Techniques for Energy-Efficient Communication Pipeline Design

Gang Qu and Miodrag Potkonjak

***Abstract*—The performance of many modern computer and communication systems is dictated by the latency of communication pipelines. At the same time, power/energy consumption is often another limiting factor in many portable systems. We address the problem of how to minimize the power consumption in system-level pipelines under latency constraints. In particular, we apply fragmentation technique to achieve parallelism and exploit advantages provided by variable voltage design methodology to optimally select voltage and, therefore, speed of each pipeline stage. We focus our study on the practical case when each pipeline stage operates at a fixed speed. Unlike the conventional pipeline system, where all stages run at the same speed, our system may have different stages running at different speeds to conserve energy while providing guaranteed latency. For a given latency requirement, we find explicit solutions for the most energy efficient fragmentation and voltage setting. We further study a less practical case when each stage can dynamically change its speed to get further energy saving. We define the problem and transform it to a nonlinear system whose solution provides a lower bound for energy consumption. We apply the obtained theoretical results to develop algorithms for power/energy minimization of computer and communication systems. The experimental result suggests that significant power/energy reduction, is possible without additional latency. In fact, we achieve almost 40% total energy saving over the combined minimal supply voltage selection and system shut-down technique and 85% if none of these two energy minimization methods is used.**

***Index Terms*—Energy minimization, latency, low-power design, pipeline.**

I. Introduction

SYSTEM level pipelines are widely acknowledged as the most likely bottleneck of many computer systems. For example, a read miss in the system data or instruction cache will block the application program until the entire block with requested data arrives [1], [23]. The tradeoff is clear: longer blocks imply fewer misses, but also longer interrupt latency. Similarly, in high-speed local and wide-area networks, selecting proper block size to exploit intrinsic concurrency in communication pipelines is a key issue [7], [27]. As the final example, where communication pipelines dictate performances we mention path-oriented operating systems [16]. Therefore, it is not surprising that recently the question of how to improve the performance of a system pipeline received a great deal of attention in computer architecture, operating systems and compilers communities. The essence of the problem is abstracted in a recent work [24] where the discussion is on how to minimize the transmission latency by careful packet fragmentation.

On the other hand, the increasing use of portable systems (such as personal computing devices, wireless communications and imaging systems) makes the power consumption one of the primary circuit and system design goals. The most effective method to reduce power consumption is to lower the supply voltage level, which exploits the quadratic dependence of power on voltage [5]. However, reducing the supply voltage increases circuit delay and decreases the clock speed. The resulting processor core consumes lower average power at the cost of increased latency. Therefore, it becomes less effective when tight deadlines are present.

Recent progress in power supply technology along with custom and commercial CMOS chips that are capable of operating reliably over a range of supply voltages makes it possible to build processor cores with supply voltages that can be varied at run time according to the application latency constraints [3], [17]. The variable voltage processor core is capable of operating at different optimal points along the power and speed curve in order to achieve high energy efficiency. In particular, with multiple supply voltages on the chip, the processor core can use high voltage for applications with tight deadlines and keep the voltage low otherwise to reduce total energy consumption [3], [22].

In this paper, we address the energy minimization problem in system-level pipelines under latency constraints. We use the recent advances in power supply technologies and the variable voltage design methodology to choose a voltage profile for each pipeline stage, which optimally minimizes the energy consumption of the entire pipeline system.

The rest of the paper is organized as follows. Section II describes the related work in communication pipeline and low power design techniques. In Section III, we discuss the pipeline model, processor model and formulate the problem. We solve the problem optimally in two cases: 1) each pipeline stage has a fixed voltage, which may vary from stage to stage and 2) every stage can have variable supply voltages (detailed proof, example and discussion can be found in the technical report [19]). We present the experimental results in Section VI, and Section VII concludes.

II. Related Work

The most relevant related work are efforts in communication pipeline design and evaluation and low power design tech-

Manuscript received February 1, 2001; revised January 7, 2002. This work was supported in part by the National Science Foundation under Grant 9734166.

G. Qu is with the Electrical and Computer Engineering Department and Institute of Advanced Computer Study, University of Maryland, College Park, MD 20742 USA (e-mail: gangqu@eng.umd.edu).

M. Potkonjak is with the Computer Science Department, University of California, Los Angeles, CA 90095 USA (e-mail: miodrag@cs.ucla.edu).

Digital Object Identifier 10.1109/TVLSI.2002.800522

niques. In particular, within the former domain fragmentation techniques for managing congestion control, packet buffering, packet losses, and the optimization techniques for improvement of distributed file systems and high-speed local networks are directly relevant. Within the latter, we focus our survey on system-level power minimization techniques and variable voltage techniques.

In the introduction, we already surveyed a number of communication pipeline systems and research efforts for latency optimization of these systems. It is important to note that many application specific systems operate at the highest-level of abstraction as processing pipelines on blocks of input (e.g., digital TV and audio and segmentation subsystems of communication devices). Apparently, fragmentation has been used in design of the Internet for quite a long time. More recently, studies on how to exploit flexible block fragmentation to improve performances of DEC workstations has been also conducted [12]. More detailed survey of fragmentation techniques is given in [24].

Dynamically adapting voltage and therefore the clock frequency, to operate at the point of lowest power consumption for given temperature and process parameters was first proposed by Macken *et al.* [13], [15]. Later, [11], [26] described implementation of several digital power supply controllers based on this idea. Several researchers have recently developed efficient dc-dc converters that allow the output voltage to be rapidly changed under external control [17], [21]. We mention that a dynamic voltage-scaled microprocessor system has been reported recently [3] and leave further discussion on variable voltage processor for the next section.

In the software world, there has been also recent research on scheduling strategies for adjusting CPU speed so as to reduce power consumption. For example, Weiser *et al.* [25] proposed an approach where time is divided into 10–50 ms intervals and the CPU clock speed (and voltage) is adjusted by the task-level scheduler based on the processor utilization over the preceding interval. Govil *et al.* [9] concluded that smoothing helps more than prediction in voltage changing. Yao *et al.* [29] described an off-line minimum-energy schedule and an average rate heuristic for job scheduling for independent processes with deadlines, though under the assumptions that 1) the processor can change its speed arbitrarily, i.e., the changes are instantaneously with no physical bounds and 2) the jobs are preemptive with no preemption penalty. Qu [18] extended this by the discussion of both nonpreemptive jobs on such ideal variable speed processor and general jobs on real variable speed processors where both the maximal/minimal voltage constraints and the limitation on speed changes are considered. Survey on system-level low power techniques can be found in [28] and energy efficient microprocessor design has been discussed in [2] and [8].

III. Background and Problem Formulation

In this section, we first describe the variable voltage processor and the store-and-forward pipelining network, then characterize the user packet and formulate the problem.

A. Variable Voltage Processor

The variable voltage is generated by the dc-dc switching regulators. Time to reach steady state at a new voltage is normally innegligible. However, recent work on dc-dc converters allows the output voltage to be changed rapidly. For example, Burd *et al.* [3] implemented a microprocessor system that consist of a dc-dc switching regulator, an ARM V4 microprocessor, a bank of SRAM ICs, and an interface IC. The supply voltage and clock frequency can be dynamically varied from 1.2 V to 3.8 V within 70 μs with an energy efficiency of 0.54–5.6 mW/MIP.

To compensate the complexity of real variable voltage system, we have seen plenty of efforts in the following two directions. On one hand, there have been many proposal and implementation of multiple supply voltage systems [6], [14], [20], [22]. These research groups have addressed the use of two or three discrete supply voltages. The idea is to switch among these simultaneously available voltages according to the processing load, computation requirement, latency constraint, etc. On the other hand, ideal variable voltage system has also been studied theoretically [18], [29]. An ideal variable voltage processor can change its speed from zero to ∞ instantaneously without any overhead. Apparently, such ideal processor is not feasible, but the study of this model gives us insightful view of the problem and more importantly, it provides the lower bound of energy consumption by using variable voltage processors. Although there is no reported studies that takes these overheads into consideration, there exist evidence showing that this bound could be tight. First, Hong *et al.* [10] reported a task scheduling heuristics which, when applied to multimedia benchmarks, results in a total energy consumption only 1.5% higher on average than the lower bound obtained from the ideal case. Furthermore, Burd and Brodersen [4] discussed various design issues for dynamic voltage scaling systems. In their prototype design, it takes 26 μs and 6.5 μJ for a full-scale transition from 1.2 V to 3.8 V. They estimated a practical limit of voltage change rate on the order of 5 V/μs, with the potential of going as high as 20 V/μs, for 0.6 μm process. This will further reduce the transition time and energy.

With different supply voltages, the processor will be able to operate at different speeds, the time and power consumed to execute the same task (or same amount of computation) will also be different. We adopt the following relationships among the voltage, delay, power, and energy [5]: Suppose with a 5-V constant supply voltage, the processor finishes a task in time $T(5)$, the power dissipation is $P(5)$. Then, with a supply voltage v, to finish the same task, the processing time $T(v)$, the power dissipation $P(v)$ and the energy to complete the task $E(v)$ are given as follows:

$$T(v) = \frac{v}{(v-v_t)^2}\frac{(5-v_t)^2}{5}T(5) \tag{1}$$

$$P(v) = v(v-v_t)^2\frac{1}{5(5-v_t)^2}P(5) \tag{2}$$

$$E(v) = P(v)t(v) = \frac{1}{25}v^2P(5)T(5) \tag{3}$$

where v_t is the threshold voltage.

B. Pipeline Model

As proposed in [24], we represent the network as a sequence of store-and-forward pipeline stages characterized by the following parameters:

- n is the number of pipeline stages;
- g_j is the fixed per-fragment overhead for stage j;
- $T_j(5)$ is the per-byte transmission time for stage j with the 5-V reference supply voltage.

The fixed per-fragment overhead, g_i, can be considered as the context switch time and may vary from stage to stage. If none of the stages has overhead, the best strategy, as we will show soon, is to fragment the packet as small as possible to utilize parallelism. $T_j(5)$ is proportional to the inverse of the bandwidth for stage j with a $5v$ supply voltage. In the extreme case, if there is no bandwidth limitation for all stages, to achieve the minimum latency the entire packet should be sent as a single fragment to avoid the per-fragment overhead.

At the sender's end, the packet is fragmented and sent to the first stage of the pipeline. A pipeline stage will start the transmission of a fragment as soon as it receives both the entire fragment from the previous stage and an acknowledgment from the next stage, which is sent when the next stage is ready for the reception of the current fragment. We refer to these as the *rules for transmission*. The transmission is completed when the receiver's end receives the last fragment of the packet.

C. Problem Formulation

Our objective is to minimize the power consumption for transmitting a packet through the network under the user-specified latency constraint. The two tools that we use to achieve this are packet fragmentation and supply voltage selection. The following variables are associated with the packet for the convenience of analysis:

- $B$is the size of the entire packet;
- T is the deadline to transmit the entire packet;
- $k$is the number of fragments;
- x_i is the size of the ith fragment ($0 \leq i \leq k-1$);
- $t_{i,j}$: (life) time that the ith fragment stays on the jth stage.

The packet's size B and the deadline T are given by the user, the network is characterized by the number of pipeline stages n, the overhead g_j, and the unit transmission time $T_j(5)$ for each pipeline stage. We further assume that the processors at all stages are identical (i.e, with the same $T(v)$, $P(v)$ and $E(v)$). A fragment's lifetime on a stage is the sum of the per-fragment overhead and the actual transmission time on this stage. Let $v_j(t)$ be the voltage at which the jth processor operates at time $t \in [0,T]$, then the processor's energy consumption is

$$E_j = \int_0^T P\left(v_j(t)\right) dt \tag{4}$$

where $P(v)$ is the power dissipation at supply voltage v. We want to minimize $E = \sum_{j=0}^{n-1} E_j$ by finding the best voltage and fragment schemes. Formally, we seek solutions for the energy minimization with deadline on variable voltage (EMDVVP) problem.

Instance: A pipeline with parameters n (number of pipeline stages), g_j (per fragment overhead on stage j) and $T_j(v_{ref})$ (per-byte transmission time on stage j at reference voltage), a packet with size B and transmission deadline T.

Question: Find the voltage scheme $v_j(t)$ for each processor and a fragment $\{x_0, x_1, \ldots\}$ of the packet, such that the entire packet is transmitted within T and the total energy consumption $E = \sum_{j=0}^{n-1} \int_0^T P(v_j(t))dt$ is minimized.

We explain our approach and give the main results with sketch of proof in the following sections, while interested readers can find detailed proof, example, and discussion in the technical report [19].

IV. Fixed Voltage Within the Same Stage

We first consider a simple case when the processor on each stage operates at a fixed voltage, but the voltages can be different from stage to stage. It is important to study this case because of the extreme simplicity of implementation. Since each processor will operate at a constant supply voltage, no additional hardware is required. Once the voltage level for each processor is determined, the pipeline can be easily set up by applying the required voltages to corresponding processors. The voltage scheme problem is reduced to finding the constant voltage v_j for the processor at stage j. The energy consumption on this stage, from (4), is simplified to $E_j = P(v_j)T$. Moreover, the lifetime that the ith fragment (with size x_i) stays on the jth stage can be expressed as

$$t_{i,j} = g_j + T_j\left(v_j\right) x_i. \tag{5}$$

Lemma 4.1: A necessary condition for the energy consumption to be minimized is to finish the transmission exactly at the deadline T.

sketch of the proof: Suppose that we have a packet fragmentation, a voltage scheme $\{v_0, v_1, \ldots, v_{n-1}\}$ where v_j is the constant voltage at which processor on the jth pipeline stage and the last fragment leaves the last stage before the deadline T, we show that this cannot be optimal by constructing another voltage scheme on the same packet fragmentation that consumes less energy.

We consider the voltage scheme $\{v_0, v_1, \ldots, v'_{n-1}\}$ where $v'_{n-1} < v_{n-1}$ is the reduced voltage on the last stage such that the transmission will complete on the deadline T. This clearly consumes less energy. However, we still need to verify the new voltage scheme does not violate *rules of transmission*. 1) With low voltage and, hence, slow transmission speed, each fragment will spend more time on the last stage. Therefore, the starting transmission time of each fragment will be no earlier than its original starting transmission time at v_{n-1}. This implies that we will not start transmitting a fragment that has not yet arrived. 2) The transmission cannot start until the next stage is ready for reception of a new fragment. The slow-down of the last stage will delay the transmission of previous stages. But this delay will not be longer than the delay on the last stage caused by lower voltage and therefore the deadline T will not be missed. Finally, the new voltage v'_{n-1} on the last stage can be easily determined. Let t be the time that the first fragment arrives at the last stage, then the total transmission time on this stage will be $T - t - k \cdot g_{n-1}$ (if there is no starving), where k is the number of fragments and g_{n-1} is the per-fragment overhead. For a packet

of size B, we select v'_{n-1} such that the per-byte transmission time becomes exactly $(T - t - k \cdot g_{n-1})/B$. $Q.E.D.$

Intuitively, Lemma 4.1 says that the pipeline will use as much time as possible for transmission such that the processors can be scheduled at low voltages and thus minimize energy consumption. On the other hand, on each single stage, the best strategy is to transmit a fragment immediately upon its reception and the accomplishment of sending the previous fragment. This implies that the voltage should be adjusted such that all stages are synchronized and leads to the following lemma.

Lemma 4.2: If the packet can only be fragmented into fixed size, then a voltage scheme $\{v_0, v_1, \ldots, v_{n-1}\}$ minimizes the energy consumption if and only if

$$t_{i,j} = g_j + T_j(v_j)\, x_i = constant \tag{6}$$

Sketch of the proof: When we restrict fragmentation to be equal-sized, $t_{i,j} = g_j + T_j(v_j)x_i = constant$ for all i when j is fixed, i.e., equal lifetime for all fragments on the same stage. We will show that these constants are the same for all stages by contradiction.

Suppose $t_{i,j}$'s are not the same, then there exists $l \in [0, n-1]$, such that either $t_{i,l} < t_{i,l-1}$, or $t_{i,l} < t_{i,l+1}$, or both. We can reduce the supply voltage on the lth stage and construct a better solution with less energy consumption. In fact, such solution can be found in four steps.

1) Find the smallest $t_{i,l}$ such that $t_{i,l} < t_{i,l-1}$ or $t_{i,l} < t_{i,l+1}$. (Assuming $t_{i,l} < t_{i,l-1}$ for simplicity).
2) Reduce v_l to v'_l such that $t_{i,l} = t_{i,l-1}$.
3) Make appropriate changes on the stages after the lth stage because of the delay of fragments by $T_l(v'_l)$.
4) Modify the voltage schemes to fit the deadline T.

The new solution will consume less energy. Therefore, any strategy with different $t_{i,j}$'s cannot be optimal. $Q.E.D.$

From (6), the processor at the stage that has the largest per-fragment overhead must operate at a high voltage to achieve a short per-byte transmission time $T_j(v_j)$. Therefore, this stage will consume more energy than other stages and we call such a stage *dominant stage* because it dominates the total energy consumption.

Theorem 4.3: Let stage d be the dominant stage, then there is a unique solution for the EMDVVP problem. The number of fragments is given by

$$k = \sqrt{\frac{T}{g_d}(n-1)} - (n-1) \tag{7}$$

and each stage will operate at a fixed supply voltage that can be determined by (6) with the constant on the right-hand side equals $\sqrt{T \cdot g_d/(n-1)}$.

Sketch of the proof: Let k be the number of fragments, $x = B/k$ be the size of each fragment for a packet of size B and $\{v_0, v_1, \ldots, v_{n-1}\}$ be an optimal voltage scheme. The time to transmit the entire packet is

$$\sum_{j=0}^{n-1} [g_j + T_j(v_j)\, x] + (k-1)\,[g_{n-1} + T_{n-1}(v_{n-1})\, x].$$

The first term is the time for the first fragment to travel through the entire pipeline. From Lemma 4.2, it equals to $n[g_j + T_j(v_j)x]$ for any j. The second term is the time to send the rest of the packet from the last stage. Lemma 4.1 requires their sum to be T for the energy to be minimized, therefore we have

$$(n + k - 1)\,[g_j + T_j(v_j)\, x] = T. \tag{8}$$

$T_j(v_j)$ can be easily solved in terms of k from (8) and considering (1), we get

$$\frac{v_j}{(v_j - v_t)^2}\frac{(5 - v_t)^2}{5} T_j(5) = \frac{k}{B}\frac{T - (n + k - 1)g_j}{n + k - 1}. \tag{9}$$

Solving this quadratic equation gives us, for a given number of fragments k

$$v_j = v_t + D_j(k) + \sqrt{D_j(k)^2 + 2v_t D_j(k)} \tag{10}$$

where

$$D_j(k) = \frac{1}{2}\frac{(5 - v_t)^2}{5}\frac{T_j(5)(n + k - 1)\frac{B}{k}}{T - (n + k - 1)g_j}.$$

Next, we plug the values of v_j's into (3) and get the total energy consumption, which is expressed in terms of k. Since the energy is dominated by stage d and we know that low voltage results in low energy. To find the optimal scheme, we take the first derivative of v_d with respect to k, set it to zero and get the unique solution (7).

It follows from (8) immediately that the constant on the right–hand side of (6) is $T/(n + k - 1) = \sqrt{T \cdot g_d/(n-1)}$. The voltage level on each stage can be easily determined from (6). $Q.E.D.$

Remarks: How do the network's parameters and the latency affect the optimal scheme?

- T: When the latency constraint is loose (i.e., T is large), (7) predicts more fragments. Energy consumption is reduced because each processor gets a long transmission time and thus can use low voltage.
- n: From (7), we see k is an increasing function with respect to n, the number of pipeline stages. This means that the more stages in the network, the more fragments we should have. This takes advantage of the parallelism.
- g_d: If the per-fragment overhead at the energy dominating stage is high, less fragments should be used to avoid a large total overhead. If there is no overhead, then we should fragment the packet as small as possible so that more parts of the packet can be transmitted in parallel.
- B: The number of fragments in the optimal scheme is independent of the packet size. However, B does play a very important role in the voltage scheme (10). This is not surprising, since we use the ideal variable voltage processor, which can adjust its speed (by changing supply voltage) according to the size of the packet.

To end this section, we show the following corollary.

Corollary 4.4: Theorem 4.3 holds for deep submicrometer.

Sketch of the proof: For deep submicrometer technology, voltage-delay is given by $T(v) \propto v/(v - v_t)^{\alpha}$ ($1 < \alpha < 2$, the current technology has α as 1.5 or 1.6). Recall

that (8) is independent of the voltage-delay model. When $T(v) \propto v/(v-v_t)^\alpha$, (9) will be replaced by

$$\frac{v_j}{(v_j-v_t)^\alpha}\frac{(5-v_t)^\alpha}{5}T_j(5) = \frac{k}{B}\frac{T-(n+k-1)g_j}{n+k-1}.$$

We rewrite this as $(v_j - v_t)^\alpha = D(k)v_j$. Differentiating both sides with respect to D, we get

$$\alpha(v_j-v_t)^{\alpha-1}\frac{\partial v_j}{\partial D(k)} = v_j + D(k)\frac{\partial v_j}{\partial D(k)}$$

and

$$\frac{\partial v_j}{\partial D(k)} = \frac{v_j(v_j-v_t)}{(\alpha-1)D(k)v_j + D(k)v_t} > 0$$

Therefore, $\partial v_j/\partial k = (\partial v_j/\partial D(k))(\partial D(k)/\partial k) = 0$ if and only if $\partial D(k)/\partial k = 0$ and the latter gives (7). $Q.E.D.$

V. Variable Voltages Within the Same Stage

We first explain how to transform the energy minimization problem to a nonlinear system and then discuss implementation challenges for variable voltage on the same stage.

A solution to the EMDVVP problem requires a supply voltage profile for each processor and a packet fragmentation. Suppose that there are n pipeline stages and the packet is cut into k fragments, an optimal solution to the general EMDVVP problem consists of n voltage vs time functions $v_j(t)$ for $j = 0, 1, \ldots n-1$ and the size of fragment x_i for $i = 0, 1, \ldots k-1$.

Power/energy is a convex function on supply voltage, so any best voltage scheme will not change voltage during the transmission of a fragment on a single stage. That is, we have the following lemma.

Lemma 5.1: In an optimal solution, the supply voltage changes either on the arrival of a new fragment or on the completion of transmission of the current fragment.

This outlines the shape of the voltage functions $v_j(t)$, which are step functions with all possible discontinuous points at the time when new fragment arrives or the current fragment leaves. Therefore, we only need to determine $(k \cdot n)$ constants $v_{i,j}$ $(i = 0, 1, \ldots k-1, j = 0, 1, \ldots n-1)$, the voltage for processor j to transmit fragment i.

Lemma 4.2 synchronizes all processors on a fixed length fragmentation such that no stage will congest or starve. We can generalize this for variable fragment size:

Lemma 5.2: The optimal voltage scheme, for a given fragmentation, provides the lifetime $t_{i,j}$ of the ith fragment on the jth stage such that for all $0 \le i \le k-2$ and $0 \le j \le n-2$, the following holds:

$$t_{i-1,j+1} = t_{i,j}. \tag{11}$$

This gives a recursive relationship among adjacent fragment's transmission time on adjacent stages. From the $(n-1)(k-1)$ such recursive formulas in (11), we can easily solve $(n-1)(k-1)v_{i,j}$'s.

Finally, there are two global constraints: the transmission deadline T and the packet size B. Therefore, there are $kn-(n-1)(k-1)-1 = (n+k-2)v_{i,j}$'s and $(k-1)x_i$'s, a total of $(n+2k-3)$ constants, need to be determined. We express the total energy consumption in terms of these $n+2k-3$ variables from (4) and the EMDVVP problem becomes equivalent to finding the minimal of this function. Applying the first-order condition, we will have a nonlinear system with $(n+2k-3)$ variables where the nonlinearity comes from the nature of the power model.

Theorem 5.3: Given the number of fragments k, the EMDVVP problem with n pipeline stages is reduced to solving a nonlinear system with $n+2k-3$ free variables.

Unlike the easy-to-implement pipeline systems with fixed voltage on the same stage, system with variable voltage on the same stage introduces many implementation challenges: What is the most energy efficient way to change voltage? With a dynamically changed supply voltage (and, therefore, clock frequency), what is the system's performance? The extra hardware (e.g., the dc-dc switching regulator) that enables the variable voltage also consumes power, how should the solution change if we take this into consideration? Based on a simplified model, Qu [18] describes how to dynamically vary voltage to minimize the energy for a give task. The more practical multiply supply voltage systems have been reported. For example, the dual supply voltage media processor, graphic controller LSI and a MPEG4 codec core. Many implementation issues (placement, routing, synchronization, etc) and empirical power reduction of the system have also been addressed [19], [22].

TABLE I
Myrinet GAM Pipeline Parameters

stage j	$g_j(\mu s)$	$T_j(\mu s/\text{KB})$	$P_j(\text{Watt})$
0	7.2	7.2	P_0
1	5.2	24.9	P_1
2	7.5	24.9	P_2
3	7.4	7.9	P_3

VI. Simulation Results

In this section, we report the results when applying our new energy minimization approach on several pipeline models, in particular the Myrinet GAM pipeline that researchers in Berkeley adopted to study the transmission latency minimization by variable sized fragmentation [24].

Myrinet GAM pipeline consists of four stages, stage 0 copies data on the sender host; stage 1 is the sender host DMA; the next stage is an abstract pipeline stage of the network DMAs at both end hosts and a receiver host DMA; stage 3 is the copy on the receiver host. The parameters of this pipeline are given in Table I [24]. The second column is the per-fragment overhead, the third column is the per-kilobyte transmission time at the reference supply voltage, the last column is the (normalized) reference power for each stage at the reference supply voltage. We further suppose there is a packet of fixed size being transmitted via this network with various user-specified latency constraints and let the threshold and reference supply voltages be 0.8 and 5 V, respectively.

We first determine the energy dominant stage. As we discussed in Section IV, energy consumption on each stage is determined by the supply voltage which is proportional to $T_j(5)/(C-g_j)$, where C is a stage-independent constant. (This is clear from (10) and the expression of $D_j(k)$ in the

TABLE II
OPTIMAL FIXED SIZE FRAGMENTATION, VOLTAGE SCHEME AND THE NORMALIZED POWER CONSUMPTION FOR MYRINET GAM PIPELINE WITH CONSTANT VOLTAGE ON EACH STAGE

Latency		stage 0		stage 1		stage 2		stage 3	
(μs)	k	voltage(v)	power(P_0)	voltage(v)	power(P_1)	voltage(v)	power(P_2)	voltage(v)	power(P_3)
200	6	2.49	8.02e-02	4.96	0.97	5.52	1.40	2.63	9.97e-02
250	7	2.11	4.13e-02	3.97	0.45	4.32	0.61	2.22	5.04e-02
300	8	1.91	2.64e-02	3.43	0.27	3.68	0.35	1.99	3.19e-02
360	9	1.72	1.67e-02	2.96	0.16	3.13	0.19	1.79	1.99e-02
420	10	1.61	1.21e-02	2.67	0.11	2.81	0.13	1.67	1.43e-02

TABLE III
ENERGY REDUCTION ON MYRINET GAM PIPELINE OVER TRADITIONAL ENERGY MINIMIZATION TECHNIQUES

Latency	stage 0	stage 1	stage 2	stage 3	Total	stage 0	stage 1	stage 2	stage 3	Total
(μs)	saving over best fixed-voltage system w/o shut-down					saving over best fixed-voltage system with shut-down				
200	94.2%	30.1%	N/A	92.8%	54.3%	79.6%	19.2%	N/A	77.2%	44.0%
250	93.2%	25.2%	N/A	91.7%	52.5%	76.1%	15.4%	N/A	73.7%	41.3%
300	92.3%	22.0%	N/A	90.7%	51.3%	73.0%	12.9%	N/A	70.6%	39.1%
360	91.3%	19.0%	N/A	89.7%	50.0%	69.7%	10.7%	N/A	67.3%	36.9%
420	90.5%	16.9%	N/A	88.8%	49.0%	66.8%	9.2%	N/A	64.4%	35.1%
Average	92.3%	22.6%	N/A	90.7%	51.4%	73.0%	13.5%	N/A	70.6%	39.3%
	saving over fixed (5 volt) system w/o shut-down					saving over fixed (5 volt) system with shut-down				
200	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
250	95.8%	54.6%	39.3%	95.0%	71.1%	82.1%	36.8%	25.3%	80.4%	56.2%
300	97.4%	73.9%	66.5%	96.9%	83.7%	85.6%	53.7%	46.8%	84.4%	67.6%
360	98.3%	84.4%	80.7%	98.0%	90.4%	88.1%	65.0%	60.8%	87.2%	75.3%
420	98.8 %	89.6%	87.5%	98.6%	93.6%	89.7%	71.8%	68.9%	88.9%	79.8%
Average	97.6%	75.6%	68.5%	97.1%	84.7%	86.4%	56.8%	50.5%	85.2%	69.7%

proof of Theorem 4.3.) Therefore, the larger the per-byte transmission time $T_j(5)$ is, the more energy is consumed. So is the per-fragment overhead g_j. In the Myrinet GAM pipeline, it is clear that stage 2 is the dominant stage because it has both the largest per-fragment overhead and the longest per-byte transmission time.

After identifying the energy dominant stage, we can apply Theorem 4.3 to decide the optimal packet fragmentation directly from (7) which is reported in the second column of Table II. Then we can compute the constant on the right-hand side of (6) and calculate the supply voltage level for each stage from (1) and (6). Finally, the power consumption can be obtained from (2). We normalize it to the power consumption at the 5-V reference supply voltage and details are shown in Table II.

To demonstrate the energy efficiency of the new approach, we compare the above result with the traditional energy minimization techniques, namely minimal supply voltage selection and system shut-down. We report our power/energy saving over these techniques in Table III.

The minimal supply voltage selection method computes the minimal voltage that can meet the transmission deadline and applies it to the (fixed-voltage) processors on all stages. In this case, such optimal voltage is the one that we use for stage 2. Columns 2–5 in the top half of Table III give the energy saving of the new approach over the best (voltage-) configured fixed-voltage system on each individual stage. An average of 92.3%, 22.6%, and 90.7% power/energy reduction on the three pipeline stages, excluding the dominant stage 2, respectively, is achieved. At both end hosts (stages 0 and 3), significant amounts of power/energy are saved because of the high transmission speed at these two stages (see Table I). Stage 1 has the same per-byte transmission time as stage 2, however, it has a smaller per-fragment overhead g_1, so we can lower the supply voltage (as shown in Table II) and this little difference in the per-fragment overhead results in a more than 22% power/energy saving. There is no saving from stage 2 because this approach uses the same voltage on stage 2 as our approach.

If we use systems with a fixed 5–V voltage, the energy dominant stage 2 becomes the bottleneck as it has the largest per-fragment overhead and the slowest transmission speed. For a tight 200 μs latency, it fails to meet the transmission deadline. The use of variable voltage processor solves this problem since we can speed up the bottleneck stage by applying a higher voltage as indicated in Table II. Columns 2–5 in the bottom half of Table III show the power/energy saving for loose latency constraints. The average saving is almost 85% and we save nearly 70% from the energy dominant stage.

The system shut-down technique shuts the system (or some components of the system) down when the system is idle to save energy. We compare our approach with an ideal system shut-down technique that shuts the system down whenever there is no processing load and turns the system back on whenever necessary and there is no overhead associated with system shut-down and wake-up. Because energy consumption is the product of power and execution time, it becomes necessary to distinguish power and energy consumption when the system shut-down technique is applied. Basically, reducing supply voltage saves power and energy consumption, but not at the same rate since low voltage results in long execution time to complete the same amount of workload. In our simulation, we assume that the system shuts down to save energy when idle, either waiting for packet from the previous stage or waiting for the acknowledge from the next stage. In this case, our approach saves more than 85% energy on both end hosts and 56% and

TABLE IV
PIPELINE PARAMETERS AND ENERGY REDUCTION ON ADPCM, MC UNIT AND RLS PIPELINES (THE ENERGY DOMINANT STAGES ARE IN BOLD)

	parameters			optimal voltages($volt$)					average energy saving	
stage j	$g_j(\mu s)$	$T_j(\mu s/KB)$	P_j	test_1	test_2	test_3	test_4	test_5	w/o shut-down	w shut-down
ADPCM										
0	1	10	2.0	1.95	1.79	1.56	1.47	1.31	95.14%	76.33%
1	2	5	1.0	2.62	2.35	1.98	1.83	1.57	85.39%	62.31%
2	**3**	**23**	**4.6**	4.85	4.14	3.22	2.84	2.26	N/A	N/A
Total									36.27%	28.29%
MC Unit										
0	1	10	5	1.25	1.20	1.15	1.11	1.08	99.52%	89.94%
1	1	1	1	0.92	0.91	0.90	0.89	0.88	99.97%	93.79%
2	2	200	200	4.87	4.24	3.61	3.20	2.91	1.13%	0.64%
3	1	10	5	1.25	1.20	1.15	1.11	1.08	99.52%	89.94%
4	**2**	**201**	**101**	4.89	4.25	3.62	3.21	2.92	N/A	N/A
Total									4.23%	3.59%
RLS										
0	1	28	1.0	1.09	1.07	1.03	1.00	0.93	99.79%	90.39%
1	2	25	4.0	1.07	1.06	1.02	0.99	0.92	99.81%	90.59%
2	**5**	**1228**	**156.8**	4.91	4.60	3.73	3.25	2.04	N/A	N/A
Total									3.08%	2.80%

50%, respectively, on stages 1 and 2. This gives a total energy saving of almost 70% over the fixed 5-V system combined with the system shut-down technique. When both minimal supply voltage selection and system shut-down are applied, our approach achieves an average 39.3% energy saving. Detailed energy saving on each stage is reported in the right half of Table III.

Comparing the four blocks in Table III, one can see that both minimal supply voltage selection and system shut-down techniques can save system's power/energy consumption. Their combination, the best fixed-voltage system with shut-down in the top right block, is capable of reducing more than half of the energy consumed by the fixed 5-V system without shut-down. Our approach can save 39.3% more on top of this. Furthermore, energy saving is mainly determined by g_j and T_j; loose latency results in less energy saving over the minimal supply voltage selection method and more energy saving over the fixed 5-V system.

We have constructed several other communication pipelines using the Hyper tool and Table IV shows the pipeline parameters and our simulation results. The three systems, ADPCM, MC Unit, and RLS, have three, five and three pipeline stages, respectively. The energy dominant stages are marked in bold. The g_j and T_j columns are the same as before. P_j column is the relative power consumption. We simulate the transmission under different latency constraints and the optimal voltages for each pipeline stage are reported. The last two columns show the energy saving on each stage over constant supply voltage without and with the system shut-down technique. On most stages, we see significant (close to or more than 90%) energy saving. The last row of each pipeline system gives the total energy saving over all the pipeline stages when the relative power consumption P_j is considered. For ADPCM, we are able to save 28% even when system shut-down technique is applied. However, for the other two systems, the energy savings are less than 5%. The reason is that in these systems the energy dominant stages consume large portion, e.g., almost 97% on stage 2 in RLS, of the system's total energy. Therefore, our technique is especially efficient for pipeline systems where the nondominant stages also contribute significant to the total energy consumption.

VII. CONCLUSION

In this paper, we address the problem of how to minimize the power consumption in system-level pipelines under latency constraints. In particular, we exploit advantages provided by variable voltage design methodology to optimally select speed and therefore voltage of each pipeline stage. We define the problem and solve it optimally under realistic and widely accepted assumptions. We apply the obtained theoretical results to develop algorithms for power minimization of computer and communication systems. We direct our discussion in detail in two specific cases: 1) the packet has to be equally fragmented and the supply voltage on a stage cannot be changed; and 2) both the size of the fragment and the voltage are variables. We derive an explicit formula for the first case and transform the latter to the problem of finding the minimum of a nonlinear function. The simulation with real-life pipeline parameters shows that even with the former approach, significant power reduction is possible without additional latency.

ACKNOWLEDGMENT

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

REFERENCES

[1] T. E. Anderson, M. D. Dahlin, J. M. Neefe, and D. A. Patterson *et al.*, "Serverless network file systems," *ACM Trans. Comput. Syst.*, vol. 14, no. 1, pp. 41–79, Feb. 1996.

[2] T. D. Burd and R. W. Brodersen, "Processor design for portable systems," *J. VLSI Signal Processing*, vol. 13, no. 2–3, pp. 203–221, Aug. 1996.

[3] T. D. Burd, T. Pering, A. Stratakos, and R. Brodersen, "A dynamic voltage-scaled microprocessor system," in *IEEE Int. Solid-State Circuits Conf.*, vol. 466, Feb. 2000, pp. 294–295.

[4] T. D. Burd and R. W. Brodersen, "Design issues for dynamic issues scaling," in *Int. Symp. Low Power Electron. Design*, July 2000, pp. 9–14.

[5] A. P. Chandrakasan, S. Sheng, and R. W. Broderson, "Low-power CMOS digital design," *IEEE J. Solid-State Circuits*, vol. 27, no. 4, pp. 473–484, 1992.
[6] J. M. Chang and M. Pedram, "Energy minimization using multiple supply voltages," in *Int. Symp. Low Power Electron. Design*, 1996, pp. 157–162.
[7] B. N. Chun, A. M. Mainwaring, and D. E. Culler, "Virtual network transport protocols for Myrinet," *IEEE Micro*, vol. 18, no. 1, pp. 53–63, Jan. 1998.
[8] R. Gonzalez and M. Horowitz, "Energy dissipation in general purpose microprocessors," *IEEE J. Solid-State Circuits*, vol. 31, pp. 1277–1284, Sept. 1996.
[9] K. Govil, E. Chan, and H. Wasserman, "Comparing algorithms for dynamic speed-setting of a low-power CPU," in *ACM Int. Conf. Mobile Comput. Networking*, Nov. 1995, pp. 13–25.
[10] I. Hong, D. Kirovski, G. Qu, M. Potkonjak, and M. B. Srivastava, "Power optimization of variable-voltage core-based systems," *IEEE Trans. Comput.-Aided Design*, vol. 18, pp. 1702–1714, Dec. 1999.
[11] M. Horowitz, "Low power processor design using self-clocking," in Workshop on Low-Power Electronics, Aug. 1993.
[12] H. A. Jamrozik, M. J. Feeley, G. M. Voelker, and J. Evans *et al.*, "Reducing network latency using subpages in a global memory environment," in *Proc. Int. Conf. Architectural Support Programming Languages Operating Systems*, vol. 31, Sept. 1996, pp. 258–267.
[13] V. Von Kaenel, P. Macken, and M. G. R. Degrauwe, "A voltage reduction technique for battery-operated systems," *IEEE J. Solid-State Circuits*, vol. 25, pp. 1136–1140, Oct. 1990.
[14] Y. R. Lin, C. T. Hwang, and A. C. Wu, "Scheduling techniques for variable voltage low power designs," *ACM Trans. Design Automat. Electron. Syst.*, vol. 2, no. 2, pp. 81–97, 1997.
[15] P. Macken, M. Degrauwe, M. Van Paemel, and H. Oguey, "A voltage reduction technique for digital systems," in *Proc. IEEE Int. Solid-State Circuits Conf.*, Feb. 1990, pp. 238–239.
[16] D. Mosberger and L. L. Peterson, "Making paths explicit in the Scout operating system," in *Proc. USENIX Symp.n Operating Systems Design Implementation*, Oct. 1996, pp. 28–31.
[17] W. Namgoong, M. Yu, and T. Meng, "A high-efficiency variable-voltage CMOS dynamic dc-dc switching regulator," in *Proc. IEEE Int. Solid-State Circuits Conf.*, vol. 489, Feb. 1997, pp. 380–381.
[18] G. Qu, "Scheduling Problems for Reduced Energy on Variable Voltage Systems," Master Thesis, Comput. Sci. Dept., Univ. California, Los Angeles, 1998.
[19] G. Qu and M. Potkonjak, "Techniques for Energy-Efficient Communication Pipeline Design," Univ. Maryland Inst. Advanced Computer Studies (UMIACS), Tech. Rep. UMIACS-TR-2002-16, 2002.
[20] S. Raje and M. Sarrafzadeh, "Variable voltage scheduling," in *Int. Symp. Low Power Design*, 1995, pp. 9–14.
[21] A. J. Stratakos, S. R. Sanders, and R. W. Brodersen, "A low-voltage CMOS dc-dc converter for a portable battery-operated system," in *Proc. Power Electronics Specialist Conf.*, vol. 1, June 1994, pp. 619–626.
[22] K. Usami and M. Igarashi, "Low-power design methodology and applications utilizing dual supply voltages," in *Proc. Asia South Pacific Design Automation Conf.*, Jan. 2000, pp. 123–128.
[23] G. M. Voelker, H. A. Jamrozik, M. K. Vernon, and H. M. Levy *et al.*, "Managing server load in global memory systems," in *Proc. ACM Int. Conf. Measurement Modeling Computer Systems (SIGMETRICS 97)* , vol. 25, June 1997, pp. 127–138.
[24] R. Y. Wang, A. Krishnamurthy, R. P. Martin, T. E. Anderson, and D. E. Culler, "Modeling communication pipeline latency," in *Proc. Joint Int. Conf. Measurement Modeling Computer Systems (SIGMETRICS '98/PERFORMANCE'98)*, 1998, pp. 22–32.
[25] M. Weiser, B. Welch, A. Demers, and S. Shenker, "Scheduling for reduced CPU energy," in *Proc. USENIX Symp. Operating Systems Design Implementation*, Nov. 1994, pp. 13–23.
[26] G.-Y. Wei and M. Horowitz, "A low power switching power supply for self-clocked systems," in *Proc. Int. Symp. Low Power Electronics Design*, August 1996, pp. 313–317.
[27] M. Welsh, A. Basu, and T. von Eicken, "ATM and fast Ethernet network interfaces for user-level communication," in *Proc. 3rd Int. Symp. High-Performance Computer Architecture*, 1997, pp. 332–342.
[28] A. Wolfe, "Issues for low-power CAD tools: A system-level design study," *ACM Trans. Design Automat. Embedded Syst.*, vol. 1, no. 4, pp. 315–332, October 1996.
[29] F. Yao, A. Demers, and S. Shenker, "A scheduling model for reduced CPU energy," *Proc. IEEE Annual Foundations Computer Science*, pp. 374–382, Oct. 1995.

Gang Qu received the B.S. and M.S. degrees in mathematics from the University of Science and Technology of China in 1992 and 1994 and the M.S. and Ph.D. degrees in computer science from the University of California, Los Angeles (UCLA), in 1998 and 2000.

Since 2000, he has been with the University of Maryland, College Park, where he is currently an Assistant Professor in the Department of Electrical and Computer Engineering and Institute of Advanced Computer Studies. His research interests include intellectual property reuse and protection, low-power system design, applied cryptography, and sensor networks.

Dr. Qu won the Outstanding Master of Science Award in 1998 from the Henry Samueli Engineering School, UCLA, the 36th Design Automation Conference Graduate Scholarship Award in 1999, and the ACM SIGMOBILE MobiCom Best Student Paper Award in 2001.

Miodrag Potkonjak received the Ph.D. degree in electrical engineering and computer science from University of California, Berkeley, in 1991.

In 1991, he joined C&C Research Laboratories, NEC USA, Princeton, NJ. Since 1995, he has been with the University of California, Los Angeles (UCLA), where he is a Professor in the Computer Science Department. His research interests include communication systems design, embedded systems, computational security, and practical optimization techniques.

Dr. Potkonjak received the NSF CAREER award, the OKAWA Foundation Aaward, the UCLA TRW SEAS Excellence in Teaching Award, and a number of best paper awards.