# Dynamic Voltage and Frequency Scaling in Parallel Network Processors

Raffaele Bolla[1,2], Roberto Bruschi[2], Chiara Lombardo[1,2]

[1] DYNATECH – University of Genoa
Genoa, Italy
{name.surname}@unige.it

[2] CNIT – Research Unit of the University of Genoa
Genoa, Italy
{name.surname}@cnit.it

*Abstract*—**In this paper, we consider energy-aware network devices (e.g. routers, switches, etc.) able to trade their energy consumption for packet forwarding performance by means of DVFS techniques. We focus on state-of-the-art packet processing engines, which generally represent the most energy-starving components of network devices, and which are often composed of a number of parallel pipelines to "divide and conquer" the incoming traffic load. Our goal is to control both the power configuration of pipelines, and the way to distribute traffic flows among them, in order to optimize the trade-off between energy consumption and network performance indexes. With this aim, we propose and analyze a constrained optimization policy, which tries to find the best trade-off between power consumption and packet latency times. In order to deeply understand the impact of such policy, a number of tests have been performed by using real-world traffic traces.**

***Keywords-green networking; low power idle; adaptive rate.***

## I. INTRODUCTION

While the ICT industries were experiencing a huge growth in terms of number of customers and offered services, the $CO_2$ levels provoked by such expansion received little or no interest. Only in the last few years, concern started to raise both because of the environmental risks and the energy costs. Statistics provided by public organizations and internet service providers have attested the growing trend of energy demands and related carbon footprint. Among others, the Global e-Sustainability Initiative (GeSI) [1] estimates an overall network energy requirement of about 21.4 TWh in 2010 for European Telcos, and foresees a figure of 35.8 TWh in 2020 if no Green Network Technologies (GNTs) will be adopted.

Although Internet traffic presents similar trends at the same time and day of the week, it is well known that network links and devices are provisioned for busy or rush hour load, which typically exceeds their average utilization by a wide margin [2]. While this margin is seldom reached, nevertheless the power consumption is determined by it and remains more or less constant even in the presence of fluctuating traffic loads. This situation suggests the possibility of adapting network energy requirements to the actual traffic profiles, dynamically selecting resources according to the present traffic characteristics [3, 4]. In their studies, Tucker et al. [5] and Neilson [6] analyzed the different energy requirements of all HW elements with network-specific functionalities. Their attention focused on high-end router platforms, the devices with the highest complexity level among all network nodes. The data plane energy consumption reaches 54% of the overall requirement, while power and heat management consumes 35% and control plane 11%. Considering these results, it is clear that the data plane is the most energy-starving component, so we decided to focus on packet processing engines for network devices, which generally represent the most energy-harvesting physical component of many network devices, and not only of high-end routers. These engines are realized with heterogeneous HW technologies (from classical ASIC [7] or FPGA [8] chips to GPU-based ones [9]), and often have highly parallel architectures in order to “divide and conquer” the traffic load incoming from a number of high-speed interfaces.

Most of the power management techniques currently studied by the research community already exist in computer processors. Among the most common optimization strategies there are Adaptive Rate (AR) and the Low Power Idle (LPI). The former allows dynamically modulating the capacity of a processing engine (or of a single pipeline), in order to meet traffic loads and service requirements while the latter forces processing engines (or single pipelines) to enter low power states when not sending/processing packets. The impact of such power management capabilities has already been presented in a number of studies [2, 10, 11].

In this paper, similarly to the work in [12], we focus on network processors (i.e., packet processing engines) with parallel architectures, where a number of pipelines divides and conquers the incoming traffic load. We assume that AR and LPI capabilities can be used at each pipeline in order to modulate the energy consumption with respect to the current workload. Differently from, and additionally to [12], this work explicitly considers AR and LPI capabilities realized by means of the Dynamic Voltage and Frequency Scaling (DVFS) technique. This gives us the possibility of evaluating the impact of different design approaches for effectively introducing DVFS mechanisms into next generation network processors.

Our goal is to dynamically manage the network processor configuration in order to optimally balance its energy consumption with respect to its network performance. To this purpose, we modeled the energy- and network-aware dynamics of packet processing engines, and formalized an optimization problem in an enough general way to reflect different criteria, like: (*i*) the minimization of energy consumption for a certain

978-1-61284-0833-6/12/$26.00 ©2012 IEEE

constraint in packet latency time, or (*ii*) the maximization of network performance for a given energy cap, or (*iii*) the optimization of a given trade-off between the two previous policies. The optimization problem takes constraints on maximum energy consumption and packet latencies explicitly into account.

The paper is organized as follows. Section II introduces the techniques that can be used at the HW level to save energy, and their main impact on network performance indexes. Section III describes the reference architecture of the parallel network processor we considered. The model for energy-aware pipelines is described in Section IV, and the optimization problem definition in Section V. Numerical results obtained with real traffic traces are in Section VI, and the conclusions are drawn in VII.

## II. Power Management Capabilities and Network Performance

In order to understand how adaptive green network technologies and relative HW implementations can be effectively designed and applied to next generation network devices, we first have to deeply take into consideration the main features of power management approaches in state-of-the-art HW technologies. Without losing generality, this section introduces the main benefits and drawbacks of power management for the CMOS technologies. Such considerations can be easily extended to other HW technologies (e.g., FPGA).

The power consumption of a CMOS circuit arises from two main contributions [13], namely leakage and dynamic power consumption:

$$\Phi = \Phi_{leakage} + \Phi_{dyn} \qquad (1)$$

The $\Phi_{dyn}$ contribution somehow represents the “ideal” power absorption of the circuit, since it is due to the real transition of CMOS logical states. In more detail, $\Phi_{dyn}$ can be expressed as:

$$\Phi_{dyn} = \nu\, C\, f\, V_{dd}^2 \qquad (2)$$

where $\nu$ is the logical state switching probability, $C$ is the total transistor gate capacitance of the entire module, $V_{dd}$ is the supply voltage, and $f$ is the clock frequency.

It is well known that $\Phi_{leakage}$ is becoming a dominant contribution in today’s silicon, and it results from imperfect cut-off of the transistors and causes power dissipation even without any switching activity. The $\Phi_{leakage}$ usually depends on many factors, like, for instance, the size of CMOS gates, the silicon operating temperature, the supply voltage $V_{dd}$ , etc. State-of-the-art approaches to reduce the dynamic and leakage power include a number of methods, like, among the others, Dynamic Frequency Scaling (DFS), Dynamic Voltage and Frequency Scaling (DVFS), and sleep transistors to shut off power during idle periods of execution [14, 15].

Regarding the DFS methods, acting on $f$ allows linearly scaling $\Phi_{dyn}$ as shown in Eq. 2. However, it is worth noting that this operation also results in a decay of silicon performance and a consequent increase of elaboration times, since the $f$ parameter is roughly proportional to the elaboration capacity (in terms of number of operations that can be performed per second). As far as DVFS techniques are concerned, as evident again in Eq 2, lowering $V_{dd}$ leads to a quadratic reduction in dynamic power. However, a reduction in voltage results in increased delay ($t_d$ – that roughly represents the time period that is required to move from one logical state to the other one) for the circuit [16]:

$$t_d \propto \frac{V_{dd}}{(V_{dd}-V_t)^{\alpha}} \qquad (3)$$

where $V_t$ is the threshold voltage (used to distinguish the “1” and “0” logic levels), and $\alpha$ is the velocity saturation index which usually ranges between one and two. Obviously, the system frequency needs to scale along with the voltage to ensure that the operating frequency does not exceed the switching speed of the circuit. It is easy to demonstrate that $f < 1/t_d$ has to be held for guaranteeing stable HW operations. By observing Eq. 2, it is worth noting that the joint lowering of $f$ and $V_{dd}$ can yield to a cubic trade-off between $\Phi_{dyn}$ and the silicon performance, and further gains in the $\Phi_{leakage}$ contribution.

From a general point of view, DVFS can be designed to act at different levels of granularity, from large modules of the circuit to individual logic blocks [17]. The smaller is the granularity, the more complex is the design and the larger the overhead. For example, the current trend towards multi-processor architectures makes scaling on individual processors an attractive approach. Obviously, in order to support DVFS capabilities, a number of special HW modules, such as Voltage Regulation Modules (VRMs) and Clock Frequency Dividers (CFDs), need to be carefully included into the HW design.

Sleep transistors specifically target $\Phi_{leakage}$: cutting off power from the system during idle periods, sleep transistors can dramatically reduce leakage current [14, 15]. In such a case, the performance decay is mainly due to the time required to re-load the circuit upon wake-up events. The larger is the equivalent capacitance $C$ of the sleeping circuit, the longer will be the delay to recover its fully working conditions. So that, sleeping larger parts of a circuit allows dramatic savings, but at the cost of longer wake-up times.

Owing the techniques above mentioned, we can summarize that HW power management allows saving energy by two main approaches:

- *During active periods*: by reducing the elaboration capacity, and then increasing the elaboration times. $\Phi_{dyn}$ scales in a linear way with respect to the elaboration capacity in case of DFS, and up to cubic relationship in case of DVFS.
- *During idle periods*: by sleeping multiple modules at the cost of introducing delay for waking up the HW and starting the job execution. Idle sleeping approaches usually provide significant power savings, since they target both $\Phi_{dyn}$ and $\Phi_{leakage}$.

Such two approaches are clearly considered by the ACPI (Advanced Configuration and Power Interface) standard [18] for general purpose computing systems, and translated into two different sets of states: the performance (P-) and power (C-) states. When applied to network devices, they can be directly mapped into two main well-known concepts [3, 4], namely *Adaptive Rate* (AR) and *Low Power Idle* (LPI), respectively.

It is worth recalling that LPI and AR techniques have different impacts on packet forwarding performance. AR obviously causes a stretching of packet service times while the sole adoption of LPI introduces an additional delay in packet service, due to the wake-up times [2, 4]. Moreover, preliminary studies in this field [19] showed how performance scaling and idle logic work like traffic shaping mechanisms, by causing opposite effects on the traffic burstiness level. The wake-up times in LPI favor packet grouping, and then an increase in traffic burstiness, while service time expansion in AR favors burst untying, and consequently traffic profile smoothing. Finally, the joint adoption of both energy-aware capabilities may not lead to outstanding energy gains, since performance scaling causes larger packet service times, and consequently shorter idle periods. However, the overall energy saving and the network performance strictly depend on incoming traffic volumes and statistical features (interarrival times, burstiness levels, etc.). For instance, idle logic provides top energy and network performance when the incoming traffic has a high burstiness level. This is because less active-idle transitions (and wake-up times) are needed, and the HW can remain in a low consumption state for longer periods.

## III. The Parallel Network Processor

We focus on packet processing engines for network devices, which generally represent the most energy-harvesting physical component of many network devices, and not only of high-end routers. These engines are realized with heterogeneous HW technologies (from classical ASIC [7] or FPGA [8] chips to GPU-based ones [9]), and often have highly parallel architectures in order "to divide and conquer" the traffic load incoming from a number of high-speed interfaces.

Traffic flows income and outcome from the engine by means of Serializer/Deserializer busses (SerDes), which are realized with different standards like PCI Express, SGMII, XGMII, XAUI, etc. In high performance architectures, as shown in Fig. 1, a specific HW component is required in order to multiplex and de-multiplex traffic between the SerDes and the parallel pipelines of the engine. This component can be included inside the same packet processing engine [7], or it can be placed in the interface cards before the SerDes bus.

In this architecture, the workload distribution can be critical due to the presence of multiple parallel pipelines. Several studies have been made on how to distribute the load among different resources and propose different algorithms [20, 21]. In [21], the authors focus on how to preserve the packet-ordering within individual TCP connections and to achieve both load balancing and efficient system utilization. However, the problem of packet-ordering can be avoided, by considering that modern network processors include dedicated HW for packet reordering. For example, the Netlogic XLP network processor provides the Packet Ordering Engine (POE) [22]. In this respect, we do not consider a specific scheduling or reordering algorithm to be used along with the load distribution procedure among the pipelines. Starting from the main achievements of previous work [2], and in order to make an optimal use of LPI primitives, we decided not to untie the incoming packet batches, and to send every packet composing a batch to a single pipeline. We call this policy SBSP (Same Batch Same Pipeline); this allows reducing the power consumption of the system at the price of a slight increase in packet latency times, especially at low incoming traffic loads.

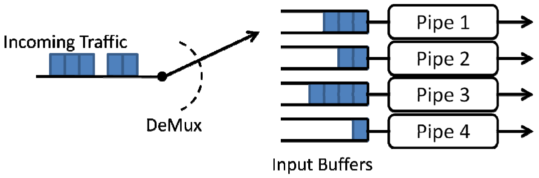

Fig. 1. Scheme of the considered architecture: the traffic incoming from a SerDes bus is de-multiplexed by a load-balancer component towards multiple parallel pipelines in a packet processing engine.

In such scenario, we assume to adopt the AR and the LPI techniques in order to reduce the energy requirements of the packet processing engine. As introduced in Section II, AR is realized by means of DVFS techniques, and then it is performed by modifying the working frequency and the voltage supplied to each pipeline. Let $f^{(i)} \in \{f_0, f_1, \dots, f_X\}$ and $v^{(i)} \in \{v_0, v_1, \dots, v_Y\}$ be the frequency and voltage provided to the *i*-th pipeline of Λ available ones. Given the nature of DVFS (see Eq. 3), the $v^{(i)}$ lowering limits the maximum admissible value $f_{max}^{(i)}$ of the operating frequency. The relationship between such parameters mainly depends on the specific HW implementation. For sake of simplicity, and without losing generality, in the rest of the paper we assume to have $X$=$Y$ available values of frequencies and voltages, and that the following simple relationship between $f_{max}^{(i)}$ and $v^{(i)}$ is maintained:

$$f_{max}^{(i)} = \{f_y \mid v^{(i)} = v_y\} \quad (4)$$

So that, when $v^{(i)} = v_y$, the pipeline can work with the subset $\{f_0, f_1, \dots, f_y\}$ (with $y \leq X$) of available frequencies.

Moreover, we explicitly consider two DVFS design approaches, namely *pipeline common voltage* and *pipeline independent voltage*, respectively. The former consists of a simpler design, where all the pipelines receive the same supply voltage $v_y$. The latter is a more complex HW design, where Λ VRM modules provide independent supply voltages to the network processor's pipelines.

Regarding the LPI, we simply assume to have a single state with a wakeup time of 1 ns, and with an idle energy consumption equal to the 90% of the absorption when active with a certain selection of frequencies and voltages.

## IV. Modeling Energy-Aware Pipelines

In this section the analytical model representing the system behavior is described. This model is derived from the one exposed in [19] but it has been adapted to the current power management capabilities. In detail, the traffic model and the performance indexes, presented in Subsections B and C, do not change according to the new context, while the parameters characterizing the pipeline model in the next section are derived in order to cope with the absence of LPI primitives.

### A. The pipeline model

As previously sketched, the model in [19] has been exploited. We assume to represent each pipeline of the network processor as a single server queuing system with a constant service rate μ. The selection of a specific $\{f^{(i)}, v^{(i)}\} = \{f_x, v_y\}$ couple, satisfying Eq. 4, affects the pipeline performance in terms of both packet service capacity and power consumption. In this respect, since service rate is thought to represent the device capacity in terms of packet headers that can be processed per second, its value is related to the selected frequency and voltage: $\mu(f_x, v_y)$ . It is easy to understand that the same relation is still valid for the active power consumption $\Phi_a(f_x, v_y)$ . Taking into account the idle consumption, $\Phi_i(f_x, v_y)$ represents the energy absorption when the pipeline is idle, while $\tau$=1 ns the wakeup time.

### B. The traffic model

The modeling and the statistical characterization of packet inter-arrival times are well known to have Long Range Dependency (LRD) and multi-fractal statistical features [23]. However, as sustained more recently in [24] and [25], a Batch Markov Arrival Process (BMAP) can effectively estimate the network traffic behavior. Therefore, we decided to model incoming traffic through a Batch Markov Arrival Process (BMAP) with Long Range Dependent (LRD) batch sizes. We assume to receive groups of $j$ packets at exponential inter-arrival times with average value equal to $1/\lambda$. The sizes $j$ of packet batches are supposed to follow the Zipf law, which can be thought as the discrete version of the Pareto probability distribution.

### C. Performance Indexes

The model we propose corresponds to an $M^x/D/1/SET$ ($M^x/D/1$ with server SETup times) queuing system [26]. Packets arrive in batches at Markov inter-arrival times with average rate λ, and are served by a single server at a fixed rate μ. With respect to the model introduced in [19], we can maintain the hypothesis of deterministic server setup times considering the delay $\tau$. Under such assumptions, the average packet waiting time $\widetilde{W}$ can be expressed as follows:

$$\widetilde{W} = \frac{2\tau+\lambda\beta\tau^2-\frac{1}{\lambda}+\frac{1}{\lambda\beta}\sum_{j=1}^{jmax}\beta_j j^2}{2(1+\lambda\beta\tau)} + \frac{\rho^2-\beta+\sum_{j=1}^{jmax}\beta_j j^2}{2\lambda\beta(1-\rho)} \tag{5}$$

and the average power consumption $\widetilde{\Phi}$ as:

$$\widetilde{\Phi} = \frac{\left[\Phi_a\left(\frac{1}{\lambda}\rho+\beta\tau-(1-\rho)\tau_{on}\right)+(1-\rho)\left(\tau_{on}\Phi_t+\frac{1}{\lambda}\Phi_i\right)\right]}{\frac{1}{\lambda}+\beta\tau} \tag{6}$$

## V. THE ENERGY-AWARE LOAD BALANCING

This section is organized as follows. The definition of the optimization problem is in subsection A. Subsection B introduces some preliminary results that can be used to better understand the proposed policy according to different trade-off values and traffic volumes.

### A. Optimization Problem Definition

We consider a traffic de-multiplexer distributing the incoming traffic among Λ parallel pipelines. As introduced in the previous section, we model pipelines as $M^x/D/1/SET$ queuing systems. The $i$-th pipeline works with a $\{f^{(i)}, v^{(i)}\} = \{f_x, v_y\}$ couple, and then with $\mu^{(i)} = \mu(f_x, v_y)$, $\tau^{(i)} = \tau(f_x, v_y)$ , $\Phi_i^{(i)} = \Phi_i(f_x, v_y)$ and $\Phi_a^{(i)} = \Phi_a(f_x, v_y)$ . The traffic incoming to the de-multiplexer is represented as a BMAP process with a batch arrival rate $\hat{\lambda}$, and with Zipf-distributed packet batches with an average length equal to $\hat{\beta}$. Now, we can simply deduce that, thanks to the SBSP policy, the process of incoming traffic is still BMAP, with the following parameters:

$$\beta^{(i)} = \hat{\beta} \tag{7}$$

$$\beta_j^{(i)} = \hat{\beta}_j \tag{8}$$

$$\sum_{i=0}^{\Lambda-1}\lambda^{(i)} = \hat{\lambda} \tag{9}$$

Thus, we can define the average power consumption of our system as the sum of the contributions from the Λ pipelines:

$$\hat{\Phi} = \sum_{i=0}^{\Lambda-1}\widetilde{\Phi}^{(i)}\left(\lambda^{(i)}, f^{(i)}, v^{(i)}\right) \tag{10}$$

and the average latency time experienced by a packet incoming into the system can be defined as in the following:

$$\widehat{W} = \sum_{i=0}^{\Lambda-1}\frac{\lambda^{(i)}}{\hat{\lambda}}\widetilde{W}^{(i)}\left(\lambda^{(i)}, f^{(i)}, v^{(i)}\right) \tag{11}$$

Given the features of incoming traffic load (in terms of $\hat{\lambda}$, $\hat{\beta}$ and $\hat{\beta}_j$) and thresholds on the maximum values of both packet latency $W^*$ and power consumption $\Phi^*$, the objective of the load balancing criterion is to find the best values of $\lambda^{(i)}$, $f^{(i)}$, and $v^{(i)}$ for $\forall i = 0, \dots, \Lambda - 1$ so that the system has the best trade-off between network performance and energy consumption. Thus, we define our optimization problem as follows:

$$\begin{cases} \min\limits_{\lambda^{(i)}, f^{(i)}, v^{(i)},\ i=0,\dots,\Lambda-1} \gamma\frac{\hat{\Phi}}{\Phi^*} + (1-\gamma)\frac{\widehat{W}}{W^*} \\ \widehat{W} < W^* \\ \hat{\Phi} < \Phi^* \\ \sum_{i=0}^{\Lambda-1}\lambda^{(i)} = \hat{\lambda} \end{cases} \tag{12}$$

where the $\gamma$ index ranges between 0 and 1, and represents the "trade-off parameter", which modulates the minimization of power consumption with respect to the one of average packet latency. It is worth noting that, for $\gamma = 0$, our optimization problem corresponds to the maximization of network performance for a given power consumption cap. While for $\gamma = 1$, it corresponds to the minimization of the system power consumption constrained to a maximum value of average latency. Regarding the optimization problem, it is quite complex, since we have a non-linear objective function, which depends on both discrete (i.e. $f^{(i)}, v^{(i)}$ $i = 0, \dots, \Lambda - 1$) and continuous (i.e., $\lambda^{(i)}$ $i = 0, \dots, \Lambda - 1$) variables. By taking into account that the number of pipelines Λ, and of available frequency and voltage values are generally low, our minimization strategy mainly consists on solving the problem for each available configuration of the pipelines. In more detail, for each feasible combination of $\{(f^{(0)}, v^{(0)}), \dots, (f^{(\Lambda-1)}, v^{(\Lambda-1)})\}$, we find the best values of $\{\dot{\lambda}^{(0)}, \dots, \dot{\lambda}^{(\Lambda-1)}\}$ minimizing the objective function and satisfying the constraints. Furthermore, exploiting the last constraints in Eq. 12, we can express $\lambda^{(\Lambda-1)} = \hat{\lambda} - \sum_{i=0}^{\Lambda-2}\lambda^{(i)}$ and consequently reduce the number of variables. Then, we simply try to find the minimum of the objective function by studying its partial derivatives in $\lambda^{(i)}$ $i = 0, \dots, \Lambda - 2$ inside the region satisfying the constraints, and in its frontier.

### B. Analyzing the trade-off

Tests are presented in this subsection with the purpose of better understanding the load balancing problem defined in the previous subsection and, in particular, to give an example on how its solution changes according to the different desired trade-offs, relied to the γ parameter.

The reference architecture used in the shown tests has $\Lambda =$ 4 pipelines, and 8 available level of frequency and supply voltage. For what concerns the incoming traffic, we decided to fix $\hat{\beta} = 4$ while $\hat{\lambda}$ varies between 1 Kpkt/s and 4 Mpkt/s. Finally, thresholds on latency and energy consumption have been fixed to $W^* = 50\ \mu s$ and $\Phi^* = 30$ W. Tests have been carried out in the "pipeline common voltage" case (i.e., $v^{(0)} = v^{(1)} = v^{(2)} = v^{(3)} = v$), and in the "pipeline independent voltage" one. The results are shown in subsections 1) and 2), respectively. Other values used for carrying out the tests are reported in Table I. These experimental results have been obtained from measurements performed on the reference architecture at varying frequency. Further results, not reported in this context, have been used to validate the model introduced in Section IV, with a maximum estimation error lower than 2% for both power consumption and packet latency times.

### 1) *Pipeline common voltage*

Figures 2-6 represent the traffic load shares among the four pipelines for γ = 0, 0.25, 0.5, 0.75, 1. The first case shows the exclusive optimization of latency: since power consumption is not included in the optimization, the best solution consists in equally dividing the load among all pipelines. Considering now Figures 3-5, we can see that, for the lowest values of $\hat{\lambda}$, the load balancer still equally shares the incoming traffic. In fact, although power consumption is part of the minimization for $\gamma \neq 0$, its value is far from $\Phi^*$ when the arrival rate is low. The behavior changes for $\gamma = 1$: without latency involved in the objective function, when the load is below 100 kpkt/s, all traffic is sent only to a single pipeline to save energy.

A more thorough analysis can be provided taking into account the set of voltage and frequency values: Figures 7 and 8 show the frequencies assumed by each pipeline $\{f^{(0)}, f^{(1)}, f^{(2)}, f^{(3)}\}$ and by the voltage $v$ at varying traffic load for $\gamma = 0.25$ and $\gamma = 1$. In the first case, we have an equal share when the incoming load is lower than 500 Kpkt/s. For that value, the control policy excludes two pipelines while increasing the value of the remaining ones and of voltage. For higher loads, the policy is to increase the frequency of as many pipelines as needed by the minimization to satisfy the constraints. In the case of $\gamma = 1$, the strategy consists in using as few pipelines as possible: as we can see in Fig. 8, a single pipeline at a time increases its frequency as traffic grows, while voltage tends to decrease when possible. If we now consider the average power consumption in Fig. 9, we can see that, as

TABLE I –$\Phi_a$ AND $\mu$ ACCORDING TO THE FREQUENCY.

| *f [MHz]* | $\Phi_a$ [W] | $\mu$ [kpkt/s] |
|---|---|---|
| *50* | 1.36 | 136 |
| *66.66* | 1.56 | 137 |
| *100* | 2.02 | 272 |
| *114.27* | 2.18 | 342 |
| *133.33* | 2.52 | 371 |
| *160* | 2.93 | 500 |
| *200* | 3.31 | 616 |
| *266.66* | 4.32 | 823 |
| *400* | 6.03 | 1071 |

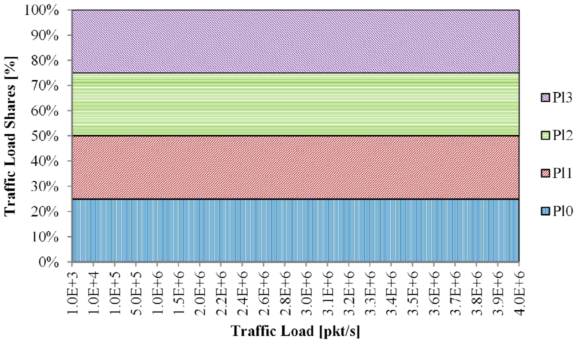

Fig. 2. Optimal load shares for each pipeline (Pl) and for γ = 0 according to increasing traffic volumes.

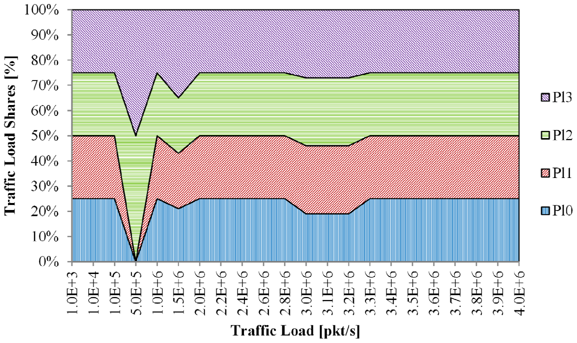

Fig. 3. Optimal load shares for each pipeline (Pl) and for γ = 0.25 according to increasing traffic volumes

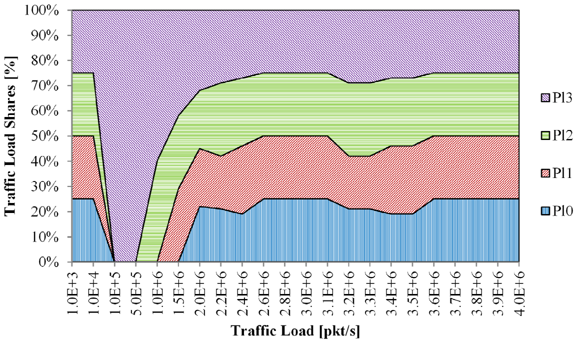

Fig. 4. Optimal load shares for each pipeline (Pl) and for γ = 0.5 according to increasing traffic volumes

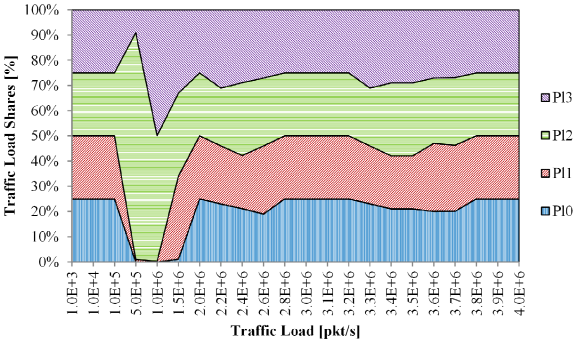

Fig. 5. Optimal load shares for each pipeline (Pl) and for γ = 0.75 according to increasing traffic volumes.

the load grows, consumption stays quite stable for γ=0: in fact, all pipelines are always kept active and at the maximum frequency value, and so is voltage. For the highest traffic loads, in order to respect the latency threshold, traffic has to be evenly distributed among the pipelines for all trade-offs. As a result, all choices of γ bring the same power consumption. Moving on to the average latencies in Fig. 10, the first element to be noticed is that, as expected, results obtained for γ=0.25 and γ=0.5 are closer to those for γ=0, while γ=0.75 and γ=1 are visibly higher than the others. However, as all test scenarios have equally shared traffic for λ=4 Mpkt/s, W finally converges to 22 μs for all trade-offs.

*2) Pipeline independent voltage*

The tests presented in the previous subsection have been repeated in case of a design allowing to individually changing the voltage supplied to each pipeline. This implementation provides lower latencies with respect to the previous tests. This is obviously the result of a more complex and costly HW design. Considering the traffic load shares among the four pipelines, results obtained for γ=0 and γ=1 are not surprisingly the same obtained in the previous tests: in order to keep latency low, it is necessary to give an equal share of traffic to each pipeline, and to decrease consumption as few pipelines as possible have to be fed. Results obtained for γ=0.25, 0.5 and 0.75, instead, have a behavior more similar to γ=1 than in the previous case: if we consider Fig. 11, it is evident how its trend can be compared to the one shown in Fig. 6. This effect is due to the choice of a single possible voltage for each frequency: although this choice did not characterize the previous tests, the load balancer still selected fixed couples of frequency and voltage, especially for the lowest traffic loads. Moreover, in this second case, load shares appear smoother at varying loads: this means that such design brings more coherent policies and less share variations. For what concerns the average power consumption and latency, results shown in Figures 12 and 13 validate the previous assertions: for the loads where we have an increase of energy consumption, latency is lower. Moreover, latency computed for γ<1 is visibly smoother.

## VI. Numerical Results

In order to evaluate the proposed optimization policy in a correct and suitable way, we decided to use daily dynamics of real Internet traffic. In more detail, we used data from the traffic traces that are publicly available in [27]. We used a 96-hour-long traffic trace divided into sequential time windows of 15 minutes. We used the packet processing engine configuration tested in section 5.B.1, and the same values of $\gamma$, $W^*$ and of $\Phi^*$ of the previous section. As far as the incoming traffic is concerned, for each time window, we used the $\lambda$, $\beta$, and $\beta_i$ values as calculated from the traffic trace. In detail, these parameters were obtained by least squares fitting of the Zipf distribution with the trace sample. The evolution of the traffic offered load over the time of the reference traffic trace is reported in Fig. 14 in terms of burst arrival rates and burst sizes. The minimum value of traffic loads is from 3:00 to 6:00, while rush hours occur at 11:00 and 14:00. It is interesting to underline how an increase in incoming traffic volume is due to the rise of both batch arrival rate and burst sizes.

Results in Figures 15 and 16 follow the trend of the traffic trace in Figure 14. However, in absence of explicit LPI capabilities, variations in both Φ and W are less evident than those in the results in [12]. Results obtained for γ=0.25, γ=0.5 and for γ=0.75 are really similar. For what concerns γ=0, these last results make clear that the behavior for this trade-off

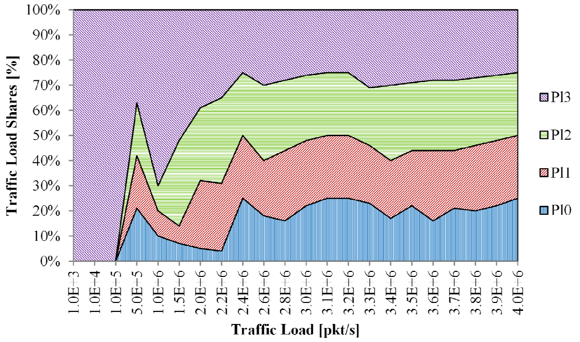

Fig. 6. Optimal load shares for each pipeline (Pl) and for γ = 1 according to increasing traffic volumes

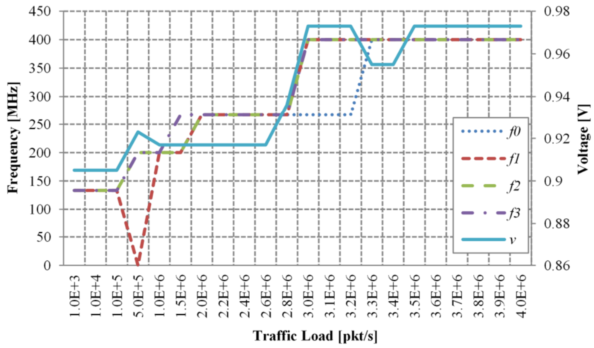

Fig. 7. Frequency of each pipeline and voltage for γ =0.25.

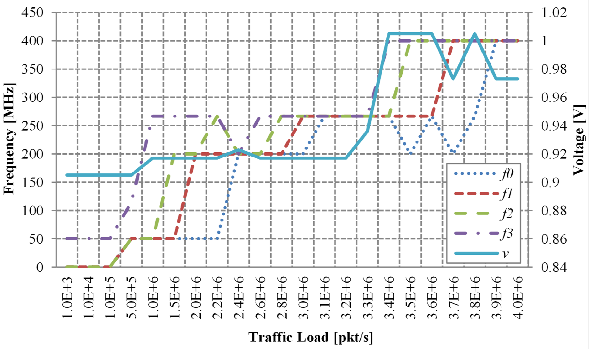

Fig. 8. Frequency of each pipeline and voltage for γ =1.

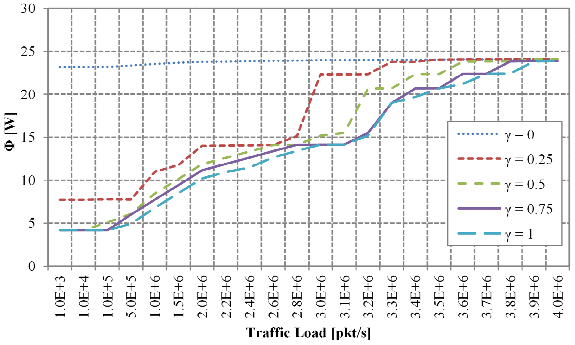

Fig. 9. Average power consumption of the device at varying γ and increasing traffic load.

corresponds to the one of a commercial device, that keeps the same level of performance regardless of the incoming traffic.

## VII. Conclusions

We considered energy-aware processors able to trade their energy consumption for packet forwarding performance by means of both low power idle and adaptive rate schemes. In particular, the AR and LPI capabilities considered in this paper are realized by means of the Dynamic Voltage and Frequency Scaling (DVFS) technique. We focused on state-of-the-art packet processing engines, which are often composed of a number of parallel pipelines to "divide and conquer" the incoming traffic load. Our goal was to control both the power configuration of pipelines, and the best way to distribute traffic flows among them, in order to optimize the trade-off between energy consumption and network performance. We proposed and analyzed a constrained optimization policy, which optimizes the trade-off between power consumption and packet latency times. In order to deeply understand and validate the impact of such policy, tests have been performed on two reference architectures, namely pipeline common voltage and pipeline independent voltage. Then, further tests have been performed by using real-world traffic traces.

## Acknowledgment

This work has been supported by the ECONET (low Energy Consumption NETworks) project co-funded by the European Commission under the 7th Framework Programme (FP7).

## References

[1] Global e-Sustainibility Initiative (GeSI), “SMART 2020: Enabling the Low Carbon Economy in the Information Age”, http://www.theclimategroup.org/assets/resources/publications/Smart2020Report.pdf

[2] S. Nedevschi, L. Popa, G. Iannaccone, D. Wetherall S. Ratnasamy, “Reducing Network Energy Consumption via Sleeping and Rate-Adaptation”, Proc. of the 5th USENIX Symp. on Net. Sys. Des. and Impl., San Francisco, CA, 2008, pp. 323-336.

[3] R. Bolla, R. Bruschi, K. Christensen, F. Cucchietti, F. Davoli, S. Singh, “The Potential Impact of Green Technologies in Next Generation Wireline Networks - Is There Room for Energy Savings Optimization?,” IEEE Comm. Mag. vol. 49, no. 8, pp. 80–86, Aug. 2011.

[4] R. Bolla, R. Bruschi, F. Davoli, F. Cucchietti, “Energy Efficiency in the Future Internet: A Survey of Existing Approaches and Trends in Energy-Aware Fixed Network Infrastructures,” IEEE Comm. Surveys and Tutorials (COMST), vol. 13, no. 2, pp. 223–244, Second Quarter 2011.

[5] R.S. Tucker, R. Parthiban, J. Baliga, K. Hinton, R.W. Aire, W.V. Sorin, “Evolution of WDM Optical IP Networks: A Cost and Energy Perspective,” IEEE J. of Lightw. Tech., vol. 27, no. 3, pp. 243-252, Feb. 2009.

[6] D. T. Neilson, ”Photonics for switching and routing”, IEEE J. of Sel. Top. in Quantum Electr. (JSTQE), vol. 12, no. 4, pp.669-678, Jul. 2006.

[7] The Netlogic XLP processor family, http://www.netlogicmicro.com/Products/MultiCore/XLP.asp.

[8] The NetFPGA project, http://www.netfpga.org/.

[9] S. Han, K. Jang, K.S. Park, S. Moon, “PacketShader: a GPU-accelerated software router,” Proc. of the ACM SIGCOMM Computer Comm. Review, New York, NY, USA, vol. 40, no. 4, pp.195-206, 2010.

[10] J. C. Cardona Restrepo, C. G. Gruber, C. Mas Machuca, “Energy Profile Aware Routing,” Proc. of the IEEE Green Comm. Work. in conjunction with IEEE ICC 09 (GreenComm09), Dresden, Germany, June 2009.

[11] J. Noguera, I.O. Kennedy, “Power Reduction in Network Equipment Through Adaptive Partial Reconfiguration,” Proc. of the 2007 Int. Conf. on Field Program. Logic and Appl. (FPL 2007), Aug. 2007, pp. 240-245.

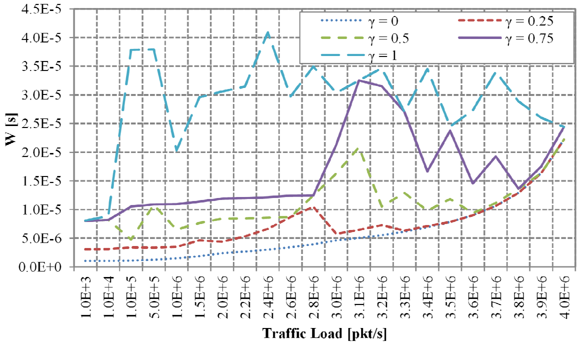

Fig. 10. Average latency of the device at varying γ and increasing traffic load.

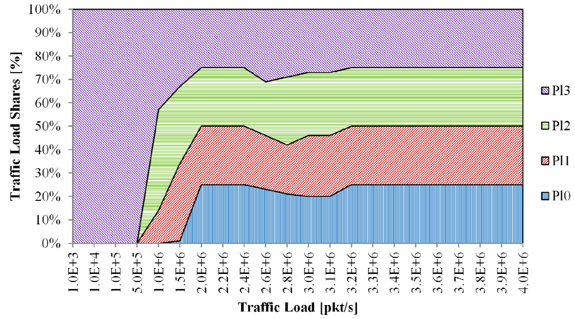

Fig. 11. Optimal load shares in the case of different voltage for each pipeline and for γ = 0.25 according to increasing traffic volumes.

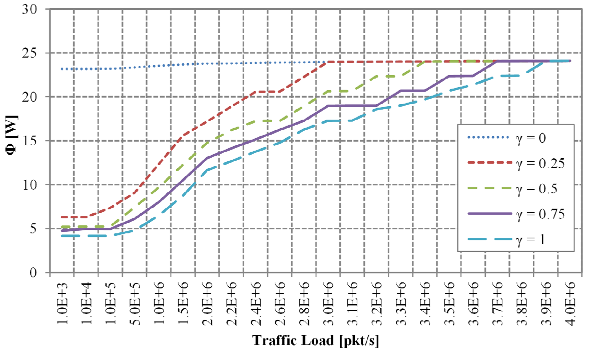

Fig. 12. Average power consumption of the device in the case of different voltage for each pipeline at varying γ and increasing traffic load.

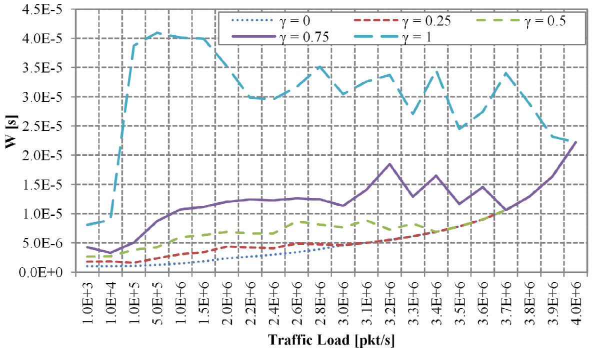

Fig. 13. Average latency of the device in the case of different voltage for each pipeline at varying γ and increasing traffic load.

[12] R. Bolla, R. Bruschi, "'Energy-Aware Load Balancing for Parallel Packet Processing Engines," Proc. of the 1st IEEE Online Conf. on Green Comm. (GreenCom 2011), Sept. 2011, pp. 105-112.

[13] S. Henzler, "Power Management of Digital Circuits in Deep Sub-Micron CMOS Technologies," Springer, Series in Adv. Microel. 2007.

[14] V. Khandelwal, A. Srivastava, "Leakage Control Through Fine-Grained Placement and Sizing of Sleep Transistors," IEEE Trans. on Comp.-Aided Des. of Integr. Circ. and Sys., vol 26, no. 7, pp. 1246-1255, July 2007.

[15] C. De-Shiuan, C. Shih-Hsin, C. Shih-Chieh, "Sleep Transistor Sizing for Leakage Power Minimization Considering Charge Balancing," IEEE Trans. on Very Large Scale Integr. (VLSI) Sys., vol. 17, no. 9, pp. 1330-1334, Sept. 2009.

[16] S. Kang, and Y. Leblebici, "CMOS digital integrated circuits analysis and design,". McGraw-Hill, New York, NY, USA, 2003.

[17] K. Agarwal, K. Nowka, "Dynamic power management by combination of dual static supply voltages," Proc. of the 8th IEEE Int. Symp. on Qual. Elect. Des. (ISQED'07), San José, CA, USA, March 2007, 82-95.

[18] ACPI Specification, http://www.acpi.info/

[19] R. Bolla, R. Bruschi, A. Carrega, F. Davoli, "Green Net. Technologies and the Art of Trading-off," Proc. of the 2011 IEEE Infocom Work. on Green Comm. And Net. (GCN), Shangai, China, Apr. 2011.

[20] W. Shi, M. MacGregor, P. Gburzynski, "Load Balancing for Parallel Forwarding," IEEE/ACM Trans. on Net., vol. 13, no. 4, pp. 790–801, Aug. 2005.

[21] S. Kandula, D. Katabi, S. Sinha, A. Berger, "Dynamic Load Balancing without Packet Reordering," SIGCOMM Comp. Comm. Rev., vol. 37, pp. 51–62, March 2007.

[22] Netlogic. The Netlogic XLP processor family. [Online]. Available: http://www.netlogicmicro.com/Products/MultiCore/XLP.asp

[23] V. Paxson, S. Floyd, "Wide-area Traffic: The Failure of Poisson Modeling," IEEE/ACM Trans. on Net., vol. 3, no. 3, pp. 226-244, 1995.

[24] P. Salvador, A. Pacheco, R. Valadas "Modeling IP Traffic: Joint Characterization of Packet Arrivals and Packet Sizes Using BMAPs," Computer Networks, vol. 44, no. 3, Feb. 2004, pp. 335-352.

[25] A. Klemm, C. Lindemann, M. Lohmann, "Modeling IP Traffic Using the Batch Markovian Arrival Process," Computer Networks, vol. 54, no. 2, Oct. 2003, pp. 149-173, Oct 2003.

[26] G. Choudhury, "An $M^X$/G/1 Queueing System with a Setup Period and a Vacation Period," Queueing Sys., Springer Netherlands, vol. 36, no. 1-3, pp. 23–38, 2000.

[27] MAWI Woring Group Traffic Archive, Sample Point F, available at http://mawi.nezu.wide.ad.jp/mawi/samplepoint-F/20080318/.

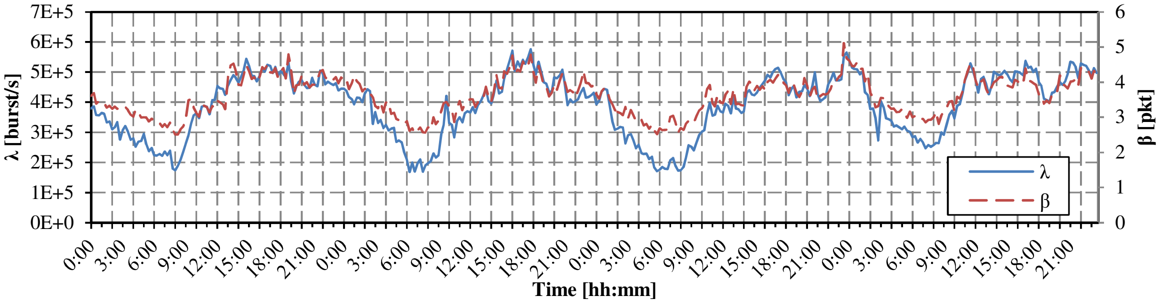

Fig. 14 . Values of $\hat{\lambda}$ and $\hat{\beta}$ as extract from the traffic source in [27].

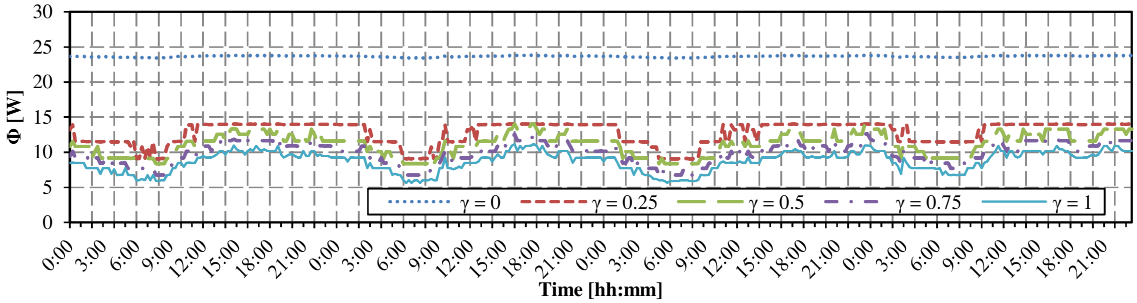

Fig. 15 . Power consumption $\widehat{\Phi}$ for various value of γ with respect to the traffic source in [27].

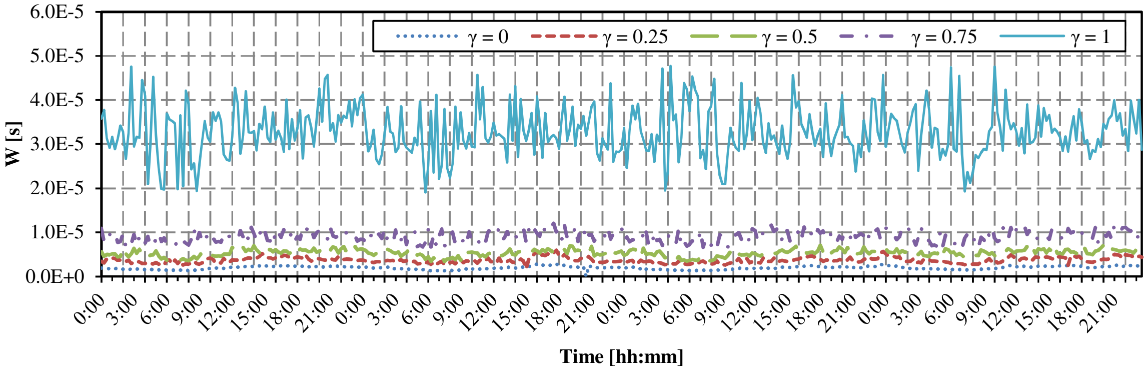

Fig. 16. Average latency times $\widehat{W}$ for various value of $\gamma$ with respect to the traffic trace in [27].