Energy-Aware Load Balancing for Parallel Packet Processing Engines

Raffaele Bolla
Department of Communications, Computer and Systems Science (DIST)
University of Genoa
Genoa, Italy
raffaele.bolla@unige.it

Roberto Bruschi
National Inter-University Consortium for Telecommunications (CNIT)
Genoa, Italy
roberto.bruschi@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 both low power idle and adaptive rate schemes. 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 try 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 experimental data from SW router architectures and real-world traffic traces.**

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

I. INTRODUCTION

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 [1]. 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. Thus the key of any advanced power saving criteria resides in dynamically adapting resources, provided at network, link or equipment levels, to current traffic requirements and loads [2] [3].

In more detail, it is well known that today's network relies very strongly on electronics, despite the great progresses of optics in transmission and switching. Operational power requirements arise from all the HW elements realizing network-specific functionalities, like the ones related to data- and control-planes, as well as from elements devoted to auxiliary functionalities (e.g., air cooling, power supply, etc.). In this respect, the data-plane certainly represents the most energy-starving and critical element in the largest part of network device architectures, since it is generally composed by special purpose HW elements (packet processing engines, network interfaces, etc.) that have to perform per-packet forwarding operations at very high speeds.

In this sense, Tucker *et al.* [4] and Neilson [5] focused on high-end IP routers, and estimated that the data-plane weighs for 54% on the overall device architectures, vs. 11% for the control plane and 35% for power and heat management. The same authors further broke out energy consumption sources at the data-plane on a per-functionality basis. Internal packet processing engines require about 60% of the power consumption at the data-plane of a high-end router, network interfaces weigh for 13%, switching fabric for 18.5% and buffer management for 8.5%.

Starting from these data, 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 [6] or FPGA [7] chips to GPU-based ones [8]), 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 [6], or it can be placed in the interface cards before the SerDes bus (like in the Receive-Side Scaling – RSS – standard for server network

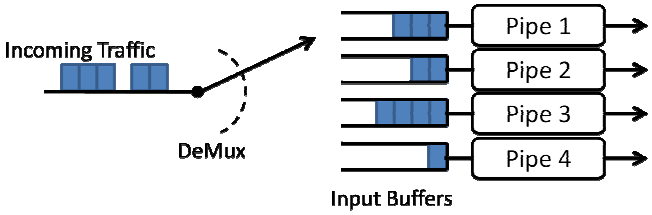

Fig. 2. Scheme of the considered architecture: the traffic incoming from a SerDes bus is de-multiplexed by a load-balancer component towards multiple parallel pipelines in a packet processing engine.

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).

interface cards [9]).

In such scenario, we assume to adopt two basic techniques, already heavily widespread in silicon technologies, in order to reduce the energy requirements of packet processing engine: the *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. As outlined in a number previous works [1] [2] [12], the use of such techniques generally allows trading energy consumption for networking performance (in terms of packet latency times, loss rate, etc.).

Assuming the possibility of selectively tuning AR and LPI mechanisms for each parallel pipeline, our goal is to dynamically manage the engine configuration in order to optimally balance its energy consumption with respect to its network performance. Given the incoming load features and parameters, we want to find *i*) how many pipelines have to actively work, *ii*) their AR and LPI configurations, and *iii*) which share of the incoming traffic volume the load balancer module must assign to them. 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 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 AR and LPI capabilities and how they can impact on network performance. The model for energy-aware pipelines is described in section III, and the optimization problem definition in section IV. Some numerical results obtained with real traffic traces are in section V, and the conclusion in VI.

II. Energy-aware Silicon and Network Performance

Nowadays, the largest part of current network equipment does not include power scaling capabilities, but power management is a key feature in today's processors across all market segments, and it is rapidly evolving also in other hardware (HW) technologies [10]. The rest of this section is structured as follows. Sub-section II.A introduces how ACPI (Advanced Configuration and Power Interface) standards make AR and LPI capabilities accessible to the SW layer. Sub-section II.B discusses the impact of AR and LPI on the forwarding performance of a network device, and how these two capabilities may interact between themselves.

A. The ACPI example

In general purpose computing systems, the ACPI [11] standard models AR and LPI functionalities by introducing two sets of energy-aware states, namely performance and power states (*P*- and *C*-states), respectively.

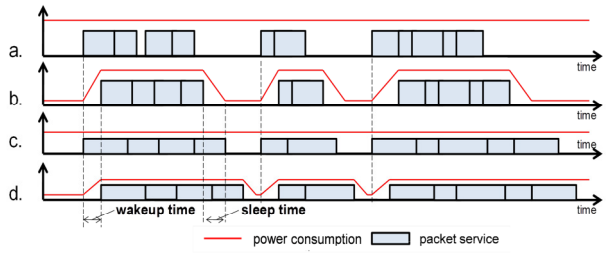

Fig. 2. Packet service times and power consumptions in the cases with: (a) no power-aware optimizations, (b) only LPI, (c) only AR, (d) AR and LPI.

Regarding the *C*-states, C_0 is an active state where the CPU executes instructions, while C_1 through C_n are processor LPI states. As the sleeping power state (C_1, …, C_n) becomes deeper, the transition between active and sleeping (and vice versa) requires longer time.

ACPI also allows the performance of the processor's core to be tuned through *P*-state transitions. *P*-states allow modifying the operating energy point of a core by altering the working frequency and/or voltage, or throttling its clock. Thus, by using *P*-states, a core can consume different amounts of power while providing different processing performance at the C_0 state. At a given *P*-state, the core can transit to higher *C*-states in idle conditions. In general, the higher the index of *P*- and *C*-states is, the less will be the power consumed, and the heat dissipated. Due to issues in silicon electrical stability, the transition time between different *P*-states is generally very slow. A large part of current CPUs can switch their operating *P*-state in about 10 ms. Given such large *P*-state transition times, it is worth noting that any closed-loop control policies with tight time constraints are not feasible and cannot be adopted for optimizing power consumption inside network device architectures.

B. The energy-aware trade-offs

As previously sketched, LPI and AR have different impacts on packet forwarding performance. As shown in Fig 2, AR (Fig. 2c) obviously causes a stretching of packet service times while the sole adoption of LPI (Fig. 2b) introduces an additional delay in packet service, due to the wake-up times. Moreover, preliminary studies in this field [1] 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 favour packet grouping, and then an increase in traffic burstiness, while service time expansion in AR favours burst untying, and consequently traffic profile smoothing. Finally, as outlined in Fig. 2d, 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. It is worth noting that 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

TABLE I –NOTATION DEFINITION.

C_x	selected C-state, $C_x \in \{C_0, C_1, \dots, C_X\}$
P_y	selected P-state, $P_y \in \{P_0, P_1, \dots, P_Y\}$
$\tau_{on}(C_x)$	time needed to wake up the HW from the C_x sleeping state
$\tau_{off}(C_x)$	time needed to put the active HW into the C_x sleeping state
$\tau_{setup}(P_y)$	time to recover forwarding operation after the HW wakeup
$\mu(P_y)$	packet service rate in the P_y state
$\Phi_a(P_y)$	power consumption when the server is active in P_y state
$\Phi_{idle}(C_x)$	power consumption when the server is sleeping in C_x state
$\Phi_t(C_x)$	power consumption during τ_{off} and τ_{off} periods
τ	server vacation time, $\tau = \tau_{on}(C_x) + \tau_{setup}(P_y) + \tau_{off}(C_x)$
λ	rate of batch arrival
β_j	probability that an incoming burst contains *j* packets
β	average number of customer in a batch
P_n	stationary probability of having $n \in [0, N]$ packets in the queuing system
ρ	traffic utilization ρ of the server, which in the case of infinite buffer can be expressed as $\rho = \frac{\lambda\beta}{\mu}$

transitions (and wake-up times) are needed, and the HW can remain in a low consumption state for longer periods.

III. MODELING ENERGY-AWARE PIPELINES

This section is organized as follows. Subsection A introduces the model for pipelines discussing how AR and LPI influence packet processing. The model for the incoming traffic is in subsection B. Finally, subsection C briefly reports some details of the adopted analytical model, and defines the energy- and network-aware performance indexes.

A. The pipeline model

In order to represent the behavior of the pipelines of an energy-aware packet processing engine with LPI and AR capabilities, we decided to adopt the model in [13]. This model is founded on classical concepts of queuing theory, and it is specifically designed to estimate energy- and network-aware performance indexes. For sake of simplicity, let us to adopt the ACPI representation of power management primitives, and refer to AR and LPI configurations in terms of P- and C-states. We assume to model the packet computation engine of the network device as a single server queuing system with maximum service rate μ.

The selection of different P- and C-states is supposed to impact on the pipeline performance in terms of both the packet service capacity, and wakeup times of the server. Similarly to [12] and as previously sketched, the μ service rate is thought to represent the device capacity in terms of packet headers that can be processed per second. Moreover, we assume all packet headers requiring a constant service time. This hypothesis represents a reasonable approximation for a large part of current routing and switching devices. The model notation is introduced in Table I.

Let $\{C_0, C_1, \dots, C_X\}$ and $\{P_0, P_1, \dots, P_Y\}$ be the set of sleeping and performance states available in the pipeline, respectively.

Each sleeping state is thought to be bound with both a different value of idle power consumption $\Phi_{idle}(C_x)$ and different transition times $\tau_{off}(C_x)$ and $\tau_{on}(C_x)$, needed to enter and to wake-up from the idle state, respectively. Let us suppose that a deeper sleeping state is characterized both by lower power consumption, and by a larger transition period.

In a similar way, each P state can be related with a different active power consumption $\Phi_a(P_y)$, as well as a different packet processing capacity $\mu(P_y)$. As the P_y state is higher, both the $\Phi_a(P_y)$ and the $\mu(P_y)$ values decrease.

However, transitions between the active state C_0 to the C_x state are not instantaneous, and a transition time τ_{off} is required. When new packets are received, the pipeline has to wake-up by exiting the C_x state and returning to the active one (this requires an additional τ_{on} period). Furthermore, depending on the specific HW/SW architecture and implementation, an additional time τ_{conf} is required to setup and to suitably configure the packet elaboration process. It is worth noting that, while τ_{on} and τ_{off} depend on the sleeping C_x state, the τ_{conf} parameter depends on the P_y state, since it represents a certain number of operations that have to be performed by the server, before re-starting packet-forwarding operations. The instantaneous power requirements can be expressed as follows:

$$\Phi = \begin{cases} \Phi_{idle}(C_x) & \text{if the server is in the } C_x \text{ state} \\ \Phi_a(P_y) & \text{if the server is in the } C_0 \text{ state} \\ \Phi_t(C_x) & \text{if the server is moving between } C_0 \text{ and } C_x \end{cases} \quad (1)$$

As in most HW platforms $\tau_{off} \ll \tau_{on}$, in the model derived in this paper, we neglect the τ_{off} period.

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 [14]. However, as sustained more recently in [15] and [16], 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 Zipf's law (which can be thought as the discrete version of the Pareto probability distribution).

C. The network- and energy-aware performance indexes

The model we propose corresponds to a M^x/D/1/SET queuing system [17]. 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. In order to take the LPI transition periods into account, the model considers deterministic server setup times. In more detail, when the system becomes empty, the server is turned off. The system returns operative only when a batch of packets arrives. At this point of time service can begin only after an interval $\tau = \tau_{on} + \tau_{conf}$ has elapsed.

Under such assumption and as demonstrated in [13], the average packet waiting time $\widetilde{W}$ can be expressed as follows:

$$\widetilde{W} = \frac{2\tau + \lambda\beta\tau^2 - \frac{1}{\lambda} + \frac{1}{\lambda\beta}\sum_{j=1}^{j_{max}} \beta_j j^2}{2(1+\lambda\beta\tau)} + \frac{\rho^2 - \beta + \sum_{j=1}^{j_{max}} \beta_j j^2}{2\lambda\beta(1-\rho)} \quad (2)$$

and the average power consumption as:

$$\widetilde{\Phi}_i = \frac{\left[\Phi_a\left(\frac{1}{\lambda}\rho+\beta\tau-(1-\rho)\tau_{on}\right)+(1-\rho)\left(\tau_{on}\Phi_t+\frac{1}{\lambda}\Phi_i\right)\right]}{\frac{1}{\lambda}+\beta\tau} \quad (3)$$

This model has been validated with respect to SW router architectures based on COTS HW. The results outlined its good accuracy, since the maximum estimation error was lower than 2% for both power consumption and packet latency times.

IV. 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 $\left\{C_x^{(i)}, P_y^{(i)}\right\}$ pair of states, and then with $\mu^{(i)} = \mu\left(P_y^{(i)}\right)$, $\tau^{(i)} = \tau\left(C_x^{(i)}\right), \tau_{on}^{(i)} = \tau\left(C_x^{(i)}\right)$, $\Phi_t^{(i)} = \Phi_t\left(C_x^{(i)}\right)$, $\Phi_i^{(i)} = \Phi_i\left(C_x^{(i)}\right)$ and $\Phi_a^{(i)} = \Phi_a\left(P_y^{(i)}\right)$.

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 $\hat{\beta}$.

Starting from the main achievements of previous works [1], and in order to make an optimal use of LPI primitives, we decided to not untie the incoming packet batches, and to send every packet composing a batch to a single pipeline. This design choice allows reducing the power consumption of the system according to a slight increase of packet latency times especially at low incoming traffic loads [1]. Under such assumptions, we can simply deduce that the process of incoming traffic is still BMAP, with the following parameters:

$$\beta^{(i)} = \hat{\beta} \quad (4)$$

$$\beta_j^{(i)} = \hat{\beta}_j \quad (5)$$

$$\sum_{i=0}^{\Lambda-1} \lambda^{(i)} = \hat{\lambda} \quad (6)$$

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

$$\widehat{\Phi} = \sum_{i=0}^{\Lambda-1} \widetilde{\Phi}^{(i)}\left(\lambda^{(i)}, C_x^{(i)}, P_y^{(i)}\right) \quad (7)$$

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

$$\widehat{W} = \sum_{i=0}^{\Lambda-1} \frac{\lambda^{(i)}}{\hat{\lambda}} \widetilde{W}^{(i)}\left(\lambda^{(i)}, C_x^{(i)}, P_y^{(i)}\right) \quad (8)$$

[1] The model and the load balancing criterion can be simply and suitably extended to consider the untying of packet batches, too.

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 Φ^*, the objective of the load balancing criterion is to find the best values of $\lambda^{(i)}$, $C_x^{(i)}$, and $P_y^{(i)}$ 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:

$$\begin{cases} \min_{\lambda^{(i)}, C_x^{(i)}, P_y^{(i)}} \gamma \frac{\widehat{\Phi}}{\Phi^*} + (1-\gamma)\frac{\widehat{W}}{W^*} \\ \widehat{W} < W^* \\ \widehat{\Phi} < \Phi^* \\ \sum_{i=0}^{\Lambda-1} \lambda^{(i)} = \hat{\lambda} \end{cases} \quad (9)$$

where the γ 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., $C_x^{(i)}, P_y^{(i)}\ \forall i = 0, \dots, \Lambda-1$) and continuous (i.e., $\lambda^{(i)}\ \forall i = 0, \dots, \Lambda-1$) variables.

By taking into account that the number of pipelines Λ, and of available C and P states are generally low, our minimization strategy mainly consists on solving the problem for each available configuration of C and P states of the pipelines. In more detail, for each feasible combination of $\left\{\left(C_x^{(0)}, P_y^{(0)}\right), \dots, \left(C_x^{(\Lambda-1)}, P_y^{(\Lambda-1)}\right)\right\}$, we find the best values of $\left\{\hat{\lambda}^{(0)}, \dots, \hat{\lambda}^{(\Lambda-1)}\right\}$ minimizing the objective function and satisfying the constraints.

Moreover, exploiting the last constraints in eq. 9, 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 $\lambda^{(i)}\ \forall i = 0, \dots, \Lambda-2$ inside the region satisfying the constraints, and in its frontier.

B. Analyzing the trade-off

In order to better understand and characterize the effects of the proposed optimization policy and the role of the trade-off parameter γ, we decided to perform some preliminary tests in presence of variable incoming load.

In more detail, we considered a packet processing engine with Λ=4 pipelines, and we used the parameters of a Xeon 5550 processor, generally used in Linux-based SW routers [12]. This choice is mainly because current HW routers do not include AR and LPI capabilities, and only their nominal and/or maximum power consumptions are reported in the datasheets.

Each pipeline corresponds to a processor core, and, as shown in Tables II and III, includes AR and LPI capabilities in terms of 4 available *P*-states, and 3 *C*-states (including the C_0 one), respectively. Previous experimentations on SW router architectures [12] suggest to use the values indicated in Table II for the τ_{on} parameter, and to fix $\tau_{conf} = \mu^{-1}$. The selection of a C- or P-state on a pipeline is fully independent from the other pipelines.

As far as the incoming traffic is concerned, by observing parameters in real traffic traces (e.g., see Fig. 12), we decided to fix $\hat{\beta} = 4$, while we increased the value of $\hat{\lambda}$ from 1 kpkt/s to 2.5 Mpkt/s (which, in our case, roughly corresponds to the threshold after that optimization constraints cannot be satisfied). The optimization problem has been solved for various values of the trade-off parameter, and, in more detail for γ =0, 0.25, 0.5, 0.75 and 1. The maximum latency W^* has been fixed to 50 μs, and the constraint on power consumption Φ^* to 250 W.

Figs. 3-7 show the optimal shares $\{\hat{\lambda}^{(0)}, \dots, \hat{\lambda}^{(3)}\}$ of incoming traffic load for each pipeline with respect to different values of γ. Figs. 8 and 9 report the estimated power consumptions and the packet latency times, respectively, in the optimal configurations. Figs. 10 and 11 shows how many pipelines are working in the available P- and C-states in the γ =0 and γ =0.75 cases.

By observing Figs. 3-7, we can outline how, in case of minimization of the latency times constrained to the energy consumption (i.e., $\gamma = 0$), the optimal policy suggests to uniformly divide the incoming load among the pipelines. Only for the highest load volumes ($\hat{\lambda} > 2.4$ Mpkt/s), this fairness is not maintained. In fact, in order to satisfy the power consumption constraint, the optimization policy maintains 3 pipelines with P_0 and C_1, and reduces the energy consumption of the whole engine by decreasing the performance of the pipeline 0. Accordingly, the load-balancer reduces the load share incoming to this pipeline.

On the contrary, when we minimize the power consumption for a given threshold on maximum latency times (i.e., $\gamma = 1$), the load balancer tries to concentrate as much traffic volume as possible into few pipelines. For instance and with reference to Fig. 7, the load-balancer redirects traffic only to the pipeline 3 at very low incoming volumes. When a change is needed on the C- or P-state configuration of pipeline 3 to satisfy the network performance constraints, the optimization policy decides to delay this configuration change, and to use also other (few) pipelines. However, by further increasing the incoming traffic load, the configuration change on the pipeline 3 becomes soon more energy-efficient, and the largest part of

TABLE II – POWER CONSUMPTIONS AND TRANSITION TIMES OF THE DEVICE'S *C*-STATES

C_x state	$\Phi_i(C_x)$	τ_{on}
C_0	Active	Active
C_1	10 Watt	10 ns
C_2	8 Watt	100 ns

TABLE III – POWER CONSUMPTIONS AND FORWARDING CAPACITIES OF THE DEVICE'S *P*-STATES.

P_y state	$\Phi_a(P_y)$	μ
P_3	50 Watt	650 kpkts/s
P_2	60 Watt	770 kpkts/s
P_1	70 Watt	890 kpkts/s
P_0	80 Watt	1010 kpkts/s

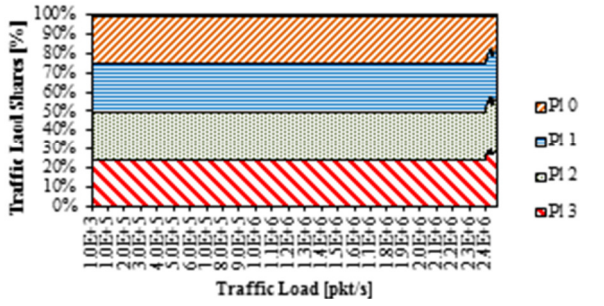

Fig. 3. Optimal load shares for each pipeline (Pl) and for $\gamma = 0$ according to increasing traffic volumes.

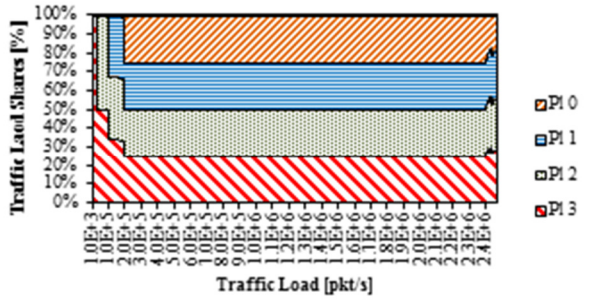

Fig. 4. Optimal load shares for each pipeline (Pl) and for $\gamma = 0.25$ according to increasing traffic volumes.

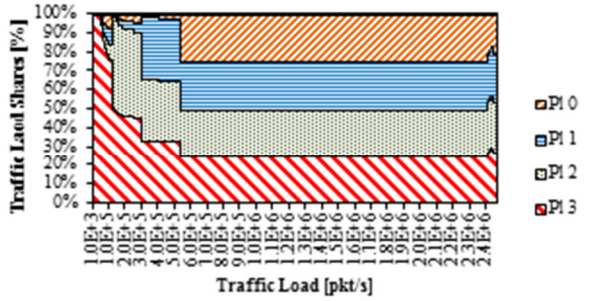

Fig. 5. Optimal load shares for each pipeline (Pl) and for $\gamma = 0.5$ according to increasing traffic volumes.

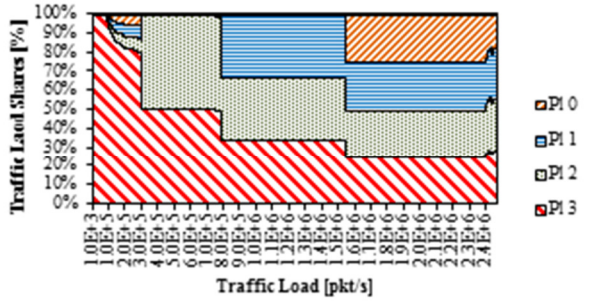

Fig. 6. Optimal load shares for each pipeline (Pl) and for $\gamma = 0.75$ according to increasing traffic volumes.

the load returns on this pipeline. When $\hat{\lambda} > 1.5$ Mpkt/s, the optimization policy starts to distribute traffic among pipelines in a more and more fair way in order to satisfy the W^* constraint.

Regarding energy consumption and average latency times, the $\gamma = 1$ case exhibits a nearly linear behavior on $\widehat{\Phi}$ with respect to $\hat{\lambda}$, while $\widehat{W}$ is almost equal to W^* for the largest part $\hat{\lambda}$ values. This behavior is sensibly different with respect to the case $\gamma = 0$, where $\widehat{\Phi}$ increases with a concave trend according to $\hat{\lambda}$, and $\widehat{W}$ values remain much lower than W^*.

As far as the other values of γ are concerned (see Figs. 4-6), the optimization policy roughly behaves as the minimization of power consumption ($\gamma = 1$) at low traffic volumes, and as the minimization of packet latency ($\gamma = 0$) at higher loads. The macroscopic role of the trade-off parameter γ appears to be moving the point where the optimization policy switches between the minimization of power consumption and the maximization of network performance: as γ increases, as the region with unfair traffic share enlarges. This role is also evident in Fig. 8, where the power consumptions of the cases γ=0.25, 0.5 and 0.75 start by agreeing with the γ=1 curve, and increasing $\hat{\lambda}$ they finish, one by one, by meeting the γ=0 values. As γ raises, as such meeting point happens at higher traffic volumes. By observing Figs. 10 and 11, we can outline also that the P- and C-states transitions become more frequent according to γ.

V. 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 [18] and part of "A Day in the Life of the Internet" [19][2]. We used a 96-hour-long traffic trace divided into sequential time windows of 15 minutes. Thus, for each time window, we applied our optimization policy with the same values of γ of section IV.B. Moreover, to obtain the results in this section we left the same packet processing engine configuration, and the same values of W^* and of Φ^* of the previous section.

As far as the incoming traffic is concerned, for each time window, we used the λ, β, and β_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. 12 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.

Figs. 13 and 14 show the estimated values for $\widehat{\Phi}$ and $\widehat{W}$, respectively, in the optimal configuration with respect to the traffic trace time windows and different values of the trade-off

[2] In order to meet the Software Router capacities in Table III, we increased the traffic volumes in the original trace by a scaling factor of 30.

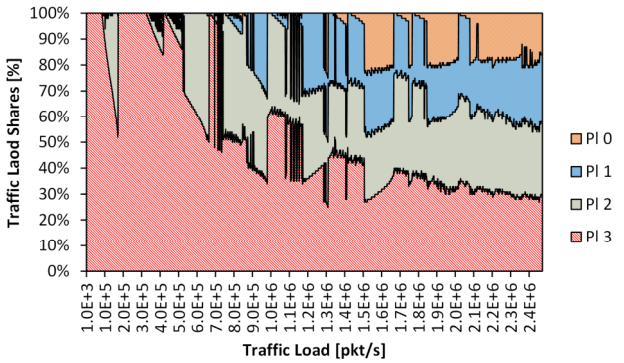

Fig. 7. Optimal load shares for each pipeline (Pl) and for $\gamma = 1$ according to increasing traffic volumes.

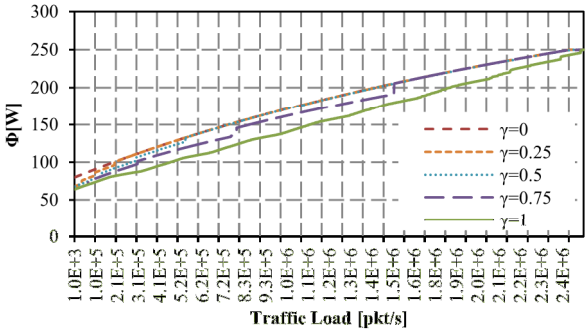

Fig. 8. Average power consumption of the packet processing engine with respect to γ and increasing values of $\hat{\lambda}$.

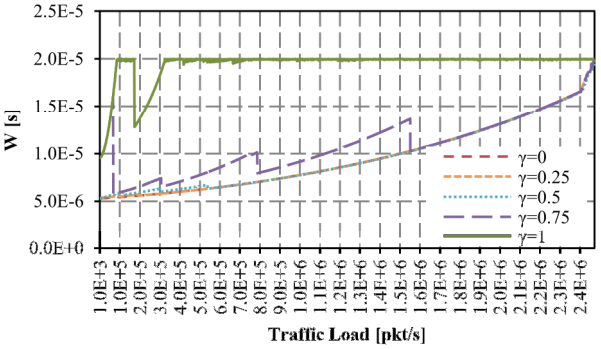

Fig. 9. Average packet latency times of the processing engine with respect to γ and increasing values of $\hat{\lambda}$.

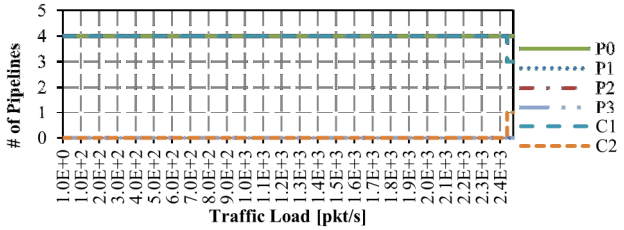

Fig. 10. Number of pipelines working in the P_y and in C_x state for $\gamma = 0$.

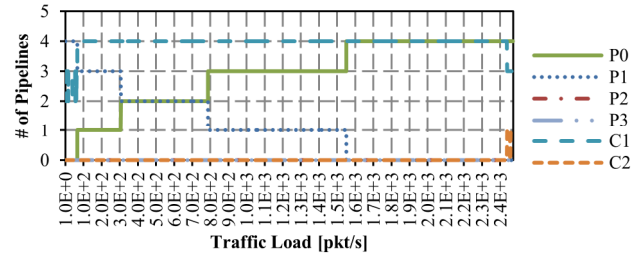

Fig. 11. Number of pipelines working in the P_y and in C_x state for $\gamma = 0.75$.

parameter γ. In the same scenario, Figs. 15 and 16 shows how many pipelines are using a certain P- or C-state, respectively.

These figures clearly outline how the optimization policies for γ =0, 0.25 and 0.5 provide almost the same results. As discussed in subsection IV.B, this behavior is mainly because, in case small value of γ (as 0.25 and 0.5), the optimization policy behaves like the pure minimization of packet latency after low volumes of incoming traffic, and the volumes in the considered traffic trace are higher than these thresholds. However, in case of $\gamma = 1$, the optimization policy allows saving about 12% of energy respect to $\gamma = 0$. On the other side, with $\gamma = 1$, the average packet latency time is always near to the W^* value. Finally, for $\gamma = 0.75$, we have an energy saving of 2.5% with respect to $\gamma = 0$, and the $\widehat{W}$ values appear to be a bit higher (max 5µs) than $\gamma = 0$ especially during low load periods (from 00:00 AM to 9:00 AM).

VI. Conclusions

In this paper, we considered energy-aware network devices (e.g. routers, switches, etc.) able to trade their energy consumption for packet forwarding performance by means of both low power idle and adaptive rate schemes. We focused 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 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 indexes. With this aim, we proposed and analyzed a constrained optimization policy, which optimize the 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 experimental data from SW router architectures and real-world traffic traces.

The obtained results showed that the proposed optimization policy, for low traffic volumes, roughly corresponds to the minimization of energy consumption constrained to a maximum packet latency. For higher values, the same policy starts to maximize network performance for a given energy-cap. By tuning the trade-off parameter in the proposed

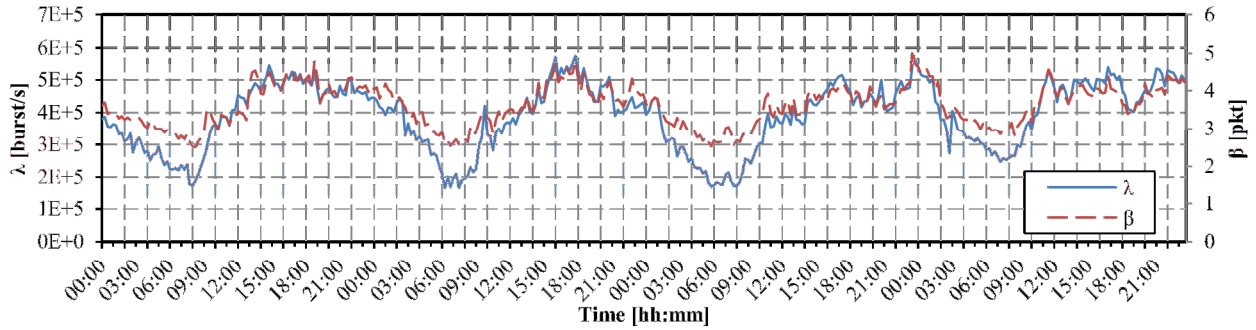

Fig. 12. Values of $\hat{\lambda}$ and $\hat{\beta}$ as extract from the real traffic trace in [18].

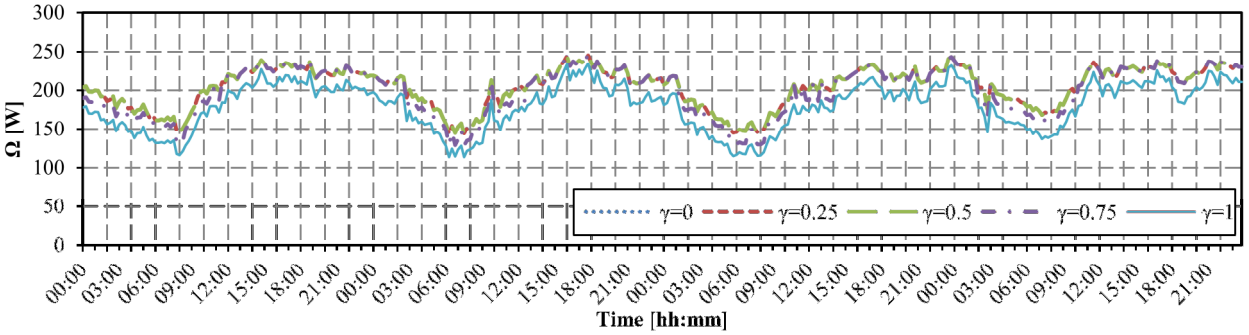

Fig. 13. Power consumption $\widehat{\Phi}$ for various value of γ with respect to the traffic trace in [18].

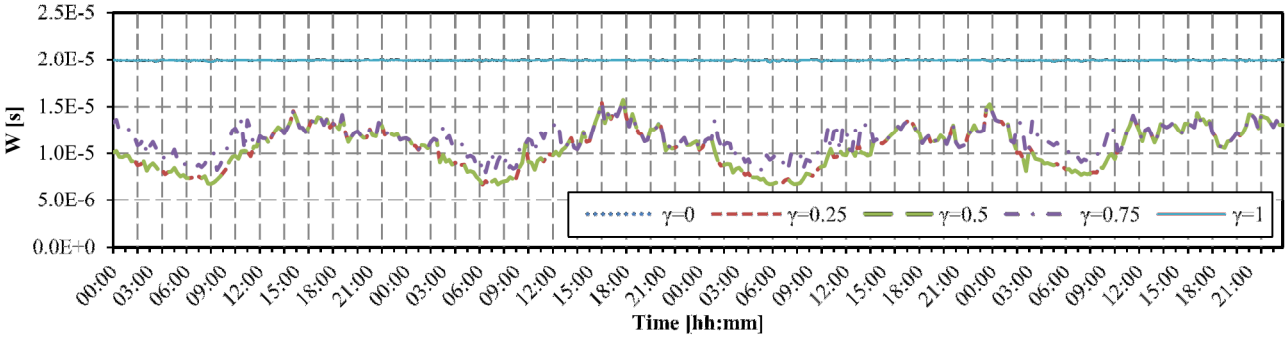

Fig. 14. Average latency times $\widehat{W}$ for various value of γ with respect to the traffic trace in [18].

objective function, we can control at which incoming load the policy switches between the two behaviors.

REFERENCES

[1] Nedevschi S., Popa L., Iannaccone G., Wetherall D. and Ratnasamy S., "Reducing Network Energy Consumption via Sleeping and Rate-Adaptation", Proc. of the 5th USENIX Symp. on Networked Systems Design and Implementation, San Francisco, CA, 2008, pp. 323-336.

[2] Bolla R., Bruschi R., Christensen. K., Cucchietti F., Davoli F. and Singh S., "The Potential Impact of Green Technologies in Next Generation Wireline Networks - Is There Room for Energy Savings Optimization?", to appear in IEEE Commun. Mag..

[3] Bolla R., Bruschi R., Davoli F., and Cucchietti F., "Energy Efficiency in the Future Internet: A Survey of Existing Approaches and Trends in Energy-Aware Fixed Network Infrastructures", to appear in IEEE Commun. Surveys and Tutorials (COMST).

[4] R. S. Tucker, R. Parthiban, J. Baliga, K. Hinton, R. W. A. Ayre, W. V. Sorin, "Evolution of WDM Optical IP Networks: A Cost and Energy Perspective", IEEE Journal of Lightwave Technology, vol. 27, no. 3, pp. 243-252, Feb 2009.

[5] Neilson, D.T., "Photonics for switching and routing," IEEE Journal of Selected Topics in Quantum Electronics (JSTQE), vol.12, no.4, pp.669-678, July-Aug. 2006.

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

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

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

[9] Z. Yi, and P.J. Waskiewicz, "Enabling Linux Network Support of Hardware Multiqueue Devices" Proc. of 2007 Linux Symposium, Ottawa, Canada, June 2007, pp. 305-310.

[10] San Martin R. and Knight J., "Power-Profiler: Optimizing ASICs Power Consumption at the Behavioral Level", Proc. of the 32nd ACM/IEEE Conf. on Design Automation, 1995, pp. 42-47.

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

[12] Bolla R., Bruschi R. and Ranieri A., "Green Support for PC-based Software Router: Performance Evaluation and Modeling", Proc. of the 2009 IEEE Internat. Conf. on Communications (ICC09), Dresden, Germany, June 2009.

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

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

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

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

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

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

[19] "A Day in the Life of the Internet" project, website available at http://www.caida.org/projects/ditl/.

ECONET Project, http://www.econet-project.eu.

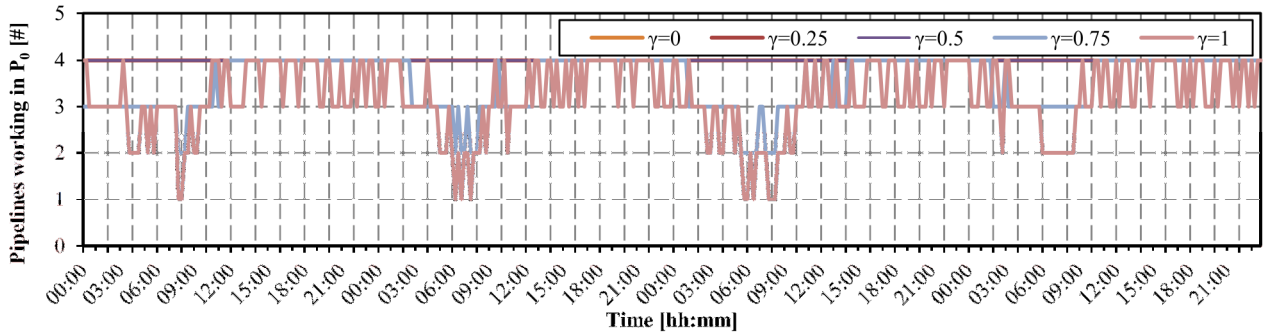

Fig. 15. Number of pipeline working in P_0 for various value of γ with respect to the traffic trace in [18].

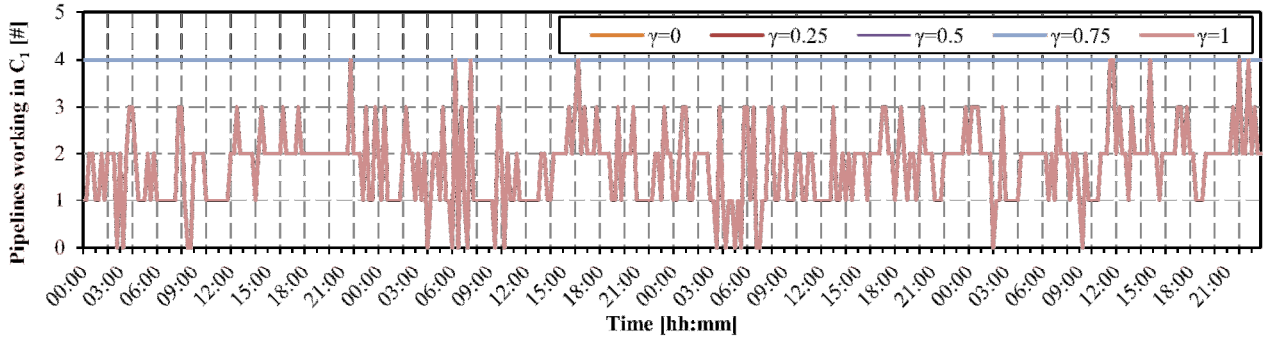

Fig. 16. Number of pipeline working in C_1 for various value of γ with respect to the traffic trace in [18].