Analysis of Energy Reduction on Dynamic Voltage Scaling-Enabled Systems

Lin Yuan *student member, IEEE* and Gang Qu *member, IEEE*

***Abstract*— Dynamic voltage scaling (DVS) is a technique that varies the supply voltage and clock frequency, based on the computation load, to provide desired performance with the minimal amount of energy consumption. It has been demonstrated as one of the most effective low power system design techniques, particularly for real time embedded systems. Most existing work are on two different system models that enable DVS: the ideal DVS system that can change its operating voltage with no physical constraints, and the multiple DVS system that has only a number of discrete voltages available. Although the ideal DVS system provides the theoretical lower bound on system's energy consumption, it is the practicability of multiple DVS systems and the emergence of other DVS-enabled systems, which do not fit either model, that challenges system designers the following questions:** *should DVS be implemented in the design or not? if so, how should DVS be implemented?*

In this paper, we answer these questions by studying the DVS-enabled systems that can vary the operating voltage dynamically under various real-life physical constraints. Based on system's different behavior during voltage transition, we define the optimistic feasible DVS system and the pessimistic feasible DVS system. We build mathematical model for each DVS-enabled system and analyze their potential in energy reduction. Finally, we simulate a secure wireless communication network with different DVS-enabled systems. The results show that DVS gives significant energy saving over system with fixed voltage. Interestingly, we also observe that although multiple DVS system may consume more energy than the theoretical lower bound, the optimistic and pessimistic feasible DVS systems can achieve energy savings very close to the theoretical bound provided by the ideal DVS system.

***Index Terms*— Low-power design, Optimization, Power minimization, Scheduling, Dynamic voltage scaling**

I. Introduction

IT has been a decade since researchers and engineers pioneered the study of CMOS circuits power/energy reduction by using different supply voltages. Nowadays, design with variable/multiple voltage has become one of the standards for low power system design and, as predicted in the latest International Technology Roadmap for Semiconductors, the trend is to have multiple voltage on a single chip to allow different part of the system operating at different voltage to reduce power while maintaining performance. Our goal in this paper is to build formal models for different variable voltage systems and use these models to analyze the limit of energy saving provided by the dynamical voltage scaling (DVS) technique.

Lin Yuan and Gang Qu are with the Electrical and Computer Engineering Department and the Institute for Advanced Computer Study at the University of Maryland, College Park, MD, 20742, USA (email: gangqu@eng.umd.edu).

A. Motivation

Power and energy efficient design has emerged as a very active research area recently. In fact, it becomes one of the most important system design concerns. Low power dissipation reduces the cost on packages and heat-sinks, and increases the circuit's reliability. For battery-operated systems, low energy consumption extends battery's lifetime, reduces the cost for system maintenance, and increases the system's lifetime when recharging or replacement is not allowed (e.g., sensor networks). Studies on energy reduction techniques have been conducted by researchers in the communities of circuit design, system-level design, real time operating systems, compilers, communication, networking, among others.

Traditionally, systems have been designed to operate at a fixed supply voltage with a fixed clock frequency. Recent advances in power supply and circuit design technologies allow the implementation of microprocessor system that can adjust the operating voltage (and thus the clock frequency) at run-time. Dynamic voltage scaling method takes advantage of the fact that lowering voltage can reduce power quadratically to reduce the energy consumption. Voltage is scaled down to an appropriate level whenever it is possible. Such variable voltage systems can achieve extremely low power/energy consumption comparing to the standard systems with fixed supply voltage.

A voltage scheduler determines when and to which level the system should scale the operating voltage. Its goal is to assign each task an execution time as long as possible such that the system can run at the lowest possible voltage. The voltage scheduler is usually built in the system's real time operating system and makes its decision based on system level information (such as current computation load and predicted future behavior). Normally, the voltage scheduler will be reevaluated on the arrival and completion of a task, and periodically during task execution [14]. Scheduling policies, from the simple but powerful earliest deadline first (EDF) to some sophisticated adaptive policies based on recursive learning and empirical studies, play an important role in DVS-enabled systems. In various application domains, DVS results in energy savings (typically over a fixed voltage system) from 1.4% to as high as 90% [8], [10], [15], [16], [17].

Early theoretical results indicate that DVS technique reaches its full potential in energy reduction on the ideal DVS system with certain unrealistic assumptions (e.g., the supply voltages can be changed simultaneously with no physical constraints)[7], [20]. However, it takes time for system to reach the steady state at a new voltage level and there exist the maximum/minimum voltages at which the circuit can reliably

operate. The impracticalness of such ideal variable voltage system motivates the study on multiple voltage systems, where only several pre-designed operating voltages are physically available on the chip and the system can select any one of these voltages and the corresponding clock frequency at run time[5], [10], [17]. Clearly, such multiple voltage systems cannot reach the full potential of energy reduction provided by the DVS technique because they are restricted to a limited number of voltage levels.

Our work is motivated by the recent implementation of low power ARM (lpARM) microprocessor that can adjust its operating speed at run-time[1], [14]. Such processor is different from both the ideal and the multiple DVS systems. There is no theoretical model and study on this system's energy saving. More importantly, we lack a formal analysis of different types of DVS-enabled systems featuring a framework estimating their maximal energy savings. This paper is the first step towards such analysis. It helps to answer questions such as whether DVS should be used and what type of DVS-enabled systems should be used to meet the design's power budget, and therefore sheds light on early design space exploration for low power system designs.

Our work is also motivated by the increasing importance of static power dissipation in deep submicron VLSI circuits. DVS technique has the system operating at lower voltage to reduce dynamic energy dissipation. It consequently results in reduced speed, longer execution time, and increased static current, which lead to increased static (or leakage) energy dissipation. It is out of the scope of this paper, but to study the interesting problem of total (dynamic and static) energy saving by DVS, understanding the potential of DVS on (dynamic) energy saving is crucial.

B. Contributions and Paper Organization

Most of the existing work on voltage scaling assumes that the system can change the operating voltage for the CPU or the microprocessor either arbitrarily or only to one of the pre-designed discrete voltage levels. Naturally, we call them the **ideal** DVS system and the **multiple** DVS system, respectively.

Our first contribution is the formal modeling of two other types of DVS-enabled systems: **optimistic feasible** DVS system and **pessimistic feasible** DVS system. They are both implementable (unlike the ideal DVS system) and have more flexibility on their operating voltages (unlike the multiple DVS system). This fills the gap between the ideal DVS system and the multiple DVS system.

The categorization of different type of DVS systems enables us to study their potential of energy saving. We propose a systematical approach to do so based on the solution to the following problem: *under different DVS models, what is the most energy efficient way to scale voltage such that the DVS system can start from a given voltage, complete a given amount of computation (a task or a job) and reach a designated voltage, which can be different from the starting voltage, at the end of a given period?*

We then give explicit expressions of the most energy efficient voltage scaling scheme under each DVS model. From these one can easily estimate the system's energy saving over the traditional fixed voltage system. The previously known results on the ideal and multiple DVS systems can be directly derived using our framework. The solutions for the optimistic and pessimistic feasible DVS systems are new and non-trivial.

Finally, we simulate these DVS-enabled systems in a secure wireless communication network. The results demonstrate that 1) all DVS systems are effective in energy saving over the fixed voltage system; 2) multiple, pessimistic feasible, optimistic feasible, and ideal DVS systems, in that order, become more energy efficient as we relax the physical constraints on how voltage may vary; 3) the optimistic and pessimistic feasible DVS systems only consume a little more energy than the ideal DVS system, which suggests that the full potential of DVS in energy saving can (almost) be reached.

The rest of the paper is organized as follows: In the next section, we define the four different types of DVS-enabled systems, describe the applications executing on such systems, and formulate the energy minimization problem. We survey the existing results on the ideal and multiple DVS systems in Section III. We present our main results in Section IV, where the proposed energy minimization problem is optimally solved in a special case for all DVS models. We show that this solution gives the upper bounds on the energy savings by DVS. We also discuss how to solve the more general case in this section. We present the simulation setup and results in Section V and conclude in Section VI. Detailed proof of most lemmas and theorems are provided in the Appendix for brevity.

II. Dynamic Voltage Scaling System Models

Reducing the supply voltage can result in substantial reduction on switching power (also known as dynamic power), the dominant source of power dissipation in CMOS circuits, which is proportional to $\alpha C_L v_{dd}^2 f_{clock}$, where αC_L is the effective switched capacitance, v_{dd} is the supply voltage, and f_{clock} is the system clock frequency. Roughly speaking, system's power dissipation is halved if we reduce V_{dd} by 30% without changing any other system parameters. However, this saving comes at the cost of reduced throughput, slower system clock frequency, and longer gate delay. The gate delay is proportional to $\frac{v_{dd}}{(v_{dd}-v_t)^\beta}$ where v_t is the threshold voltage and $\beta \in (1.0, 2.0]$ is a technology dependent constant. Naturally, we have the power and delay trade-off: on one hand, we want to scale voltage as low as possible to reduce power/energy; on the other hand, system operating at a low supply voltage may fail to complete the required computation within the given deadline. Moreover, we have the dynamic power and static power trade-off: system operating at lower voltage consumes less dynamic energy, but it may cause more static energy dissipation as both the static current and the execution time will increase.

We study the execution of a set of applications (tasks, or jobs) on different DVS-enabled systems, particularly the system's energy dissipation to complete the given applications. Coupled with system level task scheduling, DVS can achieve energy saving over the traditional fixed voltage system. Our

goal is to determine the upper bounds of such energy saving. A voltage scheduler of the DVS-enabled system determines the level of operating voltage and the application to execute at any given time. A scheduler is valid if it schedules each application after the application's arrival and completes the application before its deadline without violating any constraints on voltage scaling. We seek for a valid scheduler that consumes the least amount of energy to complete all the applications.

Formally speaking, for a given set of applications $\{\tau_1, \tau_2, \cdots, \tau_n\}$, where each τ is characterized by:

- a**:** the arrival time of application τ
- d**:** the deadline for the system to complete application τ
- W**:** the application's computation load or equivalently
- e**:** the application's execution time at the reference voltage v_{ref}

suppose that all the applications need to be processed during the interval $[t_1, t_2]$, let $v(t), s(t), P(t),$ and $\Delta(t)$ be the operating voltage, processing speed, (dynamic) power dissipation, and the application selected by the scheduler to execute at time $t \in [t_1, t_2]$, we want to minimize the total energy consumption

$$E = \int_{t_1}^{t_2} P(t)dt \tag{1}$$

or equivalently, the average power consumption $\frac{E}{t_2 - t_1}$ under the constraint that all applications are completed under their respective timing constraints:

$$W_i = \int_{t_1}^{t_2} s(t) \cdot \delta(\Delta(t), i)dt = \int_{a_i}^{d_i} s(t) \cdot \delta(\Delta(t), i)dt \tag{2}$$

where $\delta(\Delta(t), i) = 1$ if application τ_i is being executed at instant t and $\delta(\Delta(t), i) = 0$ otherwise. The second equality in Equation (2) guarantees that all applications are scheduled after their arrival and finished before their individual deadlines.

The goal of this paper is to analyze DVS technique's potential in energy saving. To reach this goal, we focus on the execution of a single task (see the problem formulation in Section IV). Although our provably optimal results can be extended to any set of scheduled tasks, it remains a challenge for the general case due to the NP-completeness nature of the general scheduling problem.

Based on how the operating voltage can be changed, we consider the following DVS models:

- **Ideal:** An ideal variable voltage system can change its operating voltage of the CPU arbitrarily and instantaneously. That is, the processing speed can go from 0 to ∞ without any delay.
- **Feasible:** A feasible variable voltage system can vary the voltage between a minimum voltage v_{min} and a maximum voltage v_{max}. The maximum voltage change rate is K. That is, if the voltage at time t is v, then at time $t + \delta$, the voltage value must be in the interval of (v_1, v_2), where $v_1 = \min\{v_{min}, v - K \cdot \delta\}$ and $v_2 = \max\{v_{max}, v + K \cdot \delta\}$.
  Because it requires non-zero time for the circuit to reach the steady state at the new voltage level, different models can be adopted for the system's processing during such voltage transition. The **optimistic feasible** DVS system allows its CPU to continue processing at the instantaneous voltage; while the **pessimistic feasible** DVS system stops execution during the voltage transition until the steady state is reached at the new voltage level.
- **Multiple:** A multiple voltage system has only a number of discrete operating voltages available simultaneously and the CPU can switch from one voltage to another instantaneously.

The speed function s in Equation (2) is a non-decreasing function of the operating voltage $v(t)$. It takes the form of a step function for the **multiple** DVS system because such system has only discrete voltage levels. For the **pessimistic feasible** system, which stops execution (that is, $s(t) = 0$) during voltage transition, the speed function remains as a step function. However, the stable processing speed can assume any value between the physical minimum s_{min} and maximum s_{max}. (The lpARM processor has a similar speed function.). The **optimistic feasible** DVS system assumes that the system processes at the instantaneous voltage between v_{min} and v_{max}. Therefore, the speed function is continuous. Finally, the **ideal** DVS system can have arbitrary speed function because there is no physical constraints on voltage scaling.

Clearly, as we move from the ideal model to the feasible model and to the multiple model, more and more constraints are imposed on voltage scaling. These constraints limit the power of DVS on energy reduction but make DVS practical.

III. State-of-the-Art on DVS

We now briefly survey previous work on DVS for energy efficient system design in three categories: (1) practice and challenges in implementing DVS systems; (2) low power study on multiple DVS system; and (3) high-level scheduling techniques for power reduction on ideal DVS system.

Dynamically adapting voltage (and the clock frequency) to operate at the point of lowest power consumption for given temperature and process parameters was first suggested by Macken et al. [12] in 1990. This idea has been implemented in numerous circuits and systems, particularly in the past few years (see, for example, [1], [2], [8], [14]). Transmeta's Crusoe, AMD's K-6, Intel's XScale and Pentium III and IV, and some DSPs developed in Bell Labs are all examples of advanced high-performance microprocessors that support DVS for energy and power efficiency. Having the system operating at lower voltage reduces dynamic energy dissipation at the cost of increased static current and reduced speed, both lead to increased static (or leakage) energy dissipation. One of the most challenging and interesting problems is how to implement DVS to minimize system's total power/energy while improving circuit's robustness and reliability [6], [11], [18].

Early research on voltage scaling is on the multiple DVS system where multiple voltage levels are simultaneously available on the chip. At behavioral level, this enables operations off the critical path to be executed at reduced voltages to save power and energy. Desired performance can be maintained as long as operations on the critical path operate at the high reference voltage [5], [17]. At high level, this makes it possible

to assign individual task different voltage to reduce energy consumption. Most of such work focus on task scheduling to achieve energy savings while providing real-time deadline guarantees [9], [10], [16], [19].

Yao et al. [20] suggested a task-level scheduling model, where they assume that the CPU speed can be changed arbitrarily as a result of voltage scaling. In this model, there is no physical constraints for speed (or voltage) and the system operates at instantaneous speed without any delay to reach the steady-state at a new voltage level. Such DVS system is impractical and hence we call it *ideal DVS system*. The study of ideal DVS system gives us an upper bound, which is unreachable and can be very loose, on how much energy can be saved by DVS. We do not elaborate these task scheduling approaches as our work is independent of task scheduling. That is, we seek for the most power/energy efficient voltage (or speed) function to complete a given set of scheduled tasks.

There has been little discussion on the voltage (or speed) scheduling policies and their potential in power/energy saving for feasible DVS systems, largely due to the fact that such systems are hard to implement. However, this has changed recently since the implementation of the low power ARM (lpARM) microprocessor system [1], [14]. The lpARM processor is based on the ARM8 core and designed to operate between 1.1v and 3.3v, resulting in speeds between 10MHz and 100MHz. Clock frequency transition take approximately $25\mu s$ (about 1250 cycles) for a complete 10MHz to 100MHz transition. The system can continue operation while the voltage/speed is changing. In another word, it belongs to feasible systems according to our classification. Further design issues for DVS systems are discussed by the same authors [2]. One objective of this paper is to provide formal model so one can analyze the power efficiency of such feasible DVS systems.

IV. Upper Bounds on Energy Saving by DVS

We now study the following fundamental problem: *For a given starting voltage (or speed), an ending voltage (or speed), and a workload to be completed in a given period, determine the most energy efficient way to scale voltage on a DVS system such that the workload is completed and the ending voltage (or speed) is reached at the end of the period.*

Mathematically speaking, we want to determine the operating voltage $v(t)$, and hence the operating speed $s(t)$, for (particularly feasible) DVS systems over the period $[t_1, t_2]$ such that the energy consumption $\int_{t_1}^{t_2} P(t)dt$ is minimized and the following conditions are satisfied:

$$s(t_1) = s_0 \tag{3}$$

$$s(t_2) = s_1 \tag{4}$$

$$s_{min} \leq s(t) \leq s_{max} \quad (t \in [t_1, t_2]) \tag{5}$$

$$|\frac{ds(t)}{dt}| \leq K \tag{6}$$

$$\int_{t_1}^{t_2} s(t)dt = W \tag{7}$$

Equations (3) and (4) specify the starting speed and ending speed for the DVS system. For multiple DVS system, we necessarily assume that both ending speeds correspond to the available supply voltage levels. Equation (5) gives the lowest and highest physical processing speeds and it does not apply to the ideal DVS system where there is no limitation on speed. Equation (6) is the maximum rate that speed (or voltage) can be changed and applies only to feasible DVS system. We take the assumption that inequality $|s_1 - s_0| \leq K \cdot (t_1 - t_0)$ holds whenever Equation (6) applies to ensure that there exist solutions (speed function) for Equations (3), (4), and (6). The last equation guarantees that the required workload will be completed during the given period.

A. Ideal and Multiple DVS Systems

From Equation (7), it is trivial to obtain the optimal speed functions for ideal and multiple DVS systems from the convexity of the power function. Here we give the analytic expressions of these functions under our single task problem formulation. Study on ideal and multiple DVS systems for the general multiple task case can be found in [10] and [20], respectively.

Theorem IV.1. (Optimal speed function for the ideal DVS system). *The speed function for the ideal DVS system that satisfies Equations (3)-(7) and minimizes energy is given by:*

$$s(t) = \begin{cases} s_0, & \text{if } t = t_1; \\ \frac{W}{t_2 - t_1}, & \text{if } t_1 < t < t_2; \\ s_1, & \text{if } t = t_2. \end{cases}$$

Theorem IV.2. (Optimal speed function for multiple DVS systems). *Let $sp_1 < sp_2 < \cdots < sp_l$ be the processing speeds supported by a multiple DVS system with l distinct voltage levels. For any valid workload $W \in [sp_1 \cdot (t_2 - t_1), sp_l \cdot (t_2 - t_1)]$, the most energy efficient speed function satisfying Equations (3)-(7) is given by:*

$$s(t) = \begin{cases} s_0, & \text{if } t = t_1; \\ sp_i, & \text{if } t_1 < t \leq t_c; \\ sp_{i+1}, & \text{if } t_c < t < t_2; \\ s_1, & \text{if } t = t_2. \end{cases}$$

where $t_c = t_1 + \frac{sp_{i+1}(t_2 - t_1) - W}{sp_{i+1} - sp_i}$ and sp_i and sp_{i+1} are the two consecutive speed levels such that $sp_i(t_2 - t_1) \leq W < sp_{i+1}(t_2 - t_1)$.

B. Optimistic Feasible DVS System

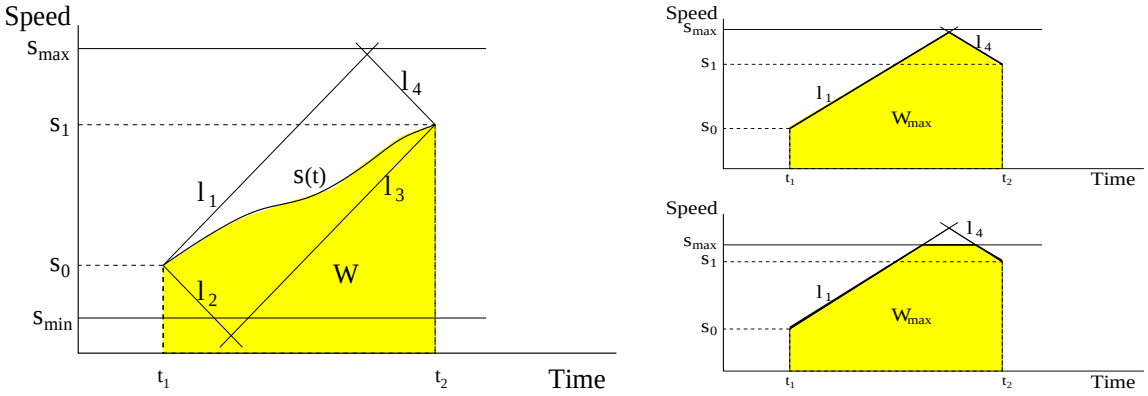
The optimistic feasible DVS system continues operating at the instantaneous speed during voltage transition. Due to the physical constraints on voltage (and thus speed) scaling, the workload accumulated by the feasible DVS system in a given period is limited. More specific, we have

Lemma IV.1. *If there exists a speed function satisfying Equations (3)-(7) on an optimistic feasible DVS system, then the workload W necessarily satisfies $W_{min} \leq W \leq W_{max}$, where*

$$W_{min} = s_c(t_2 - t_1) + \frac{(s_0 - s_c)^2}{2K} + \frac{(s_1 - s_c)^2}{2K}$$

$$W_{max} = s_d(t_2 - t_1) - \frac{(s_0 - s_d)^2}{2K} - \frac{(s_1 - s_d)^2}{2K}$$

and $s_c = max(s_{min}, \frac{(s_0 + s_1)}{2} - K\frac{(t_2 - t_1)}{2})$, $s_d = min(s_{max}, \frac{(s_0 + s_1)}{2} + K\frac{(t_2 - t_1)}{2})$.



(a) Restrictions on the speed function and illustration of the workload completed by a speed function.

(b) Illustration of the maximum workload when $s_0 < s_1$.

Fig. 1. Illustration of Lemma IV.1

[Proof:] Note that the optimistic feasible DVS system cannot change processing speed arbitrarily under the maximal rate constraint in Equation (6). Therefore, if we start with speed s_0 at time t_1 and reach speed s_1 at t_2, the speed function $s(t)$ is restrained by the four lines l_1, l_2, l_3, and l_4 as shown in Figure 1(a) (for the case $s_0 < s_1$). The two horizontal lines reflect the physical constraints on minimum speed and maximum speed. The area under the speed function $s(t)$, which equals to the integral in Equation (7), gives the workload completed by operating at $s(t)$ from t_1 to t_2.

From these constraints on the speed function, we can easily see that to accumulate the maximum workload W_{max}, we need to increase the speed at the fastest rate and stay at high speed as long as possible. That is, the speed function will move along line l_1 or the horizontal line corresponding to the maximum speed s_{max}, whichever is lower, till it hits line l_4, where it has to decrease to reach the final speed s_1 as required. This is illustrated in Figure 1 (b) assuming that $s_0 < s_1$. We can calculate the value of W_{max} in both cases and unify it as shown above in the lemma. The minimum workload W_{min} can be determined in the same way. □

Theorem IV.3. (Optimal speed function for optimistic feasible DVS systems). *For any valid workload $W \in [W_{min}, W_{max}]$, there is a unique speed function $s(t) : [t_1, t_2] \rightarrow [s_{min}, s_{max}]$, defined as follows, that satisfies Equations (3)-(7) and minimizes the energy consumption.*

if $W_{min} \leq W \leq W_1$,

$$s(t) = \begin{cases} s_0 - K(t - t_1), & \text{if } t_1 \leq t \leq x_1; \\ s_0 - K(x_1 - t_1), & \text{if } x_1 < t \leq x_2; \\ s_1 - K(t_2 - t), & \text{if } x_2 < t \leq t_2. \end{cases}$$

if $W_1 \leq W \leq W_2$,

$$s(t) = \begin{cases} s_0 + K(t - t_1), & \text{if } t_1 \leq t \leq x_1; \\ s_0 + K(x_1 - t_1), & \text{if } x_1 < t \leq x_2; \\ s_1 - K(t_2 - t), & \text{if } x_2 < t \leq t_2. \end{cases}$$

and if $W_2 \leq W \leq W_{max}$,

$$s(t) = \begin{cases} s_0 + K(t - t_1), & \text{if } t_1 \leq t \leq x_1; \\ s_0 + K(x_1 - t_1), & \text{if } x_1 < t \leq x_2; \\ s_1 + K(t_2 - t), & \text{if } x_2 < t \leq t_2. \end{cases}$$

where W_1, W_2, x_1, x_2 are constants that are dependent on the boundary conditions t_1, t_2, s_0, s_1, and the maximum speed change rate K.

[Proof:] See the Appendix.

C. Pessimistic Feasible DVS System

The pessimistic feasible DVS system differs from the optimistic feasible system in that execution stops during voltage transition until the steady state at the new voltage level is reached. We summarize our results in the following lemma and theorem whose detailed proofs can be found in the Appendix.

Lemma IV.2. *The pessimistic feasible DVS system can achieve any workload $W \leq W_{max}$ with speed functions satisfying Equations (3)-(7), where $W_{max} = (t_2 - t_1 + \frac{s_0+s_1}{K})s - \frac{2s^2}{K}$ and $s = \min\{s_{max}, \frac{K(t_2-t_1)}{4} + \frac{s_0+s_1}{4}\}$.*

Theorem IV.4. (Optimal speed function for pessimistic feasible DVS systems). *For any valid workload $0 \leq W \leq W_{max}$ on a pessimistic feasible DVS system, there is a unique speed function $s(t)$ such that Equations (3)-(7) will hold and the energy consumption is minimized.*

D. Summary and Discussion

In this section, we consider the time and energy overhead for voltage scaling, how to extend our results from a single task to multiple (scheduled) tasks, and the impact of static power to our approach and results.

1) Voltage Scaling Overhead: In our model and analysis, we assume that both time and energy overhead for voltage scaling are negligible. Recently, there have been approaches that take such overhead into consideration explicitly [13], [21]. The energy consumed by the DC-DC converter to switch voltage from one level v_1 to another level v_2 is modeled as $EO = (1 - \eta) \cdot C_{DD} \cdot |v_1^2 - v_2^2|$, where η is the efficiency of the DC-DC converter and C_{DD} is the capacitor that stores the charge [2], [21]. We can easily integrate the time and energy overhead into our analysis using this model.

Consider the multiple DVS model for example. When we ignore the voltage transition overhead, Theorem 4.2 gives the

optimal solution which operates at the lower speed sp_i during period (t_1, t_c) before speeding up to sp_{i+1} for period (t_c, t_2). Let TO and EO be the time and energy overhead to change processing speed from sp_i to sp_{i+1}, the most energy efficient speed function can be obtained as follows:

(i) compute E_{old}, the energy consumption of running at high speed sp_{i+1} only during time interval (t_1, t_2) to complete the task;
(ii) compute the best speed function $s(t)$ to complete the same task in time interval $(t_1, t_2 - TO)$. As given in Theorem 4.2, $s(t) = sp_i$ for $t \in (t_1, t'_c)$ and $s(t) = sp_{i+1}$ for $t \in (t'_c, t_2 - TO)$;
(iii) compute E_{new}, the energy consumption by the above speed function;
(iv) if $E_{old} > E_{new} + EO$, report the following optimal speed function: $s(t) = sp_i$ for $t \in (t_1, t'_c)$, voltage/speed transition during time interval $(t'_c, t'_c + TO)$, and $s(t) = sp_{i+1}$ for $t \in (t'_c + TO, t_2)$;
(v) if $E_{old} \leq E_{new} + EO$, report the following optimal speed function: use only the high speed sp_{i+1} till the completion.

2) Multiple Scheduled Tasks: General real time embedded systems deal with multiple tasks, each with its own arrival time, deadline, and computation requirement (workload). System level task scheduling and voltage scaling should be combined to solve this problem. Yao et al. [20] solved this optimally for the **ideal** DVS system when preemption is allowed. Ishihara and Yasuura [10] discovered some interesting features for the **multiple** DVS system and solved the problem by translating it into an integer linear programming problem. It is out of the scope of this paper to study the task scheduling problem, which is NP-complete. However, for any set of scheduled tasks, the optimal speed function can be determined based on our results for the single task case:

Theorem IV.5. (Optimal speed function for multiple scheduled tasks). *For a set of scheduled tasks, the most energy efficient speed function on a DVS system can be obtained by solving a non-linear system.*

[Proof:] We outline how to set up the non-linear system. Suppose that task τ_i is scheduled for execution from time t_i to t'_i with $t_1 < t'_1 \leq t_2 < t'_2 \leq \cdots \leq t_n < t'_n$. Let $s(t)$ be a speed function on a DVS-enabled system executing this set of tasks with the minimum energy. Consider the execution of task τ_i, it starts at time t_i with speed $s(t_i)$ and finishes at t'_i with speed $s(t'_i)$ to complete the required workload W_i. For any DVS-enabled system, Theorems IV.1-IV.4 give the explicit (and unique) expression of the most energy efficient speed function, in terms of $s(t_i)$ and $s(t'_i)$, in the interval $[t_i, t'_i]$. Assuming that the system shuts down in $[t'_i, t_{i+1}]$ to conserve energy, we obtain a unique expression of the speed function on $[t_1, t'_n]$. Then the energy consumed according to such speed function can be determined either by analytic [8] or empirical formulas [7]. This energy E will be a function of $s(t_i)$ and $s(t'_i)$ and the non-linear system can be set up by applying the first order conditions $\frac{\partial E}{\partial s_i} = \frac{\partial E}{\partial s'_i} = 0$. □

3) Impact of Static Power: We have mentioned that DVS reduces system's dynamic energy dissipation at the cost of increased static current and reduced speed, both lead to increased static (or leakage) energy dissipation. First, this increase may even cancel the (dynamic) energy saving by DVS. Second, high leakage current has negative impact on CMOS circuit's robustness and increases system's vulnerability to soft errors [6]. Third, leakage power depends on the die temperature which will increase when the system dissipates more power. So improper and aggressive voltage scaling may cause heat accumulation on the chip and thermal runaway [6], [11].

The two key sources of leakage are subthreshold leakage and gate leakage. With the optimistic predication that gate leakage can be controlled by high-k dielectric gate insulators, we focus on the impact of static power to our DVS models and results. The subthreshold leakage current $I_{sub} \propto ke^{-\frac{v_t}{nv_\theta}}(1 - e^{-\frac{v_{dd}}{v_\theta}})$, where k and n are constants, v_{dd}, v_t, and v_θ are supply voltage, threshold voltage, and thermal voltage (around $25mV$ at room temperature), respectively [4]. From this, we see that we can ignore I_{sub} when the system is shut down ($v_{dd} = 0$). However, when we scale down v_{dd}, v_t also needs to be scaled down to keep the performance and this cause the exponential increase of I_{sub}.

We can then model the system's total power as $P_{total} = P_{dynamic} + P_{static}$, where $P_{static} = (I_{sub} + I_{others}) \cdot v_{dd}$. Unfortunately, P_{total} is not a convex function of voltage (or speed). Therefore, our analytic results on the dynamic power reduction by DVS will not hold. Due to the complexity of the static power function and the physical constraint between v_{dd} and v_t during voltage scaling, it is impossible to derive similar analytical results. Instead, one can use advanced numerical methods (such as curve fitting) to obtain approximate solutions. Finally, we mention that the recent studies on how to reduce leakage by controlling the threshold voltage v_t, such as multiple and dynamic threshold voltage techniques, can be easily integrated into our problem formulation and numerical results can be obtained.

V. Experimental and Simulation Results

The goal of our experiment and simulation is to 1) show the energy efficiency of DVS over the fixed voltage system; 2) demonstrate that the proposed feasible DVS models provide better bounds on energy reduction than the ideal and multiple DVS models; and 3) reveal how close the feasible DVS systems are from the ideal model in terms of energy saving.

A. Simulation Setup

To achieve the above simulation goals, we consider the following five DVS-enabled systems and compare their energy consumption on a given set of applications with the energy by a fixed 3.3v system:

1) a 2-level **multiple** DVS system: 3.3v and 2.4v;
2) a 3-level **multiple** DVS system: 3.3v, 2.4v, and 1.2v;
3) a **pessimistic feasible** DVS system with $v_{min} = 1.1v$ and $v_{max} = 3.3v$ and the transition time from minimum performance to maximum performance is $25\mu s$;
4) an **optimistic feasible** DVS system with the same setting as the above pessimistic feasible DVS system;
5) an **ideal** DVS system.

The data (v_{min}, v_{max}, and the maximum voltage change rate) for the two feasible DVS systems are taken from [2]. The voltage levels for the two multiple DVS systems are selected with no specific reason. Recently, Hua and Qu [9] discussed how to select the voltage levels to optimize the energy saving when the application's execution information is known. Their approach does not work very well when the execution time is unpredictable or follows the uniform distribution like in our case. We have also conducted experiments on multiple DVS systems with up to five voltages (3.3v, 2.8v, 2.4v, 1.8v, and 1.2v). We will not elaborate all the results because systems 1 and 2 above are representative.

The applications come from a secure wireless sensor network where all the messages transmitted in the network are encrypted using RSA. On the reception of an encrypted message, the system first decrypts the message, then processes the data if necessary, and finally encrypts the result and sends it out if necessary. Each message consists of 1 to 20 1024-bit packets. A sensor node in the network may receive three different kinds of messages and take actions accordingly: (I) messages that are obsolete or have reached the wrong node that will be rejected by the receiving node (only message decryption is performed); (II) messages that only needs to be forwarded to the next node (message decryption and encryption are required); (III) messages that need to be processed on the current node and the result needs to be sent to other nodes as a new message (message decryption, data processing, and new message encryption are performed).

In our problem formulation, we assume that each task's workload is known. This assumption is valid for the above sensor network. The decryption process starts on the reception of the whole message and at that time, the length of the message (in terms of number of packets) becomes known. So is the decryption time because the message is decrypted packet by packet and the computation for decrypting each packet is fixed. For the same reason, we also know the encryption time before the encryption starts. The data processing time can be rather accurately estimated once the message is decrypted mainly because of the simplicity of the sensor node's functionality. These execution time information helps the system to select proper voltage for message decryption, data processing, and message encryption.

We simulate a sequence of messages received at one node with the following parameters: the inter-arrival rate of messages 0.125, 20% of the messages are of type (I), 50% of the messages are of type (II), and the rest 30% are type (III) messages with data processing time uniformly distributed between $500ms$ and $4000ms$. Both the receiving message and processing result are of size 200 bits to 20000 bits (that is, 1 to 20 packets). We conduct a one-hour-long simulation where we generate a total of 461 messages: 85 messages of type (I), 246 messages of type (II), and the rest 130 are of type (III).

The time and energy consumption to decrypt/encrypt one packet on MIPS R4000 processor at different voltages are given in Table I. The data for 3.3v are from [3], where they measured the time and energy required to perform a multiply function with 128-bit result, the basic building block for most public key algorithms, and estimated the number of such multiply will be used in each algorithm. (Interested readers can refer to [3] for details.) The other data are calculated based on the power-voltage and delay-voltage relations in Section II.

voltage (volt)	3.3		2.4		1.2	
	t(ms)	E(mJ)	t(ms)	E(mJ)	t(ms)	E(mJ)
Decryption	72.7	16.7	107.6	8.8	290.4	2.2
Encryption	3.5	0.81	5.2	0.43	14.1	0.11

TABLE I
TIME AND ENERGY CONSUMPTION FOR RSA DECRYPTION AND ENCRYPTION AT DIFFERENT VOLTAGES.

B. Simulation Results and Analysis

For the traditional system with a fixed 3.3v supply voltage, decryption takes $351.9s$, data processing takes $286s$, encryption takes $13.8s$, and the system is idle for the rest of the time. The energy consumption is $149J$ when we assume that the system will shut down during the idle period and will not consume any energy, dynamic or static.

The DVS-enabled systems take advantage of their flexibility on changing voltages to operate at the lowest level for each phase of the message processing. Table II reports various DVS-enabled system's total amount of non-idle time and their energy consumption. We see that as we go from single voltage system, to the 2- and 3-level voltage systems, to the pessimistic and the optimistic feasible DVS systems, and eventually to the ideal DVS system, the processor's execution time goes up from $652s$ to $3596s$ while the dynamic energy consumption drops down from $149J$ to $32J$. More specifically, with the addition of a low voltage 2.4v, the 2-level system saves about 36% energy over the fixed 3.3v system. The use of an even lower voltage 1.2v saves 31% more. When we add two more voltages 2.8v and 1.8v, there is another 5% dynamic energy saving over the fixed 3.3v system. This indicates that the gain of introducing new voltage levels diminishes quickly. The feasible DVS systems can save only less than 10% further because the lowest voltage in the 3-level system is very close the physical minimum voltage $v_{min} = 1.1v$. The ideal DVS system consumes $32.17J$, about 22% less than the dynamic energy required by the 3.3v system. This confirms our conclusion that the less constraint we have on voltage/speed, the more energy we can save by DVS. The optimistic and pessimistic feasible DVS-enable systems have very similar behavior because the voltage transition time ($25\mu s$ for the switch from the lowest performance to the highest performance) is almost negligible compared to the execution time for message decryption and data process.

The static power has little impact in our simulation. This is mainly because that the MIPS R4000 processor, like other microprocessors used in sensor nodes, has extremely low power dissipation when the system is not in the run mode. For example, the Intel StrongARM SA-1110 in the sensor nodes developed by the Rockwell Science Center has static power below 1mW in sleep mode [22]. The fixed voltage system has the shortest time in run mode, which is $652s$ as shown in Table II. Assuming it shuts down whenever the system is idle to avoid static power dissipation, the most static energy saving over other DVS systems will be less than $3J$. Therefore, we consider only the dynamic power in our study. Static power

	fixed	2-level	3-level	pessimistic	optimistic	ideal
voltages (volt)	3.3	{3.3, 2.4}	{3.3, 2.4, 1.2}	any value in [1.1, 3.3]		$[0, \infty)$
non-idle time(s)	652	893	2005	2351	2376	3596
dynamic energy(J)	149.08	95.47	48.88	36.28	36.17	32.17
static energy(J)	0.65	0.89	2.01	2.35	2.38	3.60
total energy(J)	149.73	96.36	50.89	38.63	38.55	35.77

TABLE II

TIME AND ENERGY CONSUMPTION FOR A ONE-HOUR SIMULATION ON DIFFERENT DVS-ENABLED SYSTEMS.

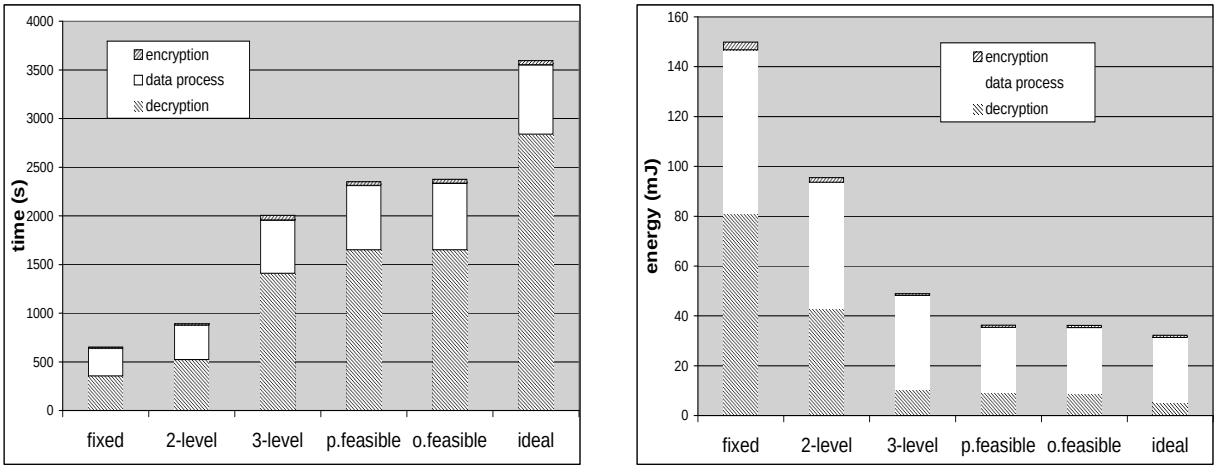


Fig. 2. Break-down: time and energy consumption for decryption, data process, and encryption. *Trend: as we go from fixed voltage system to 2-level, 3-level, feasible, and ideal DVS-enabled systems (left to right), execution time increases and energy consumption decreases.*

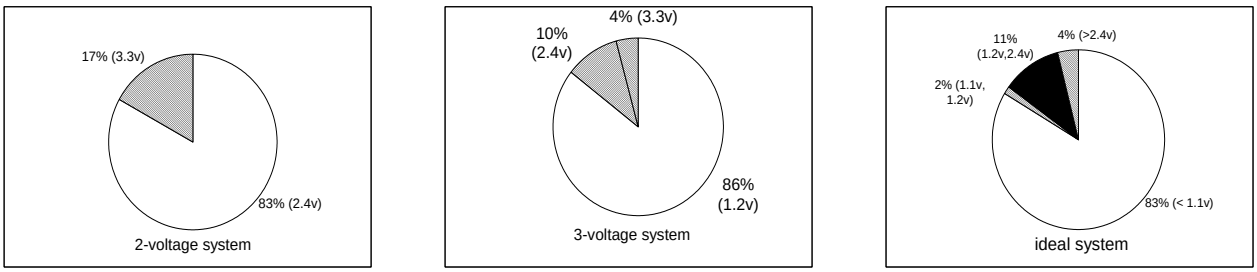


Fig. 3. Break-down: execution time at different voltages (total execution time: $893s$ for 2-voltage, $2005s$ for 3-voltage, and 3596 for ideal systems).

may play a more important role for other applications. In that case, one should focus the analysis on total power.

To see the importance of the proposed feasible DVS models, we observe that the energy consumption by the optimistic feasible model is 36.17 and that by the ideal model is 32.17. Their difference is surprisingly small considering the fact that arbitrarily fast/slow speed can be used in the ideal model. However, this small difference still represents more than 10% of the dynamic energy consumption by the optimistic feasible model. This means that although the ideal model can provide us a good estimation of DVS technique's energy reduction potential, it can be too optimistic at times. For example, if a design has already achieved a dynamic energy consumption of 37 in the same simulation, should the designer push further for more energy reduction by DVS (as 37 is still 15% more than 32.17)? The answer probably is 'NO' should the designer know that the optimistic feasible model gives the bound 36.17. The same argument applies for the pessimistic feasible model vs. multiple DVS model. In sum, without the feasible DVS models, we can only tell that the energy consumption of the best DVS-enabled system is bounded by 32.17 from the ideal model and 48.88 from the 3-level DVS system. We can scale this interval [32.17, 48.88] to [1, 1.52], where $1.52 \approx \frac{48.88}{32.17}$. With the optimistic and pessimistic feasible DVS models, we can improve this to [36.17, 36.28] or [1, 1.003]. This implies that the proposed feasible DVS models give a much more precise prediction on the dynamic energy reduction by DVS and could provide valuable guidance for designers.

In Figure 2, we break the total execution time and energy consumption into the time and energy spent on message decryption, data process, and (new) message encryption. We see the clear trend that energy consumption drops as the execution time goes up for each phase. Take the message decryption phase for example, it accounts about 54% of the total energy (and also time) of the fixed 3.3v system. However, this percentage drops to 45%, 22%, and less than 16% for the 2-level, 3-level multiple systems and the ideal DVS-enabled system as they allocate more and more time on decryption at lower voltages.

DVS-enabled systems operate the extended execution time

at low voltages. For instance, in the total $3596s$ when the ideal system executes during the simulation, it operates $3002s$ under 1.1v (mainly on decryption); $54.6s$ between 1.1v and 1.2v; $401s$ between 1.2v and 2.4v; and only $138s$ at 2.4v or higher. Figure 3 depicts this together with similar percentage for the 2-level and 3-level systems. It reveals that the most energy efficient strategy is to use low voltage as long as possible and switch to the higher voltage(s) only when necessary in cases such as hard deadlines to forward/send new messages.

VI. Conclusion

Dynamical voltage scaling is an effective technique for power/energy reduction. But how effective it can be? In particular for the newly implemented systems that can change the operating voltage at run time with some physical constraints. We build the model to analyze the energy saving on DVS-enabled systems. Although the general problem is NP-hard, we restrict our discussion to a simple case and provide the upper bounds of energy reduction under different DVS models. The solution can be extended to the more general case at the cost of solving a non-linear system. Simulation results from secure wireless communication networks demonstrate that DVS is effective in energy saving, and the introduce of feasible DVS models provides practical guidelines to design low power DVS-enabled system.

Acknowledgement

The authors thank the editor-in-chief, the associated editor, and the reviewers for their valuable comments and suggestions.

References

[1] T.D. Burd, T. Pering, A. Stratakos, and R. Brodersen. "A Dynamic Voltage-Scaled Microprocessor System," *ISSCC'00: IEEE International Solid-State Circuits Conference*, pp. 294-295, 466, February 2000.

[2] T.D. Burd and R.W. Brodersen. "Design Issues for Dynamic Voltage Scaling," *ISLPED'00: International Symposium on Low Power Electronics and Design*, pp. 9-14, July 2000.

[3] D.W. Carman, P.S. Kruus, and B.J. Matt, "Constraints and Approaches for Distributed Sensor Network Security," *NAI Labs Technical Report #00-10*, September 2000.

[4] A.P.Chandrakasan, W. Bowhill, and F. Fox. *Design of High-Performance Microprocessor Circuits*, IEEE Press, 2001.

[5] C. Chen and M. Sarrafzadeh. "Provably Good Algorithm for Low Power Consumption with Dual Supply Voltages," *ICCAD'99: IEEE/ACM International Conference on Computer-Aided Design*, pp. 76-79, November 1999.

[6] V. De and S. Borkar. "Technology and Design Challenges for Low Power and High Performance," *International Symposium on Low Power Electronics and Design*, pp. 163-168, 1999.

[7] I. Hong, G. Qu, M. Potkonjak, and M.B. Srivastava. "Synthesis Techniques for Low-Power Hard Real-Time Systems on Variable Voltage Processor", *19th IEEE Real-Time Systems Symposium*. pp. 178-187, December 1998.

[8] I. Hong, et al. "Power Optimization of Variable-Voltage Core-Based Systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol.18, No.12, pp. 1702-1714, December 1999.

[9] S. Hua and G. Qu. "Approaching the Maximum Energy Saving on Embedded Systems with Multiple Voltages," *IEEE/ACM International Conference on Computer-Aided Design*, pp. 26-29, 2003.

[10] T. Ishihara and H. Yasuura. "Voltage Scheduling Problem for Dynamically Variable Voltage Processors," *ISLPED'98: International Symposium on Low Power Electronics and Design*, pp. 197-202, August 1998.

[11] N. Kim, T. Austin, D. Blaauw, T. Mudge, K. Flautner, J. Hu, M.J. Irwin, M. Kandemir, and V. Narayanan, "Leakage Current: Moore's Law Meets Static Power," *Computer*, Vol. 36, No.12, pp. 68-75, 2003.

[12] P. Macken, M. Degrauwe, M. Van Paemel, and H. Oguey. "A voltage reduction technique for digital systems," *IEEE International Solid-State Circuits Conference*, pp. 238-239, 1990.

[13] B. Mochocki, X. Hu, and G. Quan. "A realistic variable voltage scheduling model for real-time applications," *IEEE/ACM International Conference on Computer-Aided Design*, pp. 726-731, 2002.

[14] T. Pering, T.D. Burd, and R.W. Brodersen. "Voltage Scheduling in the IpARM Microprocessor System," *ISLPED'00: International Symposium on Low Power Electronics and Design*, pp. 96-101, July 2000.

[15] G. Qu and M. Potkonjak. "Techniques for Energy-Efficient Communication Pipeline Design", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 10, No. 5, pp. 542-549, October 2002.

[16] G. Quan and X. Hu. "Energy Efficient Fixed-Priority Scheduling for Real-Time Systems on Variable Voltage Processors," *DAC'01: IEEE/ACM Design Automation Conference*, pp. 828-833, June 2001.

[17] S. Raje and M. Sarrafzadeh. "Variable voltage scheduling," *ISLPED'95: International Symposium on Low Power Electronics and Design*, pp. 9-14, April 1995.

[18] T. Sakurai, et al. "Low-power CMOS design through v_{th} control and low-swing circuits," *International Symposium on Low Power Design*, 1998.

[19] V. Swaminathan and K. Chakrabarty. "Generalized Network Flow Techniques for Dynamic Voltage Scaling in Hard Real-Time Systems," *IEEE/ACM International Conference on Computer-Aided Design*, pp. 21-24, 2003.

[20] F. Yao, A. Demers, S. Shenker. "A Scheduling Model for Reduced CPU Energy," *FOCS'95: IEEE Annual Foundations of Computer Science*, pp. 374-382, 1995.

[21] Y. Zhang, X. Hu, and D. Chen. "Energy Minimization of Real-time Tasks on Variable Voltage Processors with Transition Energy Overhead," *Proc. ASP-DAC*, pp. 65-70, 2003.

[22] *http://wins.rsc.rockwell.com/*.

Appendix

Proof of Theorem IV.3

Theorem IV.3 gives the unique speed function that satisfies Equations (3)-(7) and minimizes the energy consumption to complete a given workload $W \in [W_{min}, W_{max}]$. We have showed how to derive this speed function in Section IV.B. Now we give a formal proof of its optimality and uniqueness.

Let $s(t)$ be the speed function to complete workload W with the minimal energy consumption. Without loss of generality, we assume that $W_1 \leq W \leq W_2$ where $W_1 = min\{s_0, s_1\}(t_2 - t_1) + \frac{(s_1 - s_0)^2}{2K}$ and $W_2 = max\{s_0, s_1\}(t_2 - t_1) - \frac{(s_1 - s_0)^2}{2K}$ as given in Theorem IV.3. Suppose that $\tilde{s}(t) : [t_1, t_2] \rightarrow [s_{min},\ s_{max}]$ is another speed function that completes workload W and satisfies Equations (3)-(7). Since $\tilde{s}(t_1) = s_0 = s(t_1)$, there must exist $\tilde{t} \geq t_1$ such that $\tilde{s}(x) = s(x)$ for all $t_1 \leq x \leq \tilde{t}$ and $\tilde{s}(\tilde{t} + \delta) \neq s(\tilde{t} + \delta)$ for any small $\delta > 0$. We assume that, for example, $\tilde{s}(\tilde{t} + \delta) < s(\tilde{t} + \delta)$. Then we draw a horizontal line representing the constant speed $\tilde{s}(\tilde{t})$ which first intersects the speed function $\tilde{s}(t)$ at $\tilde{t} + \Delta t$ for some $\Delta t > 0$. Denote the difference of the amount of work done by $\tilde{s}(t)$ and by the constant speed $\tilde{s}(\tilde{t}) = s(\tilde{t})$ during period $[\tilde{t}, \tilde{t} + \Delta t]$ by

$$A(\Delta t) = \int_{\tilde{t}}^{\tilde{t}+\Delta t} [\tilde{s}(\tilde{t}) - \tilde{s}(t)]dt \tag{8}$$

Because we have assumed that $\tilde{s}(x) < s(x)$ for x close to and greater than $\tilde{t}$, there must exist $t' \in [t_1, t_2]$ such that $\tilde{s}(t') > s(t')$. Otherwise, $\tilde{s}(t) \leq s(t)$ for all $t \in [t_1, t_2]$ and they cannot complete the same amount of workload. For $t'_1 < t'$, we draw a horizontal line at the constant speed $\tilde{s}(t'_1)$ and let t'_2 be the first time that this line intersects $\tilde{s}(t)$ after t'. Define

$$A_1(t'_1) = \int_{t'_1}^{t'_2} [\tilde{s}(t) - \tilde{s}(t'_1)]dt \tag{9}$$

The continuity of functions $\tilde{s}(t)$ and therefore $A(\Delta t)$ and $A_1(t'_1)$ guarantees that we can find $\Delta t > 0$ and $t'_1 < t' < t'_2$ such that $A(\Delta t) = A_1(t'_1)$ as depicted in Figure 4.

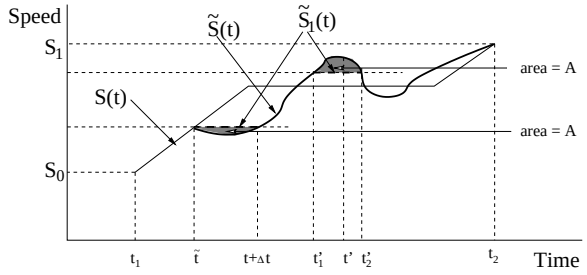


Fig. 4. The construction of a new speed function $\tilde{s}_1(t)$.

Now we define $\tilde{s}_1(t) : [t_1, t_2] \rightarrow [s_{min}, s_{max}]$, and claim that it satisfies Equations (3)-(7) while consuming less energy than $\tilde{s}(t)$:

$$\tilde{s}_1(t) = \begin{cases} \tilde{s}(t), & \text{if } t_1 \le t \le \tilde{t}; \\ \tilde{s}(\tilde{t}), & \text{if } \tilde{t} < t \le \tilde{t} + \Delta t. \\ \tilde{s}(t), & \text{if } \tilde{t} + \Delta t \le t \le t_1'; \\ \tilde{s}(t_1'), & \text{if } t_1' < t \le t_2'; \\ \tilde{s}(t), & \text{if } t_2' < t \le t_2. \end{cases}$$

$\tilde{s}_1(t_1) = \tilde{s}(t_1) = s_0$

$\tilde{s}_1(t_2) = \tilde{s}(t_2) = s_1$

$s_{min} \le \min\{\tilde{s}(t)\} \le \tilde{s}_1(t) \le \max\{\tilde{s}(t)\} \le s_{max}$

$|\frac{d\tilde{s}(t)}{dt}| \le K$ because $\frac{d\tilde{s}(t)}{dt} = \frac{ds(t)}{dt}$ for $t \in [t_1, \tilde{t} - 0] \cup [\tilde{t} + \Delta t + 0, \le t_1' - 0] \cup [t_2' + 0, t_2]$ and $\frac{d\tilde{s}(t)}{dt} = 0$ for $t \in [\tilde{t} + 0, \tilde{t} + \Delta t - 0] \cup [t_1' + 0, t_2' - 0]$

$$\begin{aligned} \int_{t_1}^{t_2} \tilde{s}(t)dt &= (\int_{t_1}^{\tilde{t}} + \int_{\tilde{t}}^{\tilde{t}+\Delta t} + \int_{\tilde{t}+\Delta t}^{t_1'} + \int_{t_1'}^{t_2'} + \int_{t_2'}^{t_2})\tilde{s}(t)dt \\ &= (\int_{t_1}^{\tilde{t}} + \int_{\tilde{t}+\Delta t}^{t_1'} + \int_{t_2'}^{t_2})s(t)dt + \int_{\tilde{t}}^{\tilde{t}+\Delta t} \tilde{s}(\tilde{t})dt + \int_{t_1'}^{t_2'} \tilde{s}(t_1')dt \\ &= \int_{t_1}^{t_2} s(t)dt + \int_{\tilde{t}}^{\tilde{t}+\Delta t}[\tilde{s}(\tilde{t}) - s(t)]dt + + \int_{t_1'}^{t_2'}[\tilde{s}(t_1') - s(t)]dt \\ &= \int_{t_1}^{t_2} s(t)dt + A(\Delta t) - A_1(t_1') \\ &= \int_{t_1}^{t_2} s(t)dt = W \end{aligned}$$

Finally, $\tilde{s}_1(t)$ consumes less energy than $\tilde{s}(t)$ because of the convexity of the power/energy function. □

Proof of Lemma IV.2

Lemma IV.2 is on the maximal workload W_{max} achievable on a pessimistic feasible DVS system. (The minimum workload accumulated on such system is $W_{min} = 0$. One can easily see this by constantly changing the speed without reaching any steady state.) We want to prove the following argument for the calculation of W_{max}:

> *if s is the system's highest steady speed during interval $[t_1, t_2]$, then the maximum workload W_{max} is achieved when the system has s as its only steady state and operates at s for the maximal period allowed by the constraints of Equations (3)-(6).*

Suppose that a speed function assumes constant speed s_i from time p_i to p_i' and changes from s_i to s_{i+1} from time p_i' to p_{i+1} for $t_1 = p_1 \le p_1' < p_2 < p_2' < \cdots < p_n \le p_n' = t_2$. Since workload is accumulated only during periods $[p_i, p_i']$ when the system is stable, we have the following for the workload completed by this speed function:

$$\begin{aligned} \int_{t_1}^{t_2} s(t)dt &= \sum_{i=1}^{n} \int_{p_i}^{p_i'} s(t)dt = \sum_{i=1}^{n} s_i \cdot (p_i' - p_i) \\ &\le \max\{s_1, s_2, \cdots, s_n\} \cdot \sum_{i=1}^{n} (p_i' - p_i) \end{aligned}$$

This implies that a speed function completes the most workload only if it has one steady state. Let s be the constant speed at such state. It will take the system at least $\frac{s - s_0}{K}$ to reach this speed and at least $\frac{s - s_2}{K}$ to slow down to the required ending speed. Therefore, the system operates at constant speed s for at most $t_2 - t_1 - \frac{s - s_1}{K} - \frac{s - s_2}{K}$, which gives the maximal workload with speed s in the amount of

$$W_{max}(s) = \int_{t_1}^{t_2} s(t)dt = s \cdot (t_2 - t_1 - \frac{s - s_1}{K} - \frac{s - s_2}{K})$$

Clearly, $W_{max}(s)$ achieves its maximum only when $s = \frac{K(t_2 - t_1)}{4} + \frac{s_1 + s_2}{4}$. Furthermore, $W_{max}(s)$ is monotonically increasing for speed less than this value. Hence, if $s > s_{max}$, the physical maximal speed, W_{max} is achieved when $s = s_{max}$. In sum, $s = \min\{s_{max}, \frac{K(t_2 - t_1)}{4} + \frac{s_0 + s_1}{4}\}$ and $W_{max} = (t_2 - t_1 + \frac{s_0 + s_1}{K})s - \frac{2s^2}{K}$ can be achieved by, for example, the following speed function

$$s(t) = \begin{cases} s_1 + K(t - t_1), & \text{if } t_1 \le t \le t_1 + \frac{s - s_1}{K}; \\ s, & \text{if } t_1 + \frac{s - s_1}{K} < t \le t_2 - \frac{s - s_2}{K}; \\ s - K(t_1 - t), & \text{if } t_2 - \frac{s - s_2}{K} < t \le t_2. \end{cases}$$

□

Proof of Theorem IV.4

Theorem IV.4 gives the existence and uniqueness of the most energy efficient speed function for the pessimistic feasible DVS system to complete a given workload. The idea to prove this is similar to the one we have used to prove Theorem IV.3. We give the outline here but omit the detailed formal proof for brevity.

The speed function for the pessimistic feasible DVS system with boundary constraints (3) and (4) also falls into the quadrilateral region as shown in Figure 1(a). Note that although the speed function contributes to the workload only when it stays at a constant speed, energy dissipation occurs both at such period and during the speed transition.

In the proof of Lemma IV.2, we have showed that the most workload will be completed only if the system has one steady speed. Similarly, one can prove that the most energy efficient way to complete a give workload is also to have one steady speed rather than having multiple steady speeds (due to the convexity of power/energy function).

Consider that we draw a horizontal line and move it up from s_{min} to the speed s given in the proof of Lemma IV.2 and then to s_{max}. The completed workload monotonically increase and reaches W_{max} as the line meet s. (When the steady speed exceeds s, the completed workload decreases monotonically.). When this workload equals to the given workload, we can prove that the speed function as shown at the end of the proof of Lemma IV.2 is the most energy efficient and it is unique from this construction.

□

Lin Yuan received the B. Eng. degree in information engineering from Xi'an Jiaotong University, China, in 2001. He is currently working toward the Ph.D. degree in the Department of Electrical and Computer Engineering at the University of Maryland, College Park. His research interests include low power design and synthesis, intellectual property protection, and the Satisfiability problem.

Gang Qu received his B.S. and M.S. degrees in Mathematics from the University of Science and Technology of China in 1992 and 1994 respectively. In 2000, he received his Ph.D. degree in Computer Science from the University of California, Los Angeles and joined the Electrical and Computer Engineering Department in the University of Maryland, College Park. He became a member of the University of Maryland Institute of Advanced Computer Studies in 2001. He has won several awards including the 36th Design Automation Conference Graduate Scholarship, the Dimitris N. Chorafas Foundation Award, and 2002 George Corcoran Award for Significant Contributions to ECE Education. His research interests include intellectual property reuse and protection, low power system design, applied cryptography, and computer-aided synthesis.