Energy-Efficient Multiprocessor-based Router Linecards

Abstract – In support of continuously increasing line rates and various Internet services, multiprocessor-based linecards have appeared in next-generation routers, significantly improving performance. However, this improvement has come at the cost of increased energy consumption. We present a simple yet effective DVS-based scheme for energy-efficient operations of multiprocessor-based linecards. We prove that for a given task and a timing constraint, those processors in a linecard consume less energy when operating at the same voltage than operating at different voltages. Additionally, we derive the optimal configuration for minimal energy consumption in multiprocessor-based linecards, and show that it is extensible to general-purpose multiprocessor systems under certain constraints.

Index terms – Dynamic voltage scaling, energy, linecards, multiprocessors, optimization, routers.

1. Introduction

Recently introduced routers have multiprocessor-based LC's (MBLs) to support port speeds up to 40 *Gbps* [1]. These LCs naturally consume greater power than their uniprocessor-based counterparts (with each MBL possibly consuming 500*W* of power). The total power consumption grows as the number of LCs increases, with a single router chassis possibly consuming some 15*KW* of power. Such high power consumption leads to increased operational and cooling costs, and also reduced reliability of the router. Work dealing with energy-efficient multiprocessor systems has been considered. In particular, scheduling algorithms for variable-length tasks have been proposed [2] to deal with scheduling tasks on a fixed number of processors, but they fail to detail the complete operating configuration of a DVS-based multiprocessor system, namely, the appropriate number of operational processors, and their desirable frequency and voltage settings.

Operating a uniprocessor at a single voltage throughout task execution to meet the deadline precisely, results in lowest energy consumption [3]. In this article, we prove for the first time that all active processors in a multiprocessor system should also maintain one single voltage to minimize power. However, unlike uniprocessor systems, where this single voltage is unique for minimum energy consumption, a multiprocessor system may have multiple such voltage levels that all complete the task exactly at the deadline. Energy optimization in a multiprocessor aims to find the optimal multiprocessor configuration, comprising N_{opt} processors operating at frequency f_{opt} (and voltage V_{opt}) to minimize energy consumption. This paper deals with such optimization for a multiprocessor incorporated in the LCs of a router.

To maximize energy efficiency, an aggressive voltage scheduling scheme hinges on accurate prediction of the input traffic load to an LC for the upcoming time period. Our DVS scheme for MBLs makes use of filter-based Internet traffic predictors. Using real Internet traces, we show the mean accuracy of these predictors to be always greater than 98%. Our scheme achieves power minimization by dynamically determining the appropriate number of active processors and adjusting the processor voltages (and speeds) to the adequate performance level needed according to the predicted load, without sacrificing performance. Due to low link utilizations of routers (typically averaging ~15% [4]), significant energy savings can be attained.

2. Background

This section first describes functionality of a multiprocessor incorporated in an LC, then followed by a brief of key DVS concepts.

2.1 Multiprocessor-based router linecard

Recent next-generation routers use MBLs (shown in Fig. 5), comprising a set of processors referred to as a processor array (PA). Using fields in individual packet headers, the PA performs the following functions: (1) a longest prefix match operation on the lookup tables, (2) packet classification, (3) metering operations as per billing rules, (4) collects statistics, which are used for implementing policies for rate guarantees, flow control, etc., (5) receives route updates and modifies the lookup table accordingly, and (6) load balancing operations for cell transfers over the switching fabric. Clearly, most LC functionality is carried out by the PA, making it the chief energy consumer.

2.2 Dynamic voltage scaling (DVS)

DVS refers to the ability for a CMOS circuit to operate at dynamically varying voltage levels, with the goal of adapting the circuit to deliver the appropriate performance level required by a task while minimizing energy consumption [5-8]. In addition to device limitations (such as threshold voltage), the minimum voltage at which a CMOS device may be operated is restricted by the task delay permitted (i.e., its deadline). To maximize its utilization, a processor should operate at the maximum possible frequency for a given voltage (thus each voltage level has a corresponding unique frequency setting). This frequency is simply the inverse of the processor's critical-path delay. A processor requires two components for energy-efficient DVS operations: (1) a prediction mechanism, needed to predict the minimum frequency (and voltage) required to complete a task exactly at its deadline, and (2) a voltage regulator to dynamically vary the operating voltage based on the output of the prediction mechanism. Of course, high prediction accuracy leads to lower energy consumption, while meeting the timing constraint.

In addition to controlling voltage levels for a given clock frequency, the voltage regulator must also have the ability to change the operating voltage when a new clock frequency is requested [6] (as per the predictor output). Two parameters of such a regulator need to be considered: (a) transition time, which is the time taken by the regulator to change the voltage from one level to the other, and (b) transition energy, which is the energy consumed while changing the voltage. Typically, the transition time may be in the order of tens of μs [6], precluding very fast response switching times. Additionally, the energy consumption per transition is generally a few μj.

3. Prediction Mechanism

An LC housing a single OC48 port, with a low link utilization of only 15% may receive over a million packets of the minimum size (40 bytes) per second, yielding packet arrival and processing times being less than 1 μs. Hence, due to the transition time and transition energy of a DVS system, it is not feasible to vary the processor voltage at the granularity of single packet arrival times. Thus, predictions are made suitably for a period of time, called the prediction interval (PI), which may be in the

order of *msecs – secs*, where the predictors estimate the number of packets that will arrive at the LC during the next PI. PI is a crucial parameter for predictors of real-time tasks (such as packet processing). If PI is too small, the predictor may not be able to gather sufficient information to accurately predict the packet arrival rate for the next PI. Furthermore, very small PIs result in greater transition energy consumption per second. On the other hand, large PIs give coarser predictions, and could lead to inadequate performance responses under bursty traffic conditions.

3.1 Task model and traffic traces

The LC performance measure of interest is its throughput delivered, i.e., the number of packets the LC can process per second. Packet processing at the LC by the PA (as stated earlier) involves operations which (based on their frequency of occurrence) may be computation-intensive (like packet header processing) or computation-light (like table updating). During a given PI, an appropriate computation level is dictated by the number of packets arrived during PI and the mix of operations involved for those arrived packets. Since the number of packets arrived tend to vary widely, the appropriate computation level calls for an effective load predictor. A typical predictor often requires at least a certain number of packets to yield good prediction accuracy (as explained earlier). The best PI values under different packet arrival rates will be investigated next using real Internet traces. Since the next generation MBLs are targeted at high data speeds, we consider only line rates of OC-48 and beyond. Once the best PI is determined, an effective prediction will provide accurate computation level for the next PI.

We use Internet traces available from the National Laboratory for Applied Network Research (NLANR) [9] for testing. The traces selected have been collected over: (1) 2.4 *Gbps* links between Cleveland and Indianapolis, (2) 2.4 *Gbps* links between Kansas City and Indianapolis, and (3) 10 *Gbps* links between Indianapolis and Kansas City. We have randomly selected four traces of 5 *mins* each, collected at different times, over both the 2.4 *Gbps* and the 10 *Gbps* links. We have selected 4 additional *pairs* of 10 *Gbps* traces, where a trace pair comprises packets traveling in opposite directions on the same link, at the same time. Each trace pair is combined and the entries are time-sorted to synthesize a 20 *Gbps* trace. Fig. 1 shows variations in the number of packet arrivals. The 2.4 *Gbps* traces are labeled tf_j, the 10 *Gbps* traces are labeled te_j, and the 20 *Gbps* traces are labeled tw_j, where $1 \leq j \leq 4$. Note that due to variations in link utilizations, traffic over slower links may have larger number of packet arrivals than traffic at faster links. The mean link utilization of these traces is < 40% (not illustrated in Fig. 1), which gives the opportunity for significant energy savings.

3.2 Packet predictors

Packet processing operations are ideally suited for parallelization, with each packet processed independently by a processor. The number of actual packets arriving at an LC at any given time is highly variable. Unlike complex offline methods that capture parameters from trace data to model the packet arrival rate (λ), our simple online predictors estimate λ using observed values of λ during past PIs. In addition to being very accurate (> 98% mean accuracy), the predictors possess low complexity (thus consuming little power).

An efficient DVS operation relies on accurate prediction about the total number of processing cycles required during the next PI. It in essence translates to predict λ. To this end, three different low complexity predictors based on filters described in [10] are employed. The input to these predictors is simply the observed values of λ over the past PI(s), while the output is the predicted λ for the next PI. An important property of these filter-based predictors is their reactivity, which defines how a predictor responds to dynamic changes in traffic loads. A predictor can be stable (low reactivity), whereby it tends to reject very noisy observations. Alternatively, it can be agile (high reactivity), which refers to its ability to detect and react to rapid changes in input traffic. Stable predictors often result from observing the values of λ for many past PIs. On the other hand, agile predictors give more weight to the last observed value of λ.

a) Value Predictor

The value predictor (VP) is the simplest of our three predictors, where the predicted value for the next PI is simply the observed value for the last PI. i.e.,

$$P_v(t) = O_v(t-1) \quad (1)$$

where $P_v(t)$ is the predicted λ value for PI(t), and $O_v(t\text{-}1)$ is the observed λ value during PI($t-1$). The prediction error (er_v) for PI(t) is expressed as a fraction of the observed value as:

$$er_v(t) = \frac{|O_v(t) - P_v(t)|}{O_v(t)} = \frac{|O_v(t) - O_v(t-1)|}{O_v(t)}. \quad (2)$$

Since the VP uses only the last observed value of λ to make its prediction, it has maximum agility. Low mean er_v values are attained when the duration of the PI (denoted by PI_d) is long enough to have accumulated sufficient information about the number of arriving packets to make an accurate prediction for the next PI.

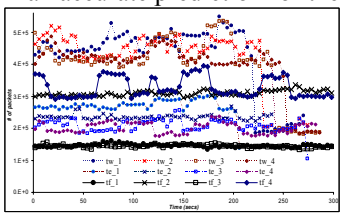

Fig. 1. Internet traces used.

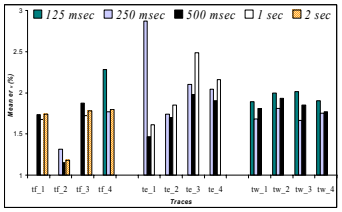

Fig. 2. Mean er_v (%).

Fig. 2 shows mean er_v for each trace. It can be observed that the value of PI_d with lowest mean er_v (denoted as PI_d^l) for each trace is dependent on the packet arrival rate. Traces with smaller λ values require a larger PI_d^l to achieve the lowest mean er_v (see Fig. 1 and Fig. 2). Since λ values of the 10 *Gbps* traces are similar, minimum er_v is found at $PI_d^l = 500\ msecs$ for all these traces. For the 20 *Gbps* traces $PI_d^l = 250\ msecs$. Since mean λ for 10 *Gbps* traces is half the mean λ for the 20 *Gbps* traces, PI_d^l for the 10 *Gbps* traces is expected to be double that of the 20 *Gbps* traces. Similarly, two 2.4 *Gbps* traces tf_1 and tf_3 have $PI_d^l = 1\ sec$, which is about half the PI_d value for the 10 *Gbps* traces. Also, due to higher λ of traces tf_2 and tf_4, their PI_d^l values are 500 *msecs* and 250 *msecs*, respectively. Mean er_v for all traces is seen to be < 2%, resulting in a mean accuracy > 98%. Note that for clarity we have shown only three er_v values for a trace. However, we have tested for PI_ds ranging from 10 *msecs* to 10 *secs*, and we observe that er_v steadily increases as the PI_d setting increases (or decreases) above (or below) the best value for all traces. Thus, we see that although the PI_d should be

long enough so that sufficient information is available to make an accurate prediction, it should be short enough to enable the predictor to track finer variations in λ.

b) Moving Average Predictor

The second predictor is the Moving Average Predictor (MAP) whose estimate of λ is the average of the observed values of λ during the last W PIs. Specifically,

$$P_m(t) = \frac{\sum_{i=1}^{W} O_m(t-i)}{W} \quad (3)$$

where $P_m(t)$ is the predicted value of λ for PI(t), and $O_m(t-i)$ is the observed value of λ during PI($t-i$). A crucial parameter of this predictor type is the window size W, which controls its reactivity. Clearly, as W grows, MAP's stability increases at the cost of its agility. Note that an MAP with $W = 1$ is nothing but a VP; thus, VP is an MAP with maximum agility. The prediction error for PI(t) is:

$$er_m(t) = \frac{|O_m(t) - P_m(t)|}{O_m(t)} = \frac{|w \cdot O_m(t) - \sum_{i=1}^{W} O_m(t-i)|}{w \cdot O_m(t)}. \quad (4)$$

Fig. 3 shows changes in mean er_m for different values of W, where the PI_d value is set to PI_d^l for each trace (from results shown in Fig.2). For all traces, $W = 2$ gives us the lowest mean error, which is < 2%. We have tested for values up to $W = 10$ (not shown in Fig. 3) and observe that $er_m(t)$ steadily grows with increasing W. This indicates that better accuracy results from agile predictors. A VP has slightly better accuracy than an MAP, where the difference in the accuracies is < 0.2%. Also, the difference in mean er_m under $W = 2$ and $W = 3$ (and $W = 5$) is less for the 2.4 *Gbps* traces than for the 10 *Gbps* (or 20 *Gbps*) ones. This is because the 2.4 *Gbps* traces have relatively more stable traffic loads than the other traces.

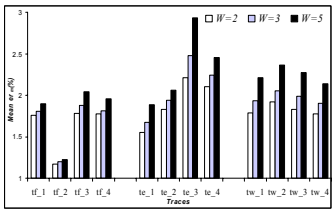

Fig. 3. Mean er_m (%).

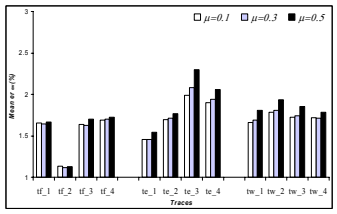

Fig. 4. Mean er_{ew} (%).

c) Exponentially-Weighted Moving Average Predictor

In an Exponentially-Weighted Moving Average Predictor (EWMAP) the estimate generated is a linear combination of the last observed value and the previous estimate, and it is given by:

$$P_{ew}(t) = \mu \cdot P_{ew}(t-1) + (1-\mu) \cdot O_{ew}(t-1) \quad (5)$$

where $P_{ew}(t)$ is the predicted value of λ for PI(t), $O_{ew}(t-1)$ is the observed value of λ in PI($t-1$), $P_{ew}(t-1)$ is the previous estimate, which is clearly a measure of observed values of λ during earlier PIs, and μ ($0 \le \mu \le 1$) gives the reactivity of the predictor. The error for PI(t) is:

$$er_{ew}(t) = \frac{|O_{ew}(t) - P_{ew}(t)|}{O_{ew}(t)}. \quad (6)$$

It can be shown that Eq. (5) may be rewritten as:

$$P_{ew}(t) = \mu^{t-1} \cdot P_{ew}(1) + (1-\mu) \cdot \sum_{i=2}^{t} \mu^{t-i} \cdot O_{ew}(i-1). \quad (7)$$

If $\mu < 1$, the first term of Eq. (7) can be ignored for large values of t. Also, for large t and small values of i (say, $i < t - r$), $\mu^{t-i} \cdot O_{ew}(i-1)$ is very small and can be ignored. Hence, Eq. (7) can be revised as:

$$P_{ew}(t) = (1-\mu) \cdot \sum_{i=t-r}^{t} \mu^{t-i} \cdot O_{ew}(i-1). \quad (8)$$

Thus, $er_{ew}(t)$ in terms of only observed values is:

$$er_{ew}(t) = \frac{|O_{ew}(t) - (1-\mu) \cdot \sum_{i=t-r}^{t} \mu^{t-i} \cdot O_{ew}(i-1)|}{O_{ew}(t)}. \quad (9)$$

Clearly, smaller values of μ makes $O_{ew}(t-1)$ dominate the predicted value, resulting in a more agile predictor (note, $\mu = 0$ reduces an EWMAP to a VP). In Fig. 4, we illustrate variations in mean er_{ew} for different values of μ, where PI_d values are set as per our results of Fig. 2. Since the 10 *Gbps* and 20 *Gbps* traces are relatively unstable, higher accuracies result when an EWMAP is agile, as seen in Fig. 4 where $\mu = 0.1$ gives the lowest error. On the other hand, $\mu = 0.2$ gives better accuracy for the 2.4 *Gbps* traces, due to their relative stability. An exception is the tf_4 trace, which is more unstable than the other 2.4 *Gbps* traces, and hence benefits from a lower μ. An EWMAP gives finer reactivity control than an MAP, where μ can be increased to give better accuracy for any stable trace. In contrast, an MAP exhibits large er_m unless traffic has better stability, reflecting that it is frequently unable to offer reasonably fine reactivity control. The prediction accuracy of EWMAP is observed to be nearly equal to that of a VP, with the difference of their mean errors negligible (< 0.05%). Although slightly costlier than a VP, an EWMAP provides flexibility, wherein (if needed) the predictor's reactivity can be varied based on traffic stability. In addition to the three predictors, we have also tested predictors based on stability and error filters [10]. Those predictors have a higher hardware cost, and their accuracy was found to be no better than the presented ones. For sake of brevity, we do not discuss them further.

4. Energy efficient LC architecture

Fig. 5 illustrates our DVS scheme for an MBL, where a predictor unit and a voltage scheduler are added to a basic MBL. The predictor takes as its input: (1) the number of packets that have arrived in the last PI, and (2) the processor queue length. Note that the units (in Fig 5) which are not essential to energy conservation are not described here.

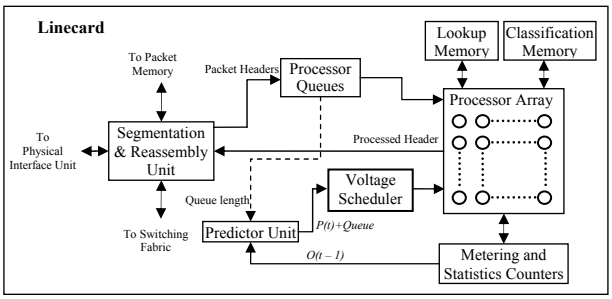

Fig. 5. An energy-efficient multiprocessor-based LC.

The number of packets in processing queues is added to the estimated arrived packets in the next PI to give rise to the predicted computation level required. This prediction drives the voltage scheduler unit, allowing it to

choose the appropriate voltage and frequency settings for the PA during the next PI. The chosen settings are then fed to the PA's voltage regulator circuits (which are not shown in Fig. 5). Our DVS scheme is simple and yet aggressive to adapt continuously to traffic load changes. It accurately predicts λ and tunes the PA to reach minimum energy consumption. This is in contrast to implemented power-saving schemes which drop the voltage level after the system has been idle for a specific period of time, and also allows the voltage level to ramp up in anticipation of the heavier processing capability needed ahead.

4.1 Conditions for energy minimization

It has been shown that if a uniprocessor completes its task before its deadline, the energy consumption is not minimized [3]. This argument naturally holds true for multiprocessors. For a PA to complete task execution exactly at the deadline, 100% prediction accuracy is required. Although this is generally impossible, our predictors are demonstrated to achieve close to 100% accuracy. With nearly perfect prediction accuracy, energy savings is attained almost to the best degree when the PA completes processing the predicted number of packets exactly at the end of the PI.

A uniprocessor which uses a single supply voltage level (v) for the entire duration of task execution and completes the task exactly at the deadline is proved to minimize energy consumption, and there is a unique v for such minimization [3]. We show for the first time that a multiprocessor system (such as an LC's PA) operating at a single voltage level (and thus single clock frequency) for all processors during the whole task execution minimizes energy consumption. However, such a voltage is not unique for the multiprocessors (explained later in this section). Note that due to relatively high transition times typical for DVS voltage regulators, a processor's voltage is set at the start of a PI and remains fixed for the whole PI_d. The following notations are needed for the proof of Theorem 1 below:

- N: total number of processors in the PA.
- N_i: number of operational processors, $N_i < N$.
- V_{min}, V_{max}: manufacturer specified limits between which the processors can be safely (and reliably) operated.
- f_{min}, f_{max}: the maximum frequencies of processors operating at V_{min} and V_{max}, respectively.
- f_i: maximum operating frequency of a processor being operated at voltage v_i, this voltage-frequency pair is denoted as (v_i, f_i).
- V_{th}: threshold voltage, where $V_{th} \leq V_{min}$.
- a, C: processor's activity factor and equivalent capacitance, respectively (which are constant for a PI). Processors completing execution at the same time are assumed to have equal activity factors.

If the prediction interval is t_x seconds long, then we have $f_i \cdot t_x$ processing cycles per PI. The dynamic energy consumed per PI by a single processor is given by $E = a \cdot C \cdot v_i^2 \cdot f_i \cdot t_x$ [5],[8]. Hence, energy consumed per PI by N_i processors operating at (v_i, f_i) (denoted as E henceforth) is:

$$E = a \cdot C \cdot v_i^2 \cdot N_i f_i \cdot t_x \,. \tag{10}$$

Theorem 1

In a multiprocessor system, under a given deadline constraint, the energy consumption is never minimized if processors operate at different supply voltages ($v_1, v_2, \ldots, v_n$) during task execution.

Proof

We first consider only 2 voltage levels v_1 and v_2. Let:

- $V_{min} \leq v_1 < v_2 \leq V_{max}$, and $f_{min} \leq f_1 < f_2 \leq f_{max}$, where f_x is the maximum operating frequency at voltage v_x.
- N_1 processors operate at (v_1, f_1) and N_2 processors operate at (v_2, f_2), where $N_1 + N_2 \leq N$.
- X be the total number of processing cycles needed by the PA in a single PI (and is a constant for this proof).
- (v_i, f_i) be an operating point, where $v_1 \leq v_i \leq v_2$ and $f_1 \leq f_i \leq f_2$, such that

$$X = N_i f_i \cdot t_x = N_1 f_1 \cdot t_x + N_2 f_2 \cdot t_x \tag{11}$$

with $N_i \leq N$. Eq. (11) gives the number of processing cycles needed per PI to satisfy the deadline constraint. For all processors to finish needed processing exactly at the deadline, a faster processor executes a larger fraction of the task. Let $p = N_1 f_1 \cdot t_x$ be the number of cycles provided by N_1 processors in t_x *secs*, then, $N_2 f_2 \cdot t_x = X - p$. We show that energy consumed by the PA is less when all processors operate at a single voltage (v_i, f_i) than when N_1 processors operate at (v_1, f_1) while N_2 processors operate at (v_2, f_2). Eq. (10) can be written as:

$$E = a \cdot C \cdot v_1^2 \cdot N_1 f_1 \cdot t_x + a \cdot C \cdot v_2^2 \cdot N_2 f_2 \cdot t_x \tag{12}$$

which gives rise to $E = a \cdot C \cdot v_1^2 \cdot p + a \cdot C \cdot v_2^2 \cdot (X - p)$. Since a and C are constant, we ignore them and rewrite E as:

$$E = v_1^2 \cdot p + v_2^2 \cdot (X - p) = (v_1^2 - v_2^2) \cdot p + v_2^2 \cdot X \tag{13}$$

which is the equation of a straight line with a negative slope (since $v_1 < v_2$), as illustrated in Fig. 6. Differentiating E with respect to p and equating the result to 0, we get:

$$\frac{dE}{dp} = v_1^2 - v_2^2 = 0 \,. \tag{14}$$

Clearly, there are no stationary points and E is minimized (and maximized) only when $v_1 = v_2$ (which implies $f_1 = f_2$) [11], i.e., when all processors operate at one single voltage (and frequency). From Eq. (13) and Fig. 6, E is minimized when $p = X$ (i.e., $N_1 f_1 \cdot t_x = N_i f_i \cdot t_x$) and processors run at a single frequency to provide exactly X processing cycles in t_x *secs*. Thus, N_i processors operate at a single frequency f_i (and voltage v_i) to minimize E (see Eq. (11)). The minimum energy under the given constraints can then be expressed as:

$$E_{min} = a \cdot C \cdot v_i^2 \cdot N_i f_i \cdot t_x \,. \tag{15}$$

Since the energy consumption is not minimized when processors operate at two different supply voltages (v_1, v_2), it can readily be shown that energy consumption can never be minimized if processors operate at more than two different voltages. ▪

When processors operate at (V_{min}, f_{min}), the number of processing cycles provided by the PA (which is $f_{min} \cdot t_x \cdot N_i$) may be less than what is required, resulting in a missed task deadline. In order to complete the task exactly at the deadline with minimal energy consumption, however, the processors should operate at such a (v_i, f_i) setting that the deadline is met exactly, where $V_{min} \leq v_i \leq V_{max}$ and $f_{min} \leq f_i \leq f_{max}$. Unlike a uniprocessor, a multiprocessor system may have multiple such (v_i, f_i) settings. Next, we demonstrate how to obtain an optimal PA configuration.

4.2 Optimization

A multiprocessor system may choose different number of processors (N_i) for task execution while operating at correspondingly different voltage levels to yield the same computation capability required in a PI. For example, the deadline may be met by operating only a small fraction of the processors at a high voltage level, say (V_{max}, f_{max}), or by increasing the number of processors

involved in execution at a lower voltage level. In general, various combinations of (v_i, f_i, N_i) may be possible to complete the task exactly at the deadline. It is thus vital to find out (v_{opt}, f_{opt}, N_{opt}) which minimizes energy consumption, with $V_{min} \le v_{opt} \le V_{max}$, $f_{min} \le f_{opt} \le f_{max}$, $1 \le N_{opt} \le N$. Since each v_i setting has a unique f_i setting for minimizing energy, the optimal operating point is simply denoted by (f_{opt}, N_{opt}). For a velocity saturated delay model [8], the energy consumption per PI in terms of (f_i,N_i) is:

$$E = \frac{a \cdot C \cdot V_{th}^2 \cdot f_i \cdot N_i \cdot t_x}{(1 - f_i \cdot k)^2} . \qquad (16)$$

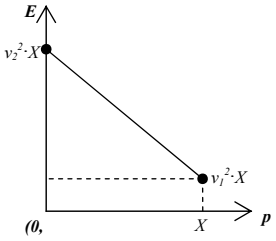

Fig. 6. Graph of E versus p.

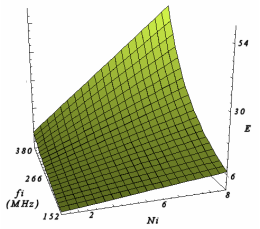

Fig. 7. Energy plane.

The plot of Eq. (16) (called the energy plane) is shown in Fig. 7, where variations in E are plotted for different (f_i, N_i), where $1 \le N_i \le 8$, $t_x = 1$ *sec*, and 152 *MHz* $\le f_i \le$ 380 *MHz*, which is the frequency range of an IBM PowerPC 405 processor [12] (note, constants a, C, V_{th}, k in Eq. (16) have been estimated for this processor).

Theorem 2

(f_{opt}, N_{opt}) can only lie on an edge of the energy plane.

Proof:

Note that f_i and N_i are the only variables in Eq. (16). Taking partial derivates of E with respect to these two variables and equating the results to zero, we get:

$$\frac{\partial E}{\partial N_i} = \frac{a \cdot C \cdot V_{th}^2 \cdot f_i \cdot t_x}{(1 - f_i \cdot k)^2} = 0 , \quad \frac{\partial E}{\partial f_i} = \frac{a \cdot C \cdot V_{th}^2 \cdot N_i \cdot t_x \cdot (1 + f_i \cdot k)}{(1 - f_i \cdot k)^3} = 0 .$$

It is clear that no stationary points exist, and E_{min} is theoretically constrained by the boundary values of (f_i, N_i) [11]. Hence, (f_{opt}, N_{opt}) at which E is minimzed must lie on one of the 4 edges of the energy plane. ▪

We define the PA's load (L_e) as the ratio of the actual number of packets to be processed in a PI to the maximum number of packets that can be processed by the PA in a PI. For processors operating at (f_i, N_i), the task deadline is met *exactly* if and only if the deadline equation of $N_i \cdot f_i \cdot t_x = L_e \cdot N \cdot f_{max} \cdot t_x$ is satisfied, where $L_e \cdot f_{max} \cdot N \cdot t_x$ denotes the computation capability available during the period of t_x. This leads to

$$N_i \cdot f_i = L_e \cdot N \cdot f_{max} . \qquad (17)$$

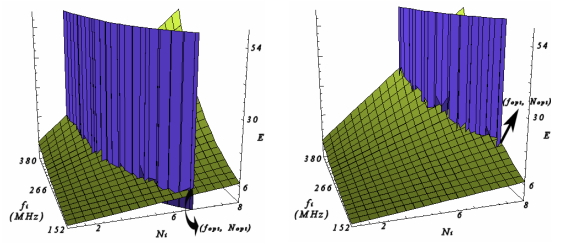

Fig. 8. Deadline restriction due to (a) $L_e = 0.3$, (b) $L_e = 0.6$.

The intersection of the energy plane (of Fig. 7) with the deadline curve of Eq. (17) denotes the energy consumption amounts for all (f_i, N_i) combinations at which the task deadline can be met *exactly*. This intersection is plotted in Fig. 8 for $L_e = 0.3$ (and 0.6). Since E_{min} can lie only along the edges of the energy plane, (f_{opt}, N_{opt}) is found only at a point-of-intersection (POI) of the deadline curve and an edge of the energy plane, as shown in Fig. 8.

The (f_{opt}, N_{opt}) setting with minimum energy consumption can be obtained as follows. For any given input load (L_e), the deadline curve intersects two edges of the energy plane. Let (f_{i1}, N_{i1}) and (f_{i2}, N_{i2}) be these two POIs. For light loads, these points lie along (f_i, *1*) and (f_{min}, N_i), but for medium loads, they are along (f_{max}, N_i) and (f_{min}, N_i). For high loads, they are found along (f_{max}, N_i) and (f_i, N). Under any load condition, the (f_{ix}, N_{ix}) value (where $x = 1$ or $x = 2$) which minimizes energy consumption is the optimal configuration (f_{opt}, N_{opt}).

4.3 Practical considerations

The above discussions have assumed that (f_i, N_i) can be varied continuously. In practical situations, however, this may not be possible. Of course, N_i can take only integer values; furthermore, a processor may have only a few manufacturer-specified discrete operational voltage levels, which results in corresponding discrete frequency settings. Recent power-optimized processors, like Transmeta's Crusoe processor [13], offer very fine-grained voltage regulation, with the voltage changed in steps of 25 *mV*. For most practical applications, this can be considered as continuously varying. Many current processors, however, do not have such a provision for fine-grained voltage regulation. Hence, it may not be always possible to set the PA exactly at the (f_{opt}, N_{opt}) values, calling for the need to find the *practically* optimal values (f_{optP}, N_{optP}), as follows. We find energy consumption at ($f_{i1\pm1}$, $N_{i1\pm1}$) and ($f_{i2\pm1}$, $N_{i2\pm1}$) for both POIs, where $N_{ix\pm1}$ are the two integer values of N_i just above and below the two POIs, and f_{ix+1} (and f_{ix-1}) is the smallest practically settable value of f_i at which the deadline can be met when N_{ix+1} (and N_{ix-1}) processors operate. Of these four (f_i, N_i) settings, the one resulting in the least energy consumption is the *practically* optimal setting (f_{optP}, N_{optP}). It should be noted that unlike the theoretically optimal value, this *practically* optimal setting may not lie along the edges of the energy plane.

4.4 Generalization

From a hardware perspective, our DVS scheme for MBLs is applicable in general to any multiprocessor system to arrive at an optimal energy configuration. As different applications have different task models, certain restrictions may be necessary. Also, the task must be amenable to parallelization with little or no overhead. Digital signal processing applications, such as video processing where frames are processed at a fixed rate are typically well suited for parallelization and our DVS scheme can be easily applied. In general, any application operating under a fixed-throughput mode [9] can optimize its energy consumption using our scheme. If an application requires the PI_d to change dynamically, hardware complexity of the DVS mechanism increases, possibly needing additional system support. Such an application can benefit from our scheme only when its energy savings outweigh the extra energy dissipated by the scheduler and its related units.

5. Performance evaluation

In this section, we illustrate the efficacy of our multiprocessor-based DVS scheme by simulating the

energy consumption of an MBL when traces (described in Section 3) are input over its external ports. Our simulation model makes the following definitions and assumptions:

- N = 8, with each processor having a continuous frequency range of 152 $MHz \leq f_i \leq$ 380 MHz.
- An EWMA predictor with reactivity μ = 0.1, where PI_ds are set as per accuracy results of Section 3.
- $X = \beta \times PC$, is the estimated number of cycles required for the next PI, where β is the number of packets predicted for the next PI, and PC is the number of cycles needed to process a single packet (which is randomly set to 5000 cycles).
- The energy of each voltage transition is set to 4 μj.

Using the methods illustrated in the previous section, (f_{optP}, N_{optP}) is determined based on the *predicted* number of packets for each PI. Additionally, (f_{opt}, N_{opt}) is determined using the *actual* number of packet arrivals instead of the predicted number of arrivals, which gives us the upper bound on the attainable power savings. Since the ideal PI_ds vary for different traces, we use power as our performance metric, instead of energy per PI. Fig. 9 shows mean (f_{opt}, N_{opt}) and (f_{optP}, N_{optP}) for each trace.

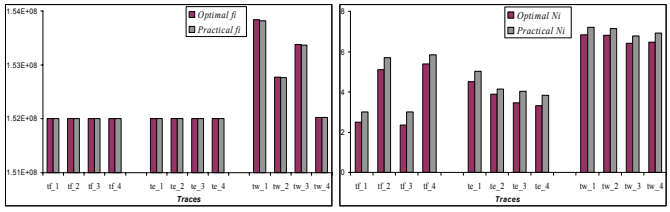

Fig. 9. Optimal and practical mean values of (a) f_i, (b) N_i.

It can be seen that the *practical* values are very close to the *optimal* values for all traces. The small difference between them is due to the predictor inaccuracy (which is < 2%). f_{opt} remains constant (at f_{min}) while N_{opt} varies for all 2.4 *Gbps* and 10 *Gbps* traces. This is because a lower input load for these traces causes (f_{opt}, N_{opt}) to lie along the (f_{min}, N_i) edge of the energy plane, as discussed in the previous section. On the other hand, relatively high input loads of the 20 *Gbps* traces causes (f_{opt}, N_{opt}) to lie on: (1) the (f_{min}, N_i) edge, and (2) the (f_i, N) edge of the energy plane, which is clear in Fig. 9, with $f_{opt} > f_{min}$, and $N_{opt} < N$. Note that the number of processors required by the tf_2 and tf_4 traces is higher than that of any other 2.4 *Gbps* or 10 *Gbps* trace due to their greater λ values (as shown in Fig. 1). The values in Fig. 9 apparently reflect variations in the input load, where a larger λ results in: (1) higher N_i, (2) higher f_i, or (3) both higher N_i and higher f_i.

Due to the lack of prior work dealing with energy savings in MBLs, there are no previous results available for comparison. To appreciate the energy savings due to our scheme, we compare the power consumption with current practice of no DVS implemented in an LC (with all processors operating at f_{max}), with the *practical* and *optimal* power consumptions due to our scheme. We assume that the activity factor a = 1 with DVS employed, and $a = L_e$ without DVS implemented. As seen in Fig. 10, the *practical* power consumption amounts are only slightly larger than their *optimal* counterparts, due largely to high accuracy of the predictor. As expected, the 20 *Gbps* traces consume most power, due to their higher λ values. Similarly, power consumption is more under tf_2 and tf_4 traces, than under other 2.4 *Gbps* and 10 *Gbps* traces. The trace set tested for our LC model achieves average power savings > 60% over an LC without DVS (as illustrated in Fig. 10). However, power savings drops if link utilization (or the incoming traffic rate) increases. Given that a typical router now consumes several KW of power, the proposed scheme has a potential to achieve large energy savings, cutting down the operational costs substantially.

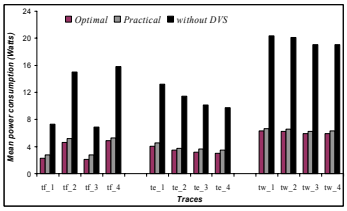

Fig. 10. Power consumption (*Watts*).

6. Conclusion

We have presented a simple yet aggressive DVS scheme for high-performance MBLs to arrive at near optimal energy savings. Such a scheme can be easily implemented with minor changes to router LCs. Our simple predictors exhibit high accuracy, essential for energy efficiency. Processors with one single operating voltage throughout a prediction interval is proved to minimize energy consumption. We also state how to attain a practically optimal PA configuration. Performance evaluation shows that significant energy savings can result from our scheme.

As line rates and router complexity continue to grow, increased hardware logics to accommodate this growth will lead to higher power consumption. There will always be situations under which energy savings can be attained due to the bursty nature of Internet traffic. Reducing operational costs has become imperative for carriers, and it is effectively achievable by our scheme.

References

[1] Cisco Systems Inc., *Next Generation Networks and the Cisco Carrier Routing System*, whitepaper, 2004, URL – http://www.cisco.com.

[2] D. Zhu, R. Melhem, and B. Childers, "Scheduling with Dynamic Voltage/Speed Adjustment Using Slack Reclamation in Multiprocessor Real-Time Systems," *IEEE Trans Parallel and Distributed Systems,* vol. 14, no. 7, July 2003, pp. 686-700.

[3] T. Ishihara and H. Yasuura, "Voltage Scheduling Problem for Dynamically Variable Voltage Processors," *Proc. Intl. Symp. On Low Power Electronics and Design (ISLPED '98)*, Monterey, CA, Aug. 1998, pp. 197-202.

[4] A. Odlyzko, "The Current State and Likely Evolution of the Internet," *Proc. Globecom 1999*, pp. 1869-1875, Dec. 1999.

[5] A. Chandrakasan and R. Brodersen, *Low Power Digital CMOS Design*, Kluwer Academic Publishers, Massachusetts, 1995.

[6] T. Burd *et al.*, "A Dynamic Voltage Scaled Microprocessor System," *IEEE J. Solid-State Circuits*, vol. 35, no. 12, Nov. 2000, pp. 1571-1580.

[7] T. Burd and R. Brodersen, "Design Issues for Dynamic Voltage Scaling," *Proc. Intl. Symp. On Low Power Electronics and Design (ISLPED '00)*, Rapallo, Italy, July 2000, pp. 9-14.

[8] T. Burd and R. Brodersen, "Processor Design for Portable Systems," *J. VLSI Signal Processing Systems*, vol. 13, no. 2-3, Aug. 1996, pp. 203-221.

[9] National Laboratory for Applied Network Research (NLANR), *Special Traces Archive*, URL - http://pma.nlanr.net/Special.

[10] M. Kim and B. Noble, "Mobile Network Estimation," *Proc. ACM Conf. Mobile Computing and Networking,* Rome, Italy, June 2001, pp. 298-309.

[11] B. Gottfried and J. Weisman, *Introduction to Optimization Theory,* Prentice Hall, NJ, 1973.

[12] B. Zhai *et al.*, "Theoretical and Practical Limits of Dynamic Voltage Scaling," *IEEE/ACM Conf. Design Automation Conf. (DAC '04)*, San Diego, CA, June 2004, pp. 868-873.

[13] M. Fleischmann, *LongRun Power Management*, Transmeta Corp., 2001, URL – http://www.transmeta.com.