7523.
- [16] J. W. Graydon, M. Panjehshahi, D. W. Kirk, Charge redistribution and ionic mobility in the micropores of supercapacitors, *Journal of Power Sources* 245 (2014) 822–829.
- [17] H. Yang, Y. Zhang, Analysis of supercapacitor energy loss for power management in environmentally powered wireless sensor nodes, *IEEE Transactions on Power Electronics* 28 (11) (2013) 5391–5403.
- [18] H. Yang, Y. Zhang, A study of supercapacitor charge redistribution for applications in environmentally powered wireless sensor nodes, *Journal of Power Sources* 273 (2015) 223–236.
- [19] H. Yang, Y. Zhang, A task scheduling algorithm based on supercapacitor charge redistribution and energy harvesting for wireless sensor nodes, *Journal of Energy Storage* 6 (2016) 186–194.
- [20] J. W. S. Liu, *Real-Time Systems*, 1st Edition, Prentice Hall, 2000 Upper Saddle River, NJ, USA.

IntechOpen

