G Model
SUSCOM-143; No. of Pages 13
ARTICLE IN PRESS
Sustainable Computing: Informatics and Systems xxx (2016) xxx–xxx

Contents lists available at ScienceDirect

Sustainable Computing: Informatics and Systems

journal homepage: www.elsevier.com/locate/suscom

On minimizing expected energy usage of embedded wireless systems with probabilistic workloads

Maryam Bandari, Robert Simon, Hakan Aydin*

George Mason University, Computer Science Department, 4400 University Drive, Fairfax, VA 22030, USA

A R T I C L E I N F O

Article history:
Received 19 May 2015
Accepted 28 February 2016
Available online xxx

Keywords:
Networked embedded systems
Energy management
Dynamic Modulation Scaling
Dynamic Voltage Scaling

A B S T R A C T

A large number of embedded wireless systems must handle complex and time-varying computational and communication workloads. Further, a significant number of these systems support real-time applications. Most of the existing energy management studies for such systems have focused on relatively simple scenarios that assume deterministic workloads, and only consider a limited range of energy management techniques, such as Dynamic Voltage Scaling (DVS). Our paper addresses these deficiencies by proposing a general purpose probabilistic workload model for computation and communication. To account for the importance of radio energy consumption, we also analyse Dynamic Modulation Scaling (DMS), an often overlooked method for energy management. We define several energy control algorithms, including an optimal combined DVS–DMS approach, and evaluate these algorithms under a wide range of workload values and hardware settings. Our results illustrate the benefits of joint power control algorithms.

1. Introduction

A large class of embedded wireless systems have real-time performance requirements for both computational and communication tasks. Examples of such systems include industrial process control, highway monitoring and building surveillance [1–3]. Many of these systems are self-powered, so from both a system design and an environmental perspective efficient energy management is of paramount importance. System architects use component level tuning knobs that tradeoff power consumption with performance. For instance, a commonly used power saving technique is Dynamic Voltage Scaling (DVS) [4]. DVS controls power consumption by reducing the CPU frequency and supply voltage, thereby saving energy expenditure while requiring computations to take longer. Dynamic Modulation Scaling (DMS) is another type of tuning technique. DMS works by changing radio modulation levels and constellation sizes, reducing energy expenditures while requiring longer transmission and reception times [5]. DMS is directly supported by embedded wireless standards such as 802.15.4 [6]. The impact of DMS usage on power consumption in wireless embedded systems is relatively understudied. Moreover, for wireless embedded nodes with both substantial computational and communication workloads, both DVS and DMS techniques are relevant. Though there are a few studies that consider DVS and DMS simultaneously [7,8], those works consider exclusively deterministic workloads.

This paper addresses that gap by studying the joint use of DVS and DMS for real-time embedded wireless systems through the design of several novel energy management algorithms. We focus on systems that have deadline constraints for both computational and communication tasks. We are specifically interested in quantifying the impact of these algorithms when both computation and communication workloads are known only probabilistically. We believe this is a direction that warrants investigation, as in practical applications the most important objective is typically to minimize the *expected energy consumption* while still providing performance guarantees. To this aim, we use probabilistic workload models for both computation and communication activities. The computational model uses *cycle groups* [9–11], a concept that supports the empirical estimation of an underlying workload probability distribution. We adopt a similar approach to model the communication workload.

Our work evaluates seven different algorithms, including a joint DVS–DMS approach and a computationally simple heuristic. Using our probabilistic workload, deadline and energy models, the joint approach formulates the problem as one that can be solved through convex optimization. We present an efficient off-line solution to this problem. Our work is based on the observation that in probabilistic workload settings, the optimal solution consists of starting with low computation and communication speed levels, and then gradually increasing the speeds as the task makes progress. We

* Corresponding author. Tel.: +1 703 9933786.
E-mail addresses: mbandari@gmu.edu (M. Bandari), simon@gmu.edu (R. Simon), aydin@gmu.edu (H. Aydin).

http://dx.doi.org/10.1016/j.suscom.2016.02.004
2210-5379/

Please cite this article in press as: M. Bandari, et al., On minimizing expected energy usage of embedded wireless systems with probabilistic workloads, Sustain. Comput.: Inform. Syst. (2016), http://dx.doi.org/10.1016/j.suscom.2016.02.004

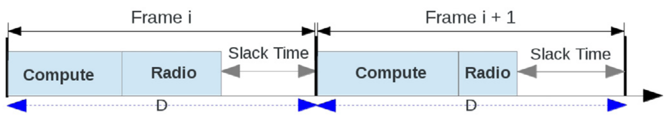

Fig. 1. Application model.

show how to compute the optimal *speed schedule* in which DVS and DMS parameters are adjusted to match the current workload conditions. We first study the optimal CPU and radio speed scheduling algorithms in the continuous speed domain. To account for the fact that in current hardware design speed levels can only change by certain step sizes, we extend the solution to cover the discrete domain.

We also present a general purpose simulation model that represents a wide variety of processor and radio types. We then describe an extensive simulation study of our various algorithms with a particular interest in evaluating the benefits of our integrated DVS–DMS approach under probabilistic workloads and as a function of the ratio of the radio power to the CPU power, by comparing to other algorithms, including those that use DVS-only or DMS-only approaches for energy management.

To our knowledge this is the first study that considers *both* DVS and DMS in a wireless embedded system using *probabilistic* computation and communication workload models. Our results precisely quantify the improvements offered by these control techniques as a function of the underlying hardware characteristics, and can be used by designers as a guideline for algorithm selection. Of particular importance of this work is the demonstration of the potential value of DMS techniques [6]. For instance, our experimental results show that an integrated DVS–DMS strategy can provide non-trivial gains on the expected energy consumption, especially when the computation and communication workloads are relatively balanced.

The rest of this paper is organized as follows. In Section 2 we present our power and application models, as well as our assumptions. By assuming an ideal system where the CPU frequency and modulation levels can be adjusted continuously, the energy minimization problem is formulated and solved in Section 3. Assuming discrete frequency and modulation levels, the same problem is formulated as a mixed binary integer programming problem in Section 4. In Section 5, a detailed performance evaluation of several algorithms, including optimal algorithms, fast heuristics and those that use only the DVS or DMS feature, is presented. Section 6 surveys related work, and we conclude in Section 7.

2. System model

This section describes the application model, presents the system level energy components and shows how to derive the expected energy.

2.1. Application model

We consider an embedded wireless node with two major activities: data processing (computation), performed by the CPU, and communication with other wireless embedded devices, performed by the radio. Specifically, as in [7], we assume that computation and communication activities form two *sub-tasks*, executed within a *frame* (Fig. 1). A frame is a time interval of length D that repeats periodically during the lifetime of the node with the rate $1/D$. Input to the radio communication sub-task depends on the output of the computation sub-task; consequently, the latter is to be executed first in each frame. Both sub-tasks must be completed within a relative deadline of D, by the end of frame. The sub-tasks may have varying resource demands from frame to frame, determined according to specific probability distributions, as explained below.

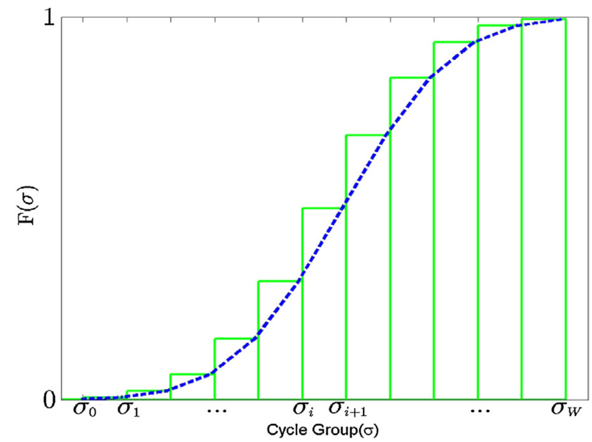

Fig. 2. Histogram-based approximation of the cumulative distribution function of the application's probabilistic workload.

2.1.1. Computation workload

In real applications, the number of CPU cycles in a given frame (the computation workload) can be known only *probabilistically* in advance. We denote the minimum and maximum computational workload demand in a single frame by C_{min} and C_{max} cycles, respectively. In general, the cumulative probability distribution function for the computation workload is:

$$F(c) = p(X \leq c)$$

where X is the random variable for the application's computation demand in a frame, and $p(X \leq c)$ represents the probability that the application will not require more than c cycles in a single frame. This function can be approximated through the histogram-based *profiling* approach [9,12,13]. Specifically, the available range of CPU cycles $[C_{\min}, C_{\max}]$ is divided into W discrete *cycle groups*, each with $\omega = (C_{\max} - C_{\min})/(W)$ cycles. We denote the upper bound on the number of cycles in the ith cycle group as σ_i, that is, $\sigma_i = C_{\min} + (i-1) \cdot \omega$.

The workload probability distribution function may be obtained by multiple means. One approach is profiling over a fixed window size for workloads with self-similarity property [14]. In general, to obtain the histogram-based profiles, the application's executions over a long time interval is monitored, and the fraction of invocations in which the number of actual cycles fall in the ith cycle group are recorded [9–11]. More precisely, the fraction of invocations where the number of executed cycles falls in the ith cycle group during the profiling phase, is assumed to correspond to the probability that the number of cycles will fall in this specific range over a long-term periodic execution. In this way, the probability that the actual number of cycles needed by the application will fall in the range $(\sigma_{i-1}, \sigma_i]$, denoted by f_i^{cp}, is derived for $i = 1, \ldots, W$. Observe that $\sum_{i=1}^{W} f_i^{cp} = 1$.

We can calculate the cumulative probability distribution function (Fig. 2) of the application's cycle demand as:

$$F_j^{cp} = \sum_{k=1}^{j} f_k^{cp}$$

F_j^{cp} denotes the probability that the application will require no more than j cycle groups (i.e., *at most* $C_{\min} + (j-1) \cdot \omega$ cycles) in one frame. Consequently, $\Gamma_j^{cp} = 1 - F_{j-1}^{cp}$ is the probability that the task will require *more than* $(j-1)$ cycle groups, or equivalently, the