# Techniques for Energy Minimization of Communication Pipelines

Gang Qu and Miodrag Potkonjak
Computer Science Department, University of California, Los Angeles, CA 90095-1596
{gangqu,miodrag}@cs.ucla.edu

## Abstract

The performance of many modern computer and communication systems is dictated by latency of communication pipelines. At the same time, power 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 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 and show that significant power reduction is possible without additional latency.

## 1 Introduction

System level pipelines are widely acknowledged as the most likely bottleneck of many computer systems [11, 15]. For example, a read miss in the system data or instruction cache blocks the application program until the entire block with requested data arrives [1, 17]. The trade-off is clear: longer blocks imply fewer misses, but also longer interrupt latency. Similarly, in high speed local and wide-area networks selecting properly block size to exploit intrinsic concurrency in communication pipelines is a key issue [2, 4, 20]. As the final example where communication pipelines dictate performances we mention path-oriented operating systems [12].

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 recent work by Wang et al [18].

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 paper is organized as follows, we review the related work in communication pipeline and low power design techniques, then we define the problem in section 3. We solve the problem optimally in two cases: (i) each pipeline stage has a fixed voltage which varies from stage to stage; (ii) every stage can have variable supply voltages. We present the experimental results in section 6 and then conclude.

## 2 Related Work

The most relevant related work are efforts in communication pipeline design and evaluation, and low power design techniques. 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 area networks are directly relevant. Within the latter, we focus our survey on system-level power minimization techniques and variable voltage techniques.

In the introduction section, 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 the design of Internet for quite a long time. More recently, studies of how to exploit flexible block fragmentation to improve performances of DEC workstations has also been conducted [8]. More detailed survey of fragmentation techniques is given in [18].

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 [9, 10]. Later, [7] described implementation of several digital power supply controllers based on this idea. Nielsen et al [14] extended the dynamic voltage adaptation idea to take into account data dependent computation times in self-timed circuits. Recently several researchers developed efficient DC-DC converters that allow the output voltage to be rapidly changed under external control [13]. Researchers at MIT [3, 6] have applied the idea of voltage adaptation based on data dependent computation time from [14] to synchronously clocked circuits.

In the software world, also there has been recent research on scheduling strategies for adjusting CPU speed so as to reduce power consumption. The existing work is in the context of non-real-time workstation-like environment. [19] 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. [5] concluded that smoothing helps more than prediction in voltage changing. Finally, [21] described an off-line minimum-energy schedule and an average rate heuristic for job scheduling for independent processes with deadlines.

## 3 Background and Problem Formulation

We describe the variable voltage processor and the store-and-forward pipelining network, characterize the user packet to be transmitted, and then we state the problem.

### 3.1 Variable Voltage Processor

In most part of this paper, we use the *ideal variable voltage processor* [16] where the supply voltage can be changed from $0$ to $\infty$

instantaneously without any overhead. Although this ideal processor is not feasible because of the inneglectable amount of time for the voltage to reach steady state at the new voltage and the feedback control behavior of the DC-DC switching regulator, the study of this model gives us insight view of the problem and more important, it provides the lower bound of energy consumption by using variable voltage processors.

With different supply voltages, the processor is able to operate at different speeds and therefore the time and power used to accomplish the same task will also be different.

### 3.2 Network Model

As proposed in [18], we represent the network as a sequence of store-and-forward pipeline stages characterized by the following parameters:

- $n$: the number of pipeline stages.
- $g_i$: the fixed per-fragment overhead for stage $i$.
- $T_i(5)$: the per-byte transmission time for stage $i$ with a reference supply voltage of 5 volts $(0 \leq i \leq n-1)$.

$g_i$, the fixed per-fragment overhead can be considered as the context switch time. It may vary from stage to stage. If none of the stages has overhead, as we will show soon, the best strategy is to fragment the packet as small as possible.

$T_i(5)$ is proportional to the inverse of the bandwidth for stage $i$ with $5v$ 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 send as a single fragment.

### 3.3 Problem Formulation

Our objective is to minimize the energy consumption for transmitting a packet through the network under the user-specified latency constraint. Following variables are associated with the packet for the convenience of analysis:

- $B$ : the size of the entire packet.
- $T$ : the deadline to transmit the entire packet.
- $k$ : the number of fragments.
- $x_i$ : the size (in byte) of the $i$th fragment $(0 \leq i \leq k-1)$.
- $t_{i,j}$ : the time that the $i$th fragment stays in the $j$th stage.

The packet's size $B$ and the deadline $T$ are given by the user, the network is characterized by $n, g_i, T_i(5)$, and we assume that the processors at all stages are identical.

Let $v_j(t)$ be the voltage at which the $j$th processor operates at time $t$, then

$$E_j = \int_0^T P(v_j(t))\, dt \tag{1}$$

is the energy consumed by this processor, 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. The problem is formulized as:

**Problem:** Energy Minimization with Deadline on Variable Voltage Processor(EMDVVP).
**Instance:** A network with parameters $n, g_i$ and $T_i(5)$, a packet with size $B$ and deadline $T$.
**Question:** Find the voltage scheme $v_j(t)$ for each processor and a fragment $\{x_0, x_1, \dots\}$ 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.

Figure 1: Problem formulation.

## 4 Fixed Voltage within the Same Stage

We first consider the simple case when the processor at each stage operates at a fixed voltage which can be arbitrary. The voltage scheme problem then becomes to finding a constant $v_j$ for the processor at the $j$th stage, and the energy consumed by this processor, from (1), is simplified to $E_j = P(v_j)T$. Moreover, the time that the $i$th fragment stays in the $j$th stage can be expressed as:

$$t_{i,j} = g_j + T_j(v_j)x_i \tag{2}$$

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

The intuition behind Lemma 4.1 is that the network will use as much time as possible to schedule the processors with low voltages and thus minimize energy consumption. On the other hand, for each single stage, the best strategy is to transmit a fragment immediately upon its reception or at the accomplishment of sending the previous fragment whichever comes later. This observation leads to the next lemma.

**Lemma 4.2**
Given that the packet can only be fragmented into fixed size and the supply voltage for each processor cannot be changed, if a voltage scheme $\{v_0, v_1, \dots, v_{n-1}\}$ minimizes the energy consumption, then

$$t_{i,j} = constant \tag{3}$$

From (2), the processor at the stage that has the largest per-fragment overhead has to operate at a high voltage to achieve a small per-byte transmission time $T_j(v_j)$ due to (3). 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:

$$k = \sqrt{\frac{T}{g_d}(n-1)} - (n-1) \tag{4}$$

and the constant on the r.h.s. of (3) is $\frac{T}{n+k-1}$.

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

- $T$: the deadline. When the latency constraint is loose (i.e., $T$ goes large), we can have more fragments from (4). Energy consumption is reduced since every processor gets a longer transmission time.
- $n$: the number of stages in the network. If we differentiate (4) with respect to $n$, the result is positive which means that the more stages in the network, the more fragments we should have. This takes advantage of the parallelism.
- $g_d$: the per-fragment overhead at the (energy) dominant stage. If this overhead goes large, less fragments should be used to cut the total overhead. And if there is no overhead, then we should fragment the packet as small as possible so that more part of the packet can be transmitted parallelly.
- $B$: the size of the entire packet. The number of fragments in the optimal scheme is independent of the packet size. 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.

## 5 Variable Voltages within the Same Stage

The discussion in the previous section is very restricted, the fragments have equal length and each processor runs at a fixed supply voltage, though different processors may run at different voltages. Now we assume each fragment can have variable size and each processor can run at different level of voltage.

First of all, Lemma 4.1 still holds, which says that we should finish the transmission on its deadline, not any other early time. Another basic fact is from the convexity of energy as a function of the supply voltage:

**Lemma 5.1**
In every stage, to minimize the energy, supply voltages change either on the arrival of a new fragment or at the accomplishment of sending the current fragment.

Recall that $t_{i,j}$ is the time that the $i$th fragment stays in the $j$th stage, which includes both the overhead $g_j$ and the actual transmission time. Lemma 4.2 synchronizes all processors on fixed length fragments such that no stage will congest or starve. This can be generalized to the case when the fragments have different sizes.

**Lemma 5.2**
In the optimal voltage and fragmentation schemes, for all $0 \leq i \leq k-2$ and $1 \leq j \leq n-1$, the following holds:

$$t_{i,j} = t_{i+1,j-1} \tag{5}$$

Combining all these, we propose an approach to the optimal scheme:

1. Let $\{x_0, x_1, \ldots, x_{k-1}\}$ be a fragmentation, with $x_{k-1}$ given by $B - \sum_{i=0}^{k-2} x_i$.
2. Let $\{v_{0,0}, v_{1,0}, \ldots, v_{k-1,0}\}$ be the voltage scheme for the processor at stage 0.
3. Let $\{v_{k-1,0}, v_{k-1,1}, \ldots, v_{k-1,n-1}\}$ be the voltages at each stage to transmit the last fragment of the packet, where $v_{k-1,n-1}$ is solved from the latency constraint $\sum_{i=0}^{k-1} t_{i,0} + \sum_{j=1}^{n-1} t_{k-1,j} = T$.
4. For each stage $j(1 \leq j \leq n-1)$, calculate its voltage scheme $\{v_{i,j} : 0 \leq i \leq k-2\}$ from (2),(5).
5. Total energy consumption: $E = \sum_{j=0}^{n-1} \sum_{i=0}^{k-1} P(v_{i,j}) t_{i,j}$.
6. Solve all the variables in steps 1 2, and 3 from the system

$$\begin{cases} \frac{\partial E}{\partial x_i} = 0, & \text{for } 0 \leq i \leq k-2 \\ \frac{\partial E}{\partial v_{i,0}} = 0, & \text{for } 0 \leq i \leq k-1 \\ \frac{\partial E}{\partial v_{k-1,j}} = 0, & \text{for } 1 \leq j \leq n-2. \end{cases}$$

Figure 2: An approach to the optimal scheme.

As formulated in Section 3.3, a solution to the EMDVVP problem means a supply voltage function for each processor and a packet fragmentation.

Lemma 5.1 outlines the shape of the voltage functions, which are step functions with all possible break points at the time when new fragment comes or current one leaves. Therefore we only need to determine the supply voltage $v_{i,j}$ for each processor to transmit each fragment, which reduces the problem from finding $n$ functions to determining $nk$ numbers, where $k$ is the number of fragments.

Lemma 5.2 predicts a recursive relation among the time that fragments stay at each stage, from which $(n-1)(k-1)$ $v_{i,j}$'s can be easily calculated. Lemma 4.1 tells us the energy is minimized only when the entire transmission finishes at the deadline, so one more variable can be eliminated. The remaining variable $v_{i,j}$'s are those listed in steps 2 and 3 in Figure 2.

Since now we allow variable supply voltages within the same stage, we can shut down the processor (or run it at the minimum voltage if shut down is not allowed) to save energy and this gives the expression of total energy consumed in step 5. Finally we apply the first order condition to solve for the optimal scheme.

Considering that the size of each fragment $x_i$ has also to be determined, we have:

**Theorem 5.3**
Given the number of fragments, the EMDVVP problem with variable-sized fragment and variable voltage at each stage is reduced to solving a nonlinear system (step 6 in Figure 2) of $n + 2k - 3$ free variables.

By repeating Theorem 5.3 for all possible values of $k$, we can solve the EMDVVP optimally. However, the difficulty is that even $n+2k-3$ variables are too many for us to handle and the nonlinear system is also hard to be solved analytically. (See the technical report for a detailed example and discussion.)

## 6 Experimental Results

In this section, we report the results when apply our new energy minimization approach on the Myrinet GAM pipeline[18].

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 1[18]. 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 reference power for each stage at the reference supply voltage.

| stage $j$ | $g_j(\mu s)$ | $T_j(\mu s$/KB) | $P_j$(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 | $P_3$ |

Table 1: Myrinet GAM pipeline parameters.

Further, we suppose there is a 4KB-packet being transmitted via this network with various user-specified latency constraints, and let the threshold and reference supply voltages be 0.8 volts and 5 volts respectively.

As discussed in Section 4, energy consumption on each stage is determined by the supply voltage which is proportional to $\frac{T_j(5)}{C-g_j}$, where $C$ is a stage-independent constant. (This is clearly from the proof of Theorem 4.3 which has been omitted due to space constraint.) Therefore, the larger the per-byte transmission time $T_j(5)$ is, the more energy is consumed. So does the per-fragment overhead $g_j$. In the Myrinet GAM pipeline, it is clear that stage 2 is the dominant stage.

We apply our new variable voltage approach with fixed-size fragmentation to schedule the supply voltage for processors at each stage. The result is shown in Table 2, where the number of fragments is calculated from (4) and the voltage and energy consumption are computed based on this best fixed length fragmentation.

The traditional energy minimization tries to find the minimal supply voltage and then apply it to the processors at all stages to meet the deadline constraint. In this case, this voltage is that in stage 2. Table 3 compares the power consumption at each stage by our new approach vs. the traditional method. At both end hosts (stages 0 and 3), significant amount of energy are saved due to the high transmission speed at these two stages. At stage 1, energy reduction comes from its small overhead $g_1$.

| Latency ($\mu s$) | Number of fragments | stage 0 | | stage 1 | | stage 2 | | stage 3 | |
|---|---|---|---|---|---|---|---|---|---|
| | | 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 2: Optimal voltage scheme for Myrinet GAM pipeline.

| Latency ($\mu s$) | power at stage 0 ($P_0$) | | | power at stage 1 ($P_1$) | | | power at stage 2 ($P_2$) | | | power at stage 3 ($P_3$) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | traditional | new | saving | traditional | new | saving | traditional | new | saving | traditional | new | saving |
| 200 | 1.40 | 8.02e-02 | 94.3% | 1.40 | 0.97 | 30.5% | 1.40 | 1.40 | 0% | 1.40 | 9.98e-02 | 92.9% |
| 250 | 0.61 | 4.13e-02 | 93.2% | 0.61 | 0.45 | 25.2% | 0.61 | 0.61 | 0% | 0.61 | 5.04e-02 | 91.7% |
| 300 | 0.35 | 2.64e-02 | 92.4% | 0.35 | 0.27 | 22.3% | 0.35 | 0.35 | 0% | 0.35 | 3.19e-02 | 90.8% |
| 360 | 0.19 | 1.67e-02 | 91.4% | 0.19 | 0.16 | 19.0% | 0.19 | 0.19 | 0% | 0.19 | 1.99e-02 | 89.7% |
| 420 | 0.13 | 1.21e-02 | 90.6% | 0.13 | 0.11 | 17.1% | 0.13 | 0.13 | 0% | 0.13 | 1.43e-02 | 88.8% |

Table 3: Energy reduction on Myrinet GAM pipeline.

## 7 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 and show that significant power reduction is possible without additional latency.

## References


[1] T.E. Anderson, M.D. Dahlin, J.M. Neefe, D.A. Patterson, and others. *Serverless network file systems.* ACM Transactions on Computer Systems, Feb. 1996, vol.14, (no.1):41-79.

[2] N.J. Boden, D. Cohen, R.E. Felderman, A.E. Kulawik, and others. *Myrinet: a gigabit-per-second local area network.* IEEE Micro, Feb. 1995, vol.15, (no.1):29-36.

[3] A. Chandrakasan, V. Gutnik, T. Xanthopoulos. *Data driven signal processing: an approach for energy efficient computing,.*International Symposium on Low Power Electronics and Design, pp. 374-352, Aug. 1996.

[4] B.N. Chun, A.M. Mainwaring, D.E. Culler. *Virtual network transport protocols for Myrinet.* IEEE Micro, Jan.-Feb. 1998, vol.18, (no.1):53-63.

[5] K. Govil, E. Chan, and H. Wasserman. *Comparing algorithms for dynamic speed-setting of a low-power CPU.* ACM International Conference on Mobile Computing and Networking (MOBICOM'95), pp. 13-25, Nov. 1995.

[6] V. Gutnik, and A. Chandrakasan. *An efficient controller for variable supply-voltage low power processing.* 1996 Symposium on VLSI Circuits, pp. 158-159, June 1996.

[7] M. Horowitz. *Low power processor design using self-clocking.* Workshop on Low-power Electronics, Aug. 1993.

[8] H.A. Jamrozik, M.J. Feeley, G.M. Voelker, J. Evans, and others. *Reducing network latency using subpages in a global memory environment.* International Conference on Architectural Support for Programming Languages and Operating Systems, Cambridge, MA, USA, 1-5 Oct. 1996). SIGPLAN Notices, Sept. 1996, vol.31, (no.9):258-67.

[9] V. Von Kaenel, P. Macken, M. G. R. Degrauwe. *A voltage reduction technique for battery-operated systems.* IEEE Journal of Solid-State Circuits, Vol. 25, No. 5, pp. 1136-1140, Oct. 1990.

[10] P. Macken, M. Degrauwe, M. Van Paemel, H. Oguey. *A voltage reduction technique for digital systems.* 1990 IEEE International Solid-State Circuits Conference (ISSCC) Digest of Technical Papers, pp. 238-239, Feb. 1990.

[11] R.P. Martin, A.M. Vahdat, D.E. Culler, T.E. Anderson. *Effects of communication latency, overhead, and bandwidth in a cluster rchitecture.* (24th Annual International Symposium on Computer Architecture. ISCA '97). Computer Architecture News, May 1997, vol.25, (no.2):85-97.

[12] D. Mosberger, L.L. Peterson. *Making paths explicit in the Scout operating system.* (Second USENIX Symposium on Operating Systems Design and Implementation (OSDI), Seattle, WA, USA, 28-31 Oct. 1996).

[13] W. Namgoong, M. Yu, T. Meng. *A high-efficiency variable-voltage CMOS dynamic dc-dc switching regulator.* 1997 IEEE International Solid-State Circuits Conference (ISSCC) Digest of Technical Papers, pp. 380-381, Feb. 1997.

[14] L. S. Nielsen, C. Niessen, J. Sparso, K. van Berkel. *Low-power operation using self-timed circuits and adaptive scaling of the supply voltage.* IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 2, No. 4, pp. 391-397, Dec. 1994.

[15] L.L. Peterson & B.S. Davie. *Computer networks : a systems approach* San Francisco, Calif. : Morgan Kaufmann Publishers, 1996.

[16] G. Qu. *Scheduling Problems for Reduced Energy on Variable Voltage Systems* Master Thesis, Computer Science Dept., Univ. of California, Los Angeles, 1998.

[17] G.M. Voelker, H.A. Jamrozik, M.K. Vernon, H.M. Levy, and others. *Managing server load in global memory systems.* (1997 ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS 97), Seattle, WA, USA, 15-18 June 1997). Performance Evaluation Review, June 1997, vol.25, (no.1):127-38.

[18] R.Y. Wang, A. Krishnamurthy, R.P. Martin, T.E. Anderson, D.E. Culler. *Towards a Theory of Optimal Communication Pipelines* to appear in SIGMETRICS, 1998

[19] M. Weiser, B. Welch, A. Demers, S. Shenker. *Scheduling for reduced CPU energy* USENIX Symposium on Operating Systems Design and Implementation (OSDI), pp. 13-23, Nov. 1994.

[20] M. Welsh, A. Basu, T. von Eicken. *ATM and fast Ethernet network interfaces for user-level communication.* Third International Symposium on High-Performance Computer Architecture 1997. p. 332-42.

[21] F. Yao, A. Demers, S. Shenker. *A scheduling model for reduced CPU energy.* IEEE Annual Foundations of Computer Science, pp. 374-382, Oct. 1995.