Available Power Analysis for Background Tasks on Ubiquitous Sensor and Actuator Networks

Junghoon Lee[1], Cheol Min Kim[2], Ho-Young Kwak[3], and Jikwang Han[4]

[1] Dept. of Computer Science and Statistics, [2] Dept. of Computer Education
[3] Dept. of Computer Engineering, Jeju National University,
690-756, Jeju-Do, Republic of Korea
[4] Jinwoo Soft Innovation
{jhlee,cmkim,kwak}@jejunu.ac.kr, hmurdoc@jinwooso.co.kr

Abstract. This paper first presents the sensor data processing framework integrating a power consumption scheduler for control actions, and then analyzes the available power for background tasks which are continuously performed using the remaining power. As the scheduler is designed to reduce the peak load for the given task set, it can make flatter the consumption level of dynamic tasks activated by the event detection. The experiment result obtained from the prototype implementation reveals that the proposed scheme can improve the minimum available power by up to 17.9 %, compared with the *Earliest* scheduling scheme, having a smaller standard deviation over the slots included in a scheduling window. Hence, our framework can achieve cost savings as the sensor network can purchase a smaller amount of energy from a utility company and can be built with lower power transmission capability.

Keywords: smart grid, power consumption control, peak reduction, background task, available power.

1 Introduction

[1] Nowadays, the ubiquitous sensor network, or USN in short, have been widely employed for environmental and wildlife habitat monitoring, particularly improving productivity and revenue in the agricultural and livestock farms [1]. The main task of USNs is monitoring sensor values, deciding the control actions, and triggering appropriate actuators. For example, if lightness drops below the permissible level, USN turns on the light in the green house. In addition, if the current CO_2 level is detected to be higher than a specific bound, a ventilator is activated to refresh the air.

The control action necessarily leads to power consumption, and thus power management is becoming an important component in the USN. In case a set of control actions are triggered in a short time window, the power consumption

[1] This research was supported by the MKE, Korea, under IT/SW Creative research program supervised by the NIPA (NIPA-2011-(C1820-1101-0002)) and also through the project of Region technical renovation, Republic of Korea.

D. Zeng (Ed.): Advances in Electrical Engineering & Electrical Machines, LNEE 134, pp. 141–146.
springerlink.com

increases unexpectedly, while the USN management purchases the energy and installs the transmission cable according to such peak load. Peak load reduction is essentially advantageous in terms of economic and efficiency aspects, and to this end, the power consumption scheduler can reshape the load by relocating some operation in the low-load interval, taking into account the time constraint of each control action. Here, each control action on a electric device or facility can be modeled as a control task and scheduled.

However, there are not just the tasks dynamically activated from external event detection. The USN has background tasks such as sensor wakeup time management, periodic watering, air conditioning, and electric device charging. Those tasks must be performed continuously as long as there is available power surplus, adapting its operation level according to the available power. For such a background task, the stable power availability is desirable and peak load reduction can give more power to those tasks. In this regard, this paper is to analyze the available power for background tasks according to a power consumption scheduling strategy on our agricultural USN framework.

2 Background and Related Work

Under the research and technical project named *Development of convergence techniques for agriculture, fisheries, and livestock industries based on the ubiquitous sensor networks*, our project team has developed an intelligent USN framework, which provides an efficient and seamless runtime environment for a variety of monitor-and-control applications on sensor networks [2]. Over the sensor network mainly exploiting the Zigbee technology, composite sensors detect a change of a target livestock via the biosensors as well as monitor humidity, CO_2, and NH_3 level via the environmental sensors. In addition, an agent-based architecture forwards real-time sensor stream to the appropriate application.

In USN, a lot of sensors are installed over the wide area and each of them reports its sensor values to the controller, creating a tremendous amount of data records. Thus, the USN must systematically handle and analyze the large volume of sensor records [3]. Figure 1 illustrates our data processing framework which consists of data mining module, data manager, activation control process, and power scheduler. Here, more than one sensor may capture the same event, there can be lots of duplicated events. Moreover, sensor values can have garbage and measurement errors due to many reasons such as network disconnection. Wrong reaction stemmed from wrong data analysis can lead to many hazardous results such as burning pump motors. Based on the predefined event detection logic specific to the agricultural sensor network, the data mining module removes the data duplication and invalid values.

The inference engine defines a set of rules to describe and detect events. First, each sensor and node is assigned a unique identifier, while $max()$, $min()$, $average()$, $count()$, and $run()$ functions are provided for better event specification. Using this, we can specify several rules, for example, report an event when average temperature of node 123 is higher than 35 °C, or turn on all fans installed in sensor node 452. Now, the middleware checks the validity of the sensor data and requests the retransmission if it has an error term. After calculating the difference from the previous sensor reading, the middleware detects an abnormal condition based on the empirically obtained event patterns and knowledge [4].

After the sensor data analysis, a series of control actions are determined. Each action has its own time constraint that it must be completed within a specific time instant and power requirement profile which is a sequence of power value consumed on each time interval. As shown in Figure 1, the actuator control process notifies the power scheduler of those tasks. The power scheduler then generates a new schedule or modifies an existing schedule based on the power consumption profile as well as pending background tasks [5]. According to the schedule, power control unit turns or off the power switch which connects the power to an electric device.

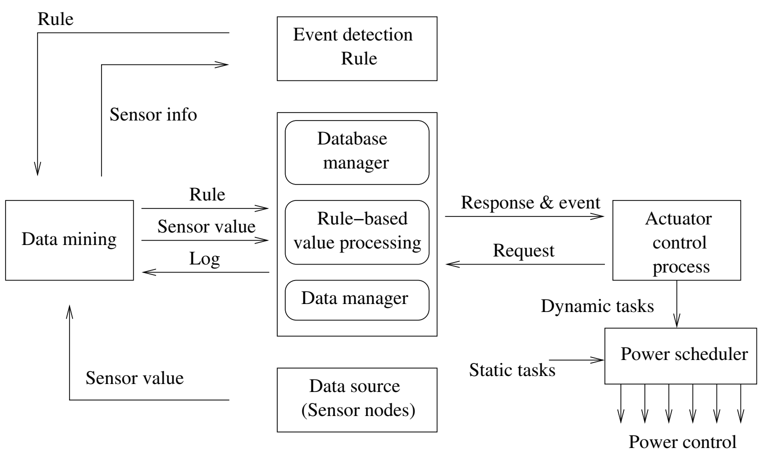


Fig. 1. Sensor and actuator network architecture

3 Analysis Result

3.1 Scheduling Policy

Our previous work has designed a power management scheme for smart homes or farms, aiming at reducing the peak power consumption [6]. It finds an optimal schedule for the task set consisting of nonpreemptive and preemptive tasks, each of which has its own consumption profile. To compensate for the intolerable scheduling time on the large number of tasks and slots, two speed enhancement techniques are employed. First, for a nonpreemptive task, the profile entries are linearly copied into the allocation table without intermittence. Second, for the preemptive task, the feasible combinations are generated in advance of search space expansion based on the operation time and deadline of a task.

Then, the scheduler maps the combination to the allocation table at each space expansion stage.

The experiment sets the schedule length, namely, M, to 20 time units. If one time unit is equal to 20 min, the total schedule length will be 6.6 hours. For a task, the start time is selected randomly between 0 and M, while the operation length is also selected randomly, but it will be set to M if the finish time, namely, the sum of start time and the operation length, exceeds M. All tasks have the common deadline, namely, M, considering the situation a customer orders a set of tasks to be done before he or she returns home, but this restriction can be eliminated. In addition, the power level for each time slot has the value of 1 through 5, the power scale such as w or kw, not being explicitly specified as the scale is quite relative according to the task set.

3.2 Available Power Analysis

The Earliest scheduling is selected for the performance comparison as in [6]. This scheme initiates tasks as soon as they get ready and makes it run without preemption. It adopts no control strategy, but it provides a measure for a comparative assessment for power scheduling strategies. In all experiments, for each parameter setting, 10 tasks sets are generated and their results are averaged. We define the available power by the remaining amount of the whole contracted power after allocation to the dynamic tasks. It depends on the total power contracted with a power supplier company. To give a sufficient margin and focus on the availability dynamics, this experiment sets the contracted power to the product of the number of tasks and the maximum power consumption level in a slot.

To begin with, Figure 2 plots per-slot available power for both schemes when the task set has just 5 nonpreemptive tasks. As the task activation time distributes over the whole slots, we can find more available power during the first slots, namely, from Slot 0 to Slot 6. The scheduler tends to put off the task operation near to its deadline, so the available power gets smaller on the slots closer to the common deadline. Figure 2 shows the difference in the available power over 20 slots, and we can allocate the power more stably to the background task when the curve is flatter.

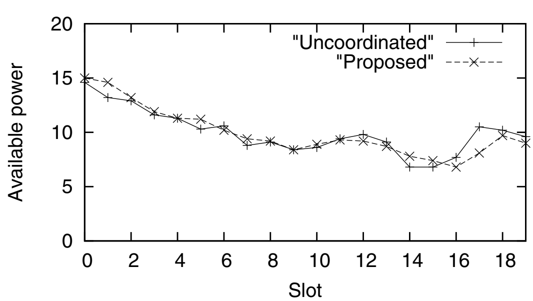
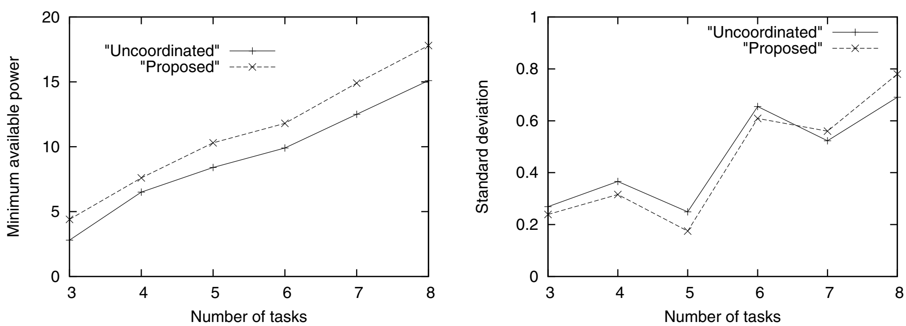


Fig. 2. Per-slot available power

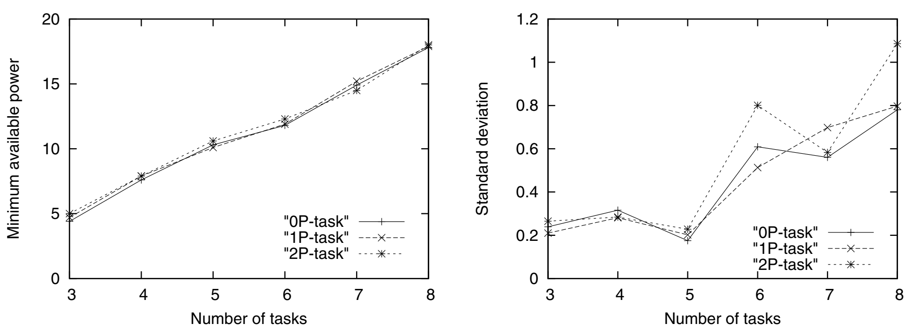
The next experiment measures the effect of the number of tasks to the minimum available power and its standard deviation, while the measurement results are shown in Figure 3. For a given task set, after deciding the task schedule according to the scheduling scheme, the experiment identifies the slot having the lowest available power, and this value is the minimum available power. If it is 0, it is not possible to run any background task during this slot. Hence, this metric indicates how steadily background tasks can be executed without being delayed or even canceled. Figure 3(a) shows that the proposed scheme improves the minimum available power by up to 17.9 % and we can achieve more improvement when the number of tasks gets larger. In addition, Figure 3(b) indicates that the proposed scheme generally has smaller standard deviation, and it reserves stable power for the background task.

Figure 4 measures the effect of the number of preemptive tasks, which can give better flexibility to the power consumption schedule. In this experiment, the number of total tasks is 5 while the number of preemptive tasks ranges from 0 to 2. Actually, the scheduling time increases unmanageably, when the number of



(a) Minimum available power (b) Standard deviation

Fig. 3. Effect of the number of tasks (with no preemptive task)



(a) Minimum available power (b) Standard deviation

Fig. 4. Effect of the number of nonpreemptive tasks

preemptive tasks goes beyond 3, as the search space size extends too vast. Figure 4(a) plots the minimum available power for the above 3 cases. 3 curves have the similar pattern, but for the 2-preemptive-task case, the minimum available power is slightly larger. In addition, Figure 4(b) shows the standard deviation of the available power for 20 slots. Even if the 2-preemptive-task case seems to have unstable power availability, especially when the number of tasks is large, the standard deviation more depends on the task set features such as the arrival time distribution and the slack distribution.

4 Concluding Remarks

This paper first has presented the sensor data processing framework integrating a power consumption scheduler for control actions, while the scheduler finds an optimal schedule capable of reducing the peak load for the given task set. Then, it has analyzed the available power for background tasks which are continuously performed using the power remaining after the allocation to the dynamic task. As the scheduler is designed to reduce the peak load, it can more evenly distribute the power consumption needed by dynamic tasks, steadily providing power to the background task. The experiment result obtained from the prototype implementation reveals that the proposed scheme can improve the minimum available power by up to 17.9 %, compared with the Earliest scheduling scheme. After all, our framework can achieve cost savings as the sensor network can purchase a smaller amount of energy from a utility company.

References

1. Culler, D., Estrin, D., Srivastava, M.: Overview of Sensor Networks. IEEE Computer 37, 41–49 (2004)
2. Lee, J., Park, G., Kwak, H., Kim, C.: Efficient and Extensible Data Processing Framework in Ubiquitous Sensor Networks. In: International Conference on Intelligent Control Systems Engineering, pp. 324–327 (2011)
3. Golab, L., Oszu, M.: Issues in Data Stream Management. ACM SIGMOD Record 32, 5–14 (2003)
4. Ghanem, T., Elmagarmid, A., Larson, P., Aref, W.: Supporting Views in Data Stream Management Systems. ACM Transactions on Database Systems 35(1) (2010)
5. Caron, S., Kesidis, G.: Incentive-Based Energy Consumption Scheduling Algorithms for the Smart Grid. In: IEEE SmartGridComm. (2010)
6. Lee, J., Park, G., Kim, S., Kim, H., Sung, C.: Power Consumption Scheduling for Peak Load Reduction in Smart Grid Homes. In: ACM Symposium on Applied Computing, pp. 584–588 (2011)