

A Low Power Real-Time Scheduling Scheme for the Wireless Sensor Network^{*}

Mikyung Kang¹ and Junghoon Lee^{2,**}

¹ Information Sciences Institute East, University of Southern California

² Dept. of Computer Science and Statistics, Cheju National University
mkkang@east.isi.edu, jhlee@cheju.ac.kr

Abstract. This paper proposes a low power real-time scheduling scheme that minimizes power consumption and reassigns polling order on the wireless sensor networks. The proposed scheme aims at enhancing the power performance based on the maximum constant power method along with the additional auxiliary schedule which can switch the primary one when a specific channel falls in a bad state or time slot is unused. Built on top of the EDF scheduling policy, the proposed scheme also tests whether an unused slot can be reclaimed without violating the constraints of subsequent real-time streams. To improve the probability of reclaimed bandwidth, it also rearranges the polling order. The simulation results show that the proposed scheme is able to enhance success ratio and it also provides minimized power consumption.

Keywords: Real-time scheduling, low power scheme, EDF policy, reclaiming scheme, wireless sensor network.

1 Introduction

Battery-powered portable systems have been widely used in many applications, such as mobile computing, wireless communications, information appliances, wearable computing as well as various industrial and military applications. As systems become more complex and incorporate more functionality, they become more power-hungry[1]. Thus, reducing energy consumption and extending battery lifespan have become a critical aspect of designing battery-powered systems. Low energy consumption is today an increasingly important design requirement for digital systems, with impact on operating time, on system cost, and, of no lesser importance, on the environment.

Reducing power and energy dissipation has long been addressed by several research groups, at different abstraction levels[2]. As a consequence, reduction of the power consumption of processors, especially in OS (Operating System) level, is important for the power-efficient design of such systems. As a method

* This research was supported by the MIC, Korea, under the ITRC support program supervised by the IITA (IITA-2006-C1090-0603-0040).

** The corresponding author.

to reduce power consumption of processors in OS level, VSP (Variable Speed Processor) can change its speed by varying the clock frequency along with the supply voltage when the required performance on the processor is lower than the maximum.

Wireless sensor networks use battery-operated computing and sensing devices[3]. Sensor network nodes have very limited battery life; moreover once deployed, a sensor network may be left unattended for its entire operational lifetime. This is due to the fact that sensor networks may be deployed in wide, remote, inaccessible areas. The energy-constrained nature of sensor networks calls for protocols that have energy efficiency as a primary design goal. These characteristics of sensor networks and applications motivate a MAC (Medium Access Control) in almost every way: energy conservation and self-configuration are primary goals, while per-node fairness and latency are less important.

EDF (Earliest Deadline First), which dynamically picks a message that has the closest deadline among the pending ones, is known to be the most common scheduling policy for real-time message streams[4]. WLAN can also exploit EDF for message scheduling in various ways on the assumption that the communication pattern of sensory data is fixed and known *in priori*. However, it is not clear how well this algorithms will work for the wireless sensor networks, since wireless channels are subject to unpredictable location-dependent and bursty errors, which make a real-time application fail to send or receive some of its real-time packets[5]. Also, the design of energy-efficient protocols has become an area of intense research. This paper addresses these problems based on the maximum constant power method along with the additional auxiliary schedule which can switch the primary one when a specific channel falls in a bad state or time slot is unused.

The rest of this paper is organized as follows: After issuing the problem in Section 1, Section 2 introduces some related works, Section 3 explains background and basic assumptions. Section 4 describes the proposed low power real-time scheduling scheme in detail, and Section 5 shows the result of performance measurement. Finally, Section 6 summarizes and concludes this paper.

2 Related Works

This section introduces previous works regarding power-aware MAC protocols. PAMAS (Power Aware Multi-Access Protocol with Signalling for Ad Hoc Networks) is one of the earliest contention-based protocols to address power efficiency in channel access[6]. It saves energy by attempting to avoid over-hearing among neighboring nodes. To achieve this, PAMAS uses out-of channel signaling.

The IEEE 802.11 DCF is a contention-based protocol based on CSMA/CA (Carrier Sensing Multiple Access with Collision Avoidance)[7]. IEEE 802.11 performs both physical and virtual carrier sensing. Virtual carrier sensing is achieved by sending information about duration of each frame in the headers which is used by stations as an indication of how long the channel will be busy. After this time is elapsed, stations can sense the channel again. In order to solve the

“hidden terminal” problem and avoid data frame collisions, the RTS-CTS handshake is used. Two power management mechanisms are supported: *active* and *power-saving* (PS).

The Sensor MAC protocol, or S-MAC, was developed with power savings as one of its design goals[3]. It also falls into the contention-based protocol category but achieves energy efficiency by making use of low-power radio mode. Nodes alternate between periodic sleep and listen periods. Listen periods are split into synchronization and data periods. During synchronization periods, nodes broadcast their sleeping schedule, and, based on the information received from neighbors, they adjust their schedule so that they all sleep at the same time. During data periods, a node with data to send will contend for the medium (RTS-CTS exchange). Unlike PAMAS, it only uses in-channel signaling. Finally, S-MAC applies message passing to reduce contention latency for sensor-network applications.

T-MAC, a contention-based MAC protocol for wireless sensor network, is another notable example[8]. Applications for these networks have some characteristics that can be exploited to reduce energy consumption by introducing an active/sleep duty cycle. To handle load variations in time and location T-MAC introduces an *adaptive* duty cycle by dynamically ending the active part of it. This reduces the amount of energy wasted on idle listening, in which nodes wait for potentially incoming messages, while still maintaining a reasonable throughput.

TRAMA (TRaffic-Adaptive Medium Access) protocol is introduced for energy-efficient collision-free channel access in wireless sensor networks[9]. It reduces energy consumption by ensuring that unicast, multicast, and broadcast transmissions have no collisions and by allowing nodes to switch to a low-power, idle state whenever they are not transmitting or receiving. TRAMA assumes that time is slotted and uses a distributed election scheme based on information about the traffic at each node to determine which node can transmit at a particular time slot.

3 Background and Basic Assumptions

3.1 WLAN

WLAN divides its time axis into CFP (Contention Free Period) and CP (Contention Period), which are mapped into PCF (Point Coordination Function) and DCF (Distributed Coordination Function), respectively. To provide the deterministic access to each node during CFP, AP (Access Point) polls each node according to the predefined order, and only the polled node can transmit its frame. In the DCF interval, every node including AP contends the medium via the CSMA/CA protocol. The AP periodically initiates CFP by broadcasting a *Beacon* frame that has the precedence in transmission via shorter IFS (Inter-Frame Space).

Each station is associated with a channel which has either of two states, namely, *error state* and *error-free state* at any time instant. A channel is defined between each mobile and the AP, and it can be modeled as a Gilbert channel[10]. We can denote the transition probability from state *good* to state *bad* by p

and the probability from state *bad* to state *good* by q . The pair of p and q representing a range of channel conditions has been obtained by using the trace-based channel estimation. The average error probability and the average length of a burst of errors are derived as $\frac{p}{p+q}$ and $\frac{1}{q}$, respectively. A packet is received correctly if the channel remains in state *good* for the whole duration of packet transmission. Otherwise, it is received in error. Channels between the AP and respective stations are independent of one another in their error characteristics.

3.2 DVS Algorithms

DVS (Dynamic Voltage Scaling), which adjusts the supply voltage and clock frequency dynamically, is an effective low-power design technique for embedded real-time systems[11]. Since the energy consumption E of CMOS circuits has a quadratic dependency on the supply voltage, lowering the supply voltage is one of the most effective ways of reducing the energy consumption.

$$E \propto C_L \cdot N_{cycle} \cdot V_{DD}^2, \quad (1)$$

where E , C_L , N_{cycle} and V_{DD} are dynamic power consumption, load capacitance of CMOS circuit, the number of executed cycle, and supply voltage, respectively.

For hard real-time systems, there are two types of voltage scheduling approaches depending on the voltage scaling granularity: intra-task DVS (IntraDVS) and inter-task DVS (InterDVS) [2]. The intra-task DVS algorithms adjust the voltage within an individual task boundary, while inter-task DVS algorithms determine the voltage on a task-by-task basis at each scheduling point. Based on inter-task DVS algorithms for periodic hard real-time systems, slack estimation techniques are as follows[11] : Based on the periodicity and WCET (Worst Case Execution Time) of tasks, we can estimate statically given slack times and exploit those slack times to lower the clock speed, i.e., the worst case processor utilization can be estimated and the clock speed can be adjusted based on that as in the maximum constant speed heuristic. Since the arrival time of tasks are known a priori, when a single task is active, its execution can be extended to the earliest time of the next task with the lowered clock speed and voltage as in the stretching to NTA (Next Task Arrival) technique[1].

While the above techniques exploit static information, other techniques utilize dynamic information such as workload variation of tasks. The execution time of each task is usually less than its WCET, and the actual processor utilization during run time is usually lower than the worst case processor utilization. Thus, when a task completes its execution much earlier than its WCET, the expected utilization can be recalculated based on the actual execution time of completed task, and the clock speed can be adjusted based on that as in utilization updating. Also, when a higher-priority task completes its execution earlier than its WCET, the following lower-priority tasks can use the slack time, and the clock speed can be lowered based on the slack time as in priority-based slack stealing[12].

4 Proposed Scheduling Scheme

4.1 System Model

This paper exploits the contention-free TDMA style access policy as in [5], for the real-time guarantee, as the contention resolution via packet collisions consumes the precious communication energy. As the real-time guarantee cannot be provided without developing a deterministic access schedule, the network time is divided into a series of equally sized slots to eliminate the unpredictability stemmed from access contention. Accordingly, the allocation scheme assigns each slot to real-time streams so as to meet their time constraints. The slot equals the basic unit of wireless data transmission and the other non-real-time traffic is also segmented to fit the slot size. Therefore, a preemption occurs only at the slot boundary. This network access can be implemented by making AP poll each station according to the predefined schedule during the CFP.

The traffic of sensory data is typically *isochronous* (or synchronous), consisting of message streams that are generated by their sources on a continuing basis and delivered to their respective destinations also on a continuing basis[4]. This paper follows the general real-time message model which has n streams, namely, S_1, S_2, \dots, S_n , and each S_i generates a message less than WCET C_i at each beginning of its period P_i . Each packet must be delivered to its destination within D_i time unit from its generation or arrival at the source, otherwise, the packet is considered to be lost. Generally, D_i coincides with P_i to make the transmission complete before the generation of the next message. A task set is called feasible if deadline of each task is satisfied at all times.

4.2 Scheduling Scheme

Assume all tasks are released simultaneously at time 0. A typical EDF schedule, which assumes that tasks run at their WCETs (C_i), is shown in Fig. 1.(a). The IC of RF module can be manufactured by using CMOS technology. If the power of the RF module is lowered by half or if the RF module with half performance is used meaning that C_i is doubled, the EDF schedule becomes as shown in Fig. 1.(b). For example, if stream C were to take a little longer to complete, stream A would miss its deadline at time 24. When some task instances are completed earlier than their WCETs, there are more idle intervals as shown in Fig. 1.(c).

For dynamic-priority scheduling, a task set is feasible if and only if the utilization, U , is less than or equal to 1. Thus, *power scaling factor*, η , is straightforward to compute because it is equal to the utilization, given by $\eta = \sum \frac{C_i}{P_i} = \frac{1}{6} + \frac{1}{8} + \frac{2}{12} = 0.458$. After all, the maximum constant power(V) can be lowered up to 0.458% and Fig. 1 shows the case whose maximum power is $\frac{V}{2}$. To reduce the power consumption considering error characteristics, this paper proposes an auxiliary schedule, X_{1t} , as shown in Fig. 1.(b). After power scaling, the auxiliary schedule helps the runtime scheduler to pick alternative stream according to the auxiliary schedule as shown in Fig. 2.(a), when it cannot poll the stream in the primary schedule. To build an auxiliary schedule, AP generates both arrival time

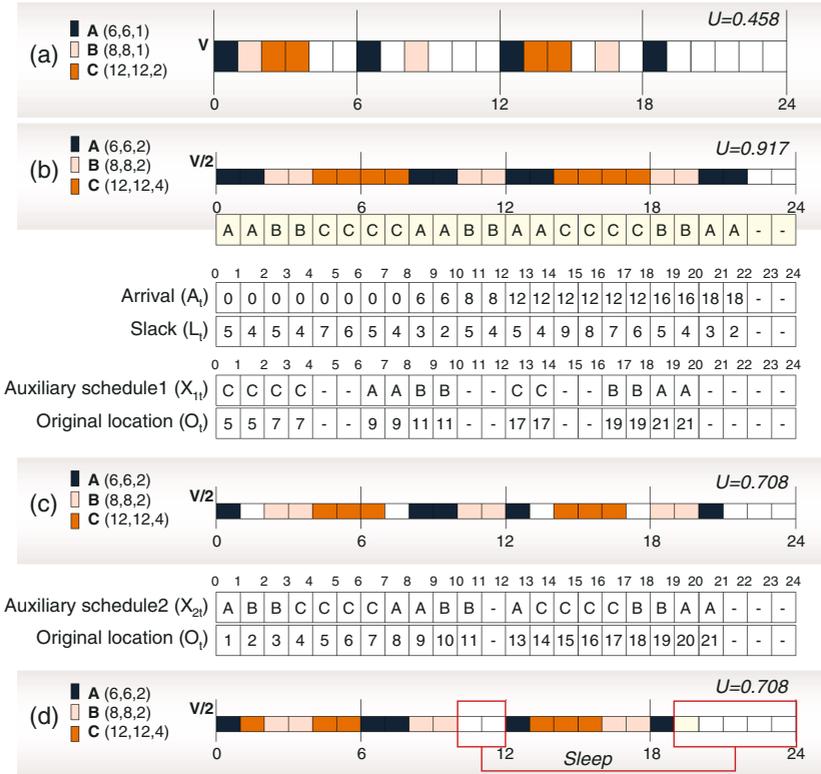


Fig. 1. A schedule for the example task set

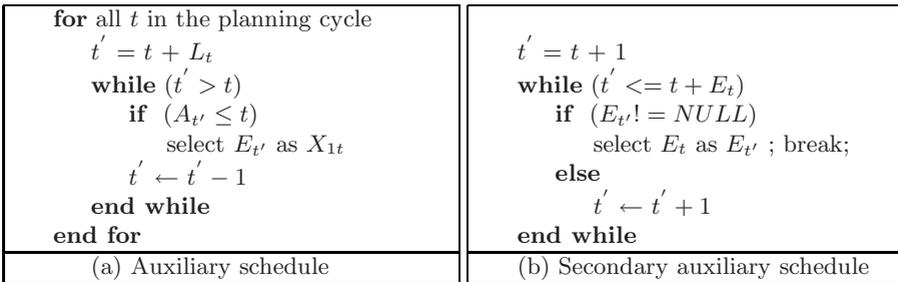


Fig. 2. Auxiliary schedule

and slack time of each slot in the primary schedule based on EDF policy. By E_t , we denote the stream AP will poll at time t . If E_t is M , A_t and L_t denote the arrival time and the slack time of M , respectively. Slack means the redundant time to the completion of a given task or message. If the status of E_t is in state *bad* or E_t is empty, the proposed scheme selects auxiliary schedule (X_{1t}). None the less, if time slot is still empty, this scheme also selects secondary auxiliary

schedule (X_{2t}) as shown in Fig. 2.(b). Fig. 1.(d) shows the final allocation. In the result, the power consumption is reduced by minimizing the maximum power and the time of continuous power-down(sleep) mode is increased by collecting unused slots into one side.

5 Performance Measurement

This section measures the performance of the proposed scheduling scheme in terms of the success ratio according to the packet error rate as well as the ratio of BCET (Best Case Execution Time) to WCET using the event simulator. The success ratio means the ratio of the number of successfully transmitted packets within their deadlines to the total number of generated packets. Because the statistics of the actual execution times of instances of the tasks are not available, it is assumed that the execution time of each instance of a task is drawn from a random Gaussian distribution. Then, the BCET is varied from 10% to 100% of the WCET for each task. For the target stream sets, we fixed the length of a planning cycle to 24 as well as the number of streams to 3, and generated every possible stream sets whose utilization ranges from 0.2 to 1.0. We measured how many errors can be recovered and power-down mode can be allocated by the auxiliary polling schedule, for the packet error rate of 0.0 through 0.5. In MPEDF policy, a slack time of a task is estimated using the maximum constant power and stretching-to-NTA methods[1].

Fig. 3 plots the measured result and compares the success ratio. As shown in this figure, the performance gap gets larger with a higher packet error rate, by up to 8.6 %. The proposedEDF scheme can widen the gap between those curves, that is, considerably relieves the problem of error-prone WLAN and poor utilization of PCF operation. Fig. 4 plots the power consumption ratio according to the ratio of BCET/WCET. In the experiment, power consumption variable is aligned to the non-power-down mode(EDF). It is certain that the improvement gets larger as the error rate increases and BCET/WCET rate decreases. This is due to the fact that the recovered portion gets larger and the probability of power-down increases.

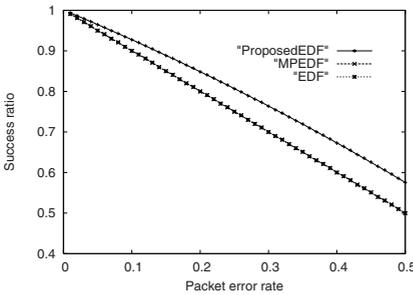


Fig. 3. Effect of error rate

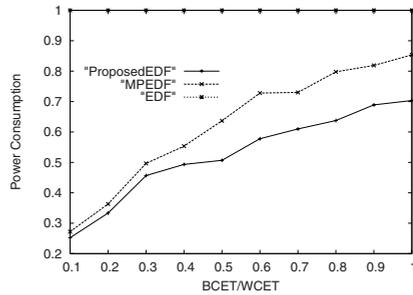


Fig. 4. Effect of BCET/WCET rate

6 Conclusion

In this paper, we have proposed a low power real-time scheduling scheme that minimizes power consumption and reassigns polling order on the wireless sensor networks. The proposed scheme aims at enhancing the power performance based on the maximum constant power method along with the additional auxiliary schedule which can switch the primary one when a specific channel falls in a bad state or time slot is unused. The simulation results show that the proposed scheme is able to enhance success ratio and it also provides minimized power consumption. Finally, we are to apply the power management scheme proposed in this paper to the RM style polling framework combined with an error control mechanism.

References

1. Shin, Y., Choi, K., Sakurai, T.: Power optimization of real-time embedded systems on variable speed processors. Proc. of the International Conference on Computer-Aided Design (2000) 365-368
2. Gruian, F.: Hard Real-Time Scheduling for Low-Energy Using Stochastic Data and DVS Processors. Proc. of the International Symposium on Low Power Electronics and Design (2001) 46-51
3. Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: Energy-Efficient Communication Protocol for Wireless Microsensor Networks. Pro. of the 33rd Hawaii International Conference on System Sciences (2000)
4. Liu J.: Real-Time Systems. Prentice Hall (2000)
5. Adamou, M., Khanna, S., Lee, I., Shin, I., Zhou, S.: Fair real-time traffic scheduling over a wireless LAN. Proc. IEEE Real-Time Systems Symposium (2001) 279-288
6. Singh, S., Raghavendra, C.: PAMAS : Power Aware Multi-Access protocol with Signalling for Ad Hoc Networks. ACM Computer Communications (1999)
7. LAN MAN Standards Committee of the IEEE Computer Society: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE std 802.11-1999 edition (1999) also available at <http://standards.ieee.org/getieee802/download/802.11-1999.pdf>.
8. Dam, T.V., Langendoen, K.: An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks. ACM Sensys '03 (2003) 171-180
9. Rajendran, V., Obraczka, K., Garcia-Luna-Aceves, J.J.: Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks. ACM Sensys '03 (2003)
10. Shah, S., Chen, K., Nahrstedt, K.: Dynamic bandwidth management for single-hop ad hoc wireless networks. ACM/Kluwer Mobile Networks and Applications (MONET) Journal. 10 (2005) 199-217
11. Kim, W., Shin, D., Yun, H., Kim, J., Min, S.: Performance Evaluation of Dynamic Voltage Scaling Algorithms for Hard Real-Time Systems. J. Low Power Electronics. Vol. 1 (2005) 1-11
12. Aydin, H., Melhem, R., Mosse, D., Alvarez, P.M.: Dynamic and aggressive scheduling techniques for power-aware real-time systems. Proc. of IEEE Real-Time Systems Symposium (2001) 95-105