Designing Optimal Energy Profiles for Network Hardware

Raffaele Bolla[1,2], Roberto Bruschi[2], Franco Davoli[1,2]

1. DITEN, University of Genoa
Genoa, Italy
{name.surname@unige.it}

2. CNIT
Genoa, Italy
{name.surname@cnit.it}

***Abstract*—The energy management of computer hardware mainly relies on primitives defined in the ACPI industrial standard. Network devices can borrow the same concepts as a basis for their energy-efficient design; however, to effectively tailor network processors for their specific applications, the set of energy-aware states defined by the ACPI and their parameters should be carefully chosen. In particular when designing network hardware, one should address the problem of optimizing parameter values (most often chosen in a discrete set) to trade-off energy efficiency and network performance. In this paper, we provide a methodology for such choice, by investigating this trade-off over a continuous spectrum of parameter values. This approach will allow us to gain insight in the behavior of the optimal solutions that achieve different desired tradeoffs.**

Keywords-Energy-efficiency; Network Processors; Power-Performance Tradeoff.

I. INTRODUCTION

The increasing processing capabilities [1] of network processors (to be used in routing, switching and home-gateway devices), combined with the growth of networking and network-attached devices and with the related energy consumption, has spawned a number of investigations on the power management of these systems [1]. Among other solutions, dynamic adaptation techniques are a promising tool to achieve better network energy efficiency.

Two basic mechanisms that can be jointly adopted in this respect are Low Power Idle (LPI) and Adaptive Rate (AR) [2, 3]. The former puts the hardware into low power consumption states during idle periods; the latter exploits clock frequency–power dependence, by adjusting the operating frequency when the processor is working. Both techniques are implemented by defining "energy-aware states" (corresponding to stable hardware configurations) that can be described by means of some parameters. More specifically, LPI requires the specification of power levels and wake-up times, whereas AR is described in terms of power levels and associated processing speeds. In computer hardware, dynamic adaptation techniques have long been used and are specified by the Advanced Configuration and Power Interface (ACPI) industrial standard [3]. Network devices can borrow the same concepts as a basis for their energy-efficient design; however, to effectively tailor network processors for their specific applications (where quality of service constraints are much tighter than in general purpose processors), the set of energy-aware states defined by the ACPI and their parameters should be carefully chosen. In particular when designing network hardware, one should address the following question: which energy-aware states and corresponding parameter values should be implemented to best suit the requirements of networks in terms of both energy-efficiency and performance related indexes?

In this paper, we investigate the whole spectrum of possible variations, by adopting continuous parameter values. This will allow us to gain insight in the behavior of the optimal solutions that achieve different desired tradeoffs. In particular, instead of working with a predefined set of possible parameters characterizing AR and LPI (as is the case for general purpose hardware from many manufacturers following the ACPI recommendation), we want to determine the most suitable values for network-specific hardware (e.g., network processors, ASICs, FPGAs), which can then be passed along as guidelines to the electronic designer to pick the subset of discrete states that best approximate the desired behavior, providing that the cost overhead associated with dynamic adaptation is not in excess of the benefits achieved.

The paper is organized as follows. We give an overview of energy-adaptation capabilities in CMOS devices in Section II. Section III introduces the tradeoff between power management and network performance, which is then specified for a packet processing engine model in Section IV. We state our parametric optimization problem in Section V. Section VI is devoted to the illustration and discussion of numerical results. Section VII contains the conclusions.

II. ENERGY ADAPTATION PRIMITIVES

In order to understand how adaptive green network technologies and relative HW implementations can be effectively designed and applied in next generation network devices, we have to take into consideration the main features of power management approaches in state-of-the-art HW technologies. Without losing of generality, this section introduces the main benefits and drawbacks of power management for the CMOS technologies. Such considerations can be easily extended to other technologies. The consumption of a CMOS circuit arises from two main contributions [6], namely leakage and dynamic power consumption:

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

The Φ_{dyn} contribution somehow represents the "ideal" power absorption of the circuit, since it is due to the real

[1] This work has been supported by the ECONET project co-funded by the European Commission under the 7th Framework Programme, from the EFFICIENT PRIN and GreenNet FIRB projects funded by the Italian Ministry of Research and Education.

transition of CMOS logical states. In more detail, Φ_{dyn} can be expressed as:

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

where ν 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 [7] [8]. Regarding the DFS methods, acting on f allows linearly scale Φ_{dyn} as shown in Eq. 2. However, it is worth noting that this operation results also 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 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 [9]:

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

where V_t is the threshold voltage (used for distinguish the "1" and "0" logic levels), and α is the velocity saturation index which usually ranges between one and two.

To lower the supply voltage without impacting the overall performance of a system, the system can run at a higher voltage during periods of high workloads and run at a lower voltage during periods of low workloads. 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. In more detail, the relationship $f < 1/t_d$ has to be held for guaranteeing stable hardware 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 Φ_{dyn} and the silicon performance, and further gains in the $\Phi_{leakage}$ contribution. From a more 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 [10]. 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 [7] [8]. 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 dramatically savings, but at the cost of longer wake-up times.

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

- *During active periods*: by reducing the elaboration capacity, and then increasing the elaboration times. Φ_{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 Φ_{dyn} and $\Phi_{leakage}$.

Such two approaches are clearly considered by the ACPI standard for 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 [2], namely Adaptive Rate (AR) and Low Power Idle (LPI), respectively.

III. Power Management vs Network Performance

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]. Preliminary studies in this field [11] 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, 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. 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 are needed, and the HW can remain in a low consumption state for longer periods.

IV. The Model

In order to represent an energy-aware packet processing engine with LPI and AR capabilities, we adopted the model in [5]. This model is founded on classical concepts of queuing theory, and it is specifically designed to estimate energy- and network-aware performance indexes. We assume to model the packet computation engine as a single server queuing system with maximum service rate μ. The μ service rate represents the device capacity in terms of packet headers that can be processed per second. We assume all packet headers requiring a constant service time. The model notation is introduced in

TABLE 1. NOTATION DEFINITION.

τ_{on}	time needed to wake up the HW
τ_{off}	time needed to put the active HW into the sleeping state
τ_{setup}	time to recover forwarding operation after the HW wakeup
μ	packet service rate
Φ_{a}	power consumption when the server is active
Φ_{idle}	power consumption when the server is sleeping
$\Phi_{\mathrm{t}}(C_x)$	power consumption during τ_{off} and τ_{on} periods
τ	server vacation time, $\tau = \tau_{\mathrm{on}} + \tau_{\mathrm{setup}} + \tau_{\mathrm{off}}$
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 $\rho = \frac{\lambda\beta}{\mu}$

TABLE 1. The selection of different AR and LPI configurations is supposed to impact on the performance in terms of both the packet service capacity, and wakeup times of the server. LPI configurations are bound with different values of both idle power consumption Φ_{idle} and transition times τ_{off} and τ_{on}, needed to enter and to wake-up from idle modes, respectively. The deeper is a sleeping state, the larger is the transition period. Each AR configuration can be related with a different active power consumption Φ_{a}, as well as a different packet processing capacity μ. Also in such case, the higher is μ, the larger is Φ_{a}. Transitions between the active and the sleeping state are not instantaneous, and a transition time τ_{off} is required. When new packets are received, the server has to wake-up and returning to the active state (this requires a τ_{on} period). Furthermore, depending on the specific HW/SW architecture and implementation, an additional τ_{setup} is required to setup the packet elaboration process. The instantaneous power requirements can be expressed as follows:

$$\Phi = \begin{cases} \Phi_{\mathrm{idle}}(\mathrm{C_x}) & \text{if the server is idle} \\ \Phi_{\mathrm{a}}(\mathrm{P_y}) & \text{if the server is active} \\ \Phi_{\mathrm{t}}(\mathrm{C_x}) & \text{if waking up} \end{cases} \quad (4)$$

As in most HW platforms $\tau_{off} \ll \tau_{on}$, in the model derived in [5], we neglect the τ_{off} period.

A. Continuous relaxation

We introduce a simple model to link network performance with power consumption indexes. In more detail, we express the energy consumption of the server when active and using AR techniques through the following empirical model [18]:

$$\Phi_{\mathrm{a}} = \Phi_a^{max}\left(\frac{\mu}{\mu_{max}}\right)^{\gamma} + \Phi_i^{max} \quad (5)$$

where μ_{max} represents the maximum packet forwarding capacity of the server, when no DFS or DVFS techniques are applied; Φ_a^{max} is the energy consumption of the server when working at the maximum speed μ_{max}; the exponential parameter γ is meant to represent the different trade-offs of DFS and DVFS techniques (and it consequently ranges from 0, in case of no AR primitives available, to 3, in case of AR realized by means of the DVFS); Φ_i^{max} is the maximum power consumption when the HW is in idle state. It is worth noting that $\Phi_{\mathrm{a}}^{\mathrm{max}}$ and $\Phi_{\mathrm{i}}^{\mathrm{max}}$ roughly correspond to the maximum Φ_{dyn} and $\Phi_{leakage}$ contributions of Eq. 1, respectively, when no power management primitives are applied. In fact, Eq. 5 implies that AR techniques may impact only on the dynamic part of circuit power absorption. Regarding the LPI primitives, we assume the energy consumption during idle periods to depend on τ as follows [19]:

$$\Phi_i = \Phi_{\mathrm{a}}\frac{1}{(1+\chi\,\tau)^{\delta}} \quad (6)$$

where χ and δ are parameters that express the implementation efficiency level of the LPI techniques. Suitable ranges for such parameters in real HW circuits for network devices are $\chi \in [10^5, 10^8]$ and $\delta \in [0, 4]$.

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 [12]. However, as sustained more recently in [13] and [14], a Batch Markov Arrival Process (BMAP) can effectively estimate the network traffic behavior. Therefore, we decided to model incoming traffic through a BMAP with 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 Pareto probability distribution).

C. The network- and energy-aware performance indexes

The model we apply corresponds to a M^x/D/1/SET queuing system [15]. 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 $\tau = \tau_{on} + \tau_{setup}$ has elapsed.

Under such assumption and as demonstrated in [5], the average packet waiting time $\widetilde{W}$ can be expressed as follows:

$$\widetilde{W} = \frac{2\tau+\lambda\beta\tau^2-\frac{1}{\lambda}+\frac{1}{\lambda\beta}\sum_{j=1}^{j_{max}}\beta_j j^2}{2(1+\lambda\beta\tau)} + \frac{\rho^2-\beta+\sum_{j=1}^{j_{max}}\beta_j j^2}{2\lambda\beta(1-\rho)} \quad (7)$$

and the average power consumption as:

$$\widetilde{\Phi} = \frac{\left[\Phi_a\left(\frac{1}{\lambda}\rho+\beta\tau-(1-\rho)\tau_{on}\right)+(1-\rho)\left(\tau_{on}\Phi_t+\frac{1}{\lambda}\Phi_i\right)\right]}{\frac{1}{\lambda}+\beta\tau} \quad (8)$$

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.

V. THE OPTIMIZATION PROBLEM

By using the expressions derived above, we can now state a parametric optimization problem to look for the optimal parameter settings (μ and τ) that minimize the average power, under a given Quality of Service (QoS) constraint (in terms of average packet processing delay). In practice, once known the traffic parameter values – which can be estimated in a router from the observation of the traffic traces – we would have a means of setting the corresponding optimal configuration of processor speed and idle state for the given workload, and of changing it upon a significant change in the workload, in an

adaptive fashion. We are therefore interested in solving the following optimization problem:

$$\begin{cases} \min_{\mu,\tau} \widetilde{\Phi} \\ \widetilde{W} \leq W^* \end{cases} \quad (9)$$

Besides the performance constraint on the maximum average delay, there are other constraints to account for the maximum allowable utilization, the maximum processor speed, and the positivity of the HW wake up time; namely,

$$\rho < 1,\ \mu \leq \mu_{max} \text{ and } \tau \geq 0 \quad (10)$$

We can then write the following Karush-Khun-Tucker conditions for problem (9)-(10) given the array of multipliers $\underline{k} = \{k_1, k_2, k_3, k_4\}$:

$$\begin{cases} \nabla\widetilde{\Phi} + k_1\nabla\widetilde{W} + k_2\nabla\rho + k_3\nabla\left(\frac{\mu}{\mu_{max}}\right) - k_4\nabla\,\tau_{on} = 0 \\ k_1\left(\widetilde{W} - W^*\right) = 0 \\ k_2(\rho - 1) = 0 \\ k_3\left(\frac{\mu}{\mu_{max}} - 1\right) = 0 \\ k_4\tau = 0 \end{cases} \quad (11)$$

Considering the continuous version of the parameters has basically two advantages: (*i*) it allows us to make use of efficient and widely available numerical optimization techniques; (*ii*) it allows exploring the whole range of parameter values, thus providing guidelines for the design of new energy-efficient network HW (e.g., network processors, ASICs, FPGAs). Regarding the second point, it is worth noting that, though parameters will be eventually available in a discrete form after HW design, their values may be fixed and/or constrained during designing process. A careful analysis of the optimal points for various traffic load configurations can then be useful to suggest the most suitable parameter values to the electronic designer.

VI. Performance Analysis

Subsection VI.A analyzes the impact of AR and LPI efficiency. This analysis is devoted also to outline which values of μ and τ have to be applied to achieve the optimal savings while meeting QoS constraints. Starting from the results of this analysis, in subsection VI.B traffic load patterns from real network infrastructures are used for driving the proposed optimization problem. This gives us the opportunity of deeply understanding which energy aware configurations have to be made available in next-generation green devices in order to maximize their efficiency. All the tests reported in this section have been carried out by fixing the following parameters: $\beta = 2$ pkts, $W^* = 50$ µs, $\tau_{setup} = 0$ s, $\Phi_t = \frac{1}{2}\Phi_a + \frac{1}{2}\Phi_{idle}$, $\Phi_a^{max} = \Phi_i^{max} = 100$ W, $\mu_{max} = 1.5$ Mpps.

A. The Impact of Green Technology Efficiency

This subsection is devoted to analyze how the efficiency parameters of power management primitives impact on the overall power savings and on which values of μ and τ have to be applied to achieve the optimal savings. To this purpose, subsections 1 and 2 introduce a parametric analysis of AR and LPI performance, respectively.

1) Adaptive Rate Technologies

Figures 1 and 2 report the results of the optimization framework in terms of average power consumption and optimal values of μ and τ with respect to increasing traffic load and γ. In more detail, we considered four different cases, namely $\gamma = 0$ (no AR optimizations), $\gamma = 1$ (AR optimizations by means of DFS), $\gamma = 2$ (AR optimizations by means of DVFS with α=1 in Eq. 3), and $\gamma = 3$ (AR optimizations by means of DVFS with α=2 in Eq. 3). In the performed tests the LPI primitive is enabled, and parameterized with the following "high performance" configuration: $\delta = 4$, and $\chi = 10^7$. As one can expect, the higher is the γ value, the lower is the average energy consumption. The energy gains start differing according the γ values as incoming traffic load increases. This is a clear effect of the presence of LPI primitives, which are for sure the dominant energy-aware primitive for low values of traffic load.

The consumption does not scale in a linear way with the γ increase: in the presence of LPI, AR through DFS ($\gamma = 1$) allows additional energy savings of less than ∼3-5% with respect to the case with only the LPI primitive available ($\gamma = 0$). Both the DVFS cases ($\gamma = 2$ and $\gamma = 3$) allow significant savings by cutting the average power absorption up to ∼20% and ∼27% with respect to the only LPI adoption. Figure 2 reports how the μ and τ values guaranteeing the optimal solution of the proposed optimization problem move with respect the incoming load for all the considered γ values.

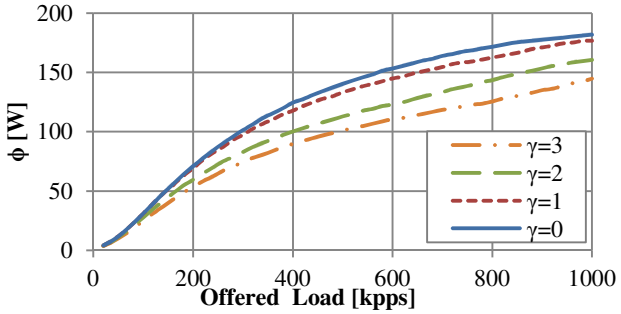

Figure 1. Average power consumption versus the incoming traffic load provided by the proposed optimization framework and according to different γ values.

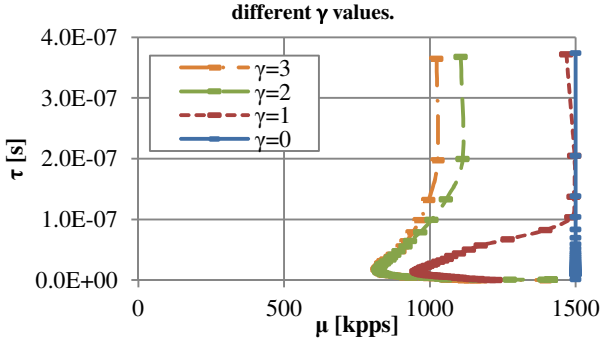

Figure 2. Locus of the extremal points provided by the proposed optimization framework and according to different γ values and incoming traffic loads.

The markers on the curves are placed at steps of 20 kpps of the incoming load. The first marker for each γ curve corresponds to a load equal to 20 kpps, and results in values of τ larger than 350 ns. As the incoming traffic load increases, optimal τ starts decreasing at an exponential pace, and gets very near to 0 s for incoming loads larger than 60-70% of μ_{max}. At a first glance, optimal μ values exhibit a quite strange behavior: for low incoming traffic volumes (less than 3% of μ_{max}), high service capacities (more than 1 Mpps) are chosen. The same service capacity values are chosen also when

incoming traffic overcomes ∼50% of μ_{max}. Indeed, this seemingly strange behavior is due to the well-known *race-to-idle* effect [16]. This effect implies that for low traffic loads is more convenient serving traffic at higher speeds in order to maximize the idle time, and using an aggressive LPI configuration. As the traffic load overcomes 3% of μ_{max}, the optimal values of μ start decreasing since the efficiency of AR techniques becomes relatively comparable to the one of the LPI primitive. Then, near to 50% of μ_{max}, the optimal values of μ change their trend, and starts to rapidly increase in order to assure the QoS constraints. It is worth noting that the larger is γ, the lower are the more frequent values chosen for μ. Moreover, while the DFS provides very high μ values (often equivalent to μ_{max}), the two DVFS cases (γ = 2 and γ = 3) provide similar optimal μ and τ values.

2) Low Power Idle Technologies

In order to evaluate the LPI potential impact, we analyzed two main study cases similar to the one in the previous subsection. In the former, the performance analysis is performed by varying the δ parameter, the latter by varying the χ one. In both the examined cases, we fixed $\gamma = 3$ and $\chi = 10^7$.

Figures 3 and 4 show the average consumption and the locus of optimal μ and τ values, respectively, according the incoming load and for various δ values. The case $\delta = 0$ corresponds to the absence of LPI optimization. The cases δ = 1, 2, 3, and 4 to increasingly efficient implementations of the LPI primitive (i.e., guaranteeing lower and lower Φ_i as δ raises). LPI primitives allow saving in the presence of low incoming traffic volumes: up to 75% for $\delta = 1$ and up to 95% for $\delta = 4$ with respect to the case of no LPI optimizations available. Such power gains quickly decrease with respect to the traffic load, and become negligible when the load overcome ∼35% of μ_{max}.

By observing Figure 4, we can note how δ mainly impacts on the optimal configurations in the presence of low traffic volumes. The more efficient is the LPI primitive, the more evident the *race-to-idle* effect becomes (i.e., higher μ values are chosen at low loads as δ increases). Regarding the case with $\delta = 0$, the race-to-idle effect is obviously not present, and the optimal μ values linearly increase with the load.

Finally, Figures 5 and 6 report results similar to the previous case, but according to various χ values (γ and δ were fixed to 3 and 4, respectively). The χ parameter represents somehow the responsiveness of the LPI primitives. Fixed τ, the larger is χ, the lower is Φ_i. The potential impact of χ is evident in Figure 5: as the value of this parameter increases, as the energy gain due to the LPI primitives becomes more significant also for higher incoming traffic loads. In case of χ=5 10^7, the LPI implementation becomes so efficient that the power consumption increases even in a linear way with respect to the load, and the optimal values of μ and τ appear to be limited to very narrow ranges (Figure 6), too. As far as the other χ values are concerned, Figure 6 exhibits optimal configurations with very similar behavior to the ones obtained in Figure 4.

B. The Impact in the Presence of Real Network Utilization

This section aims at understanding which values of optimal parameter would be most likely to occur in real operator networks. To this purpose we used typical link utilization profiles of three network operators, namely Telecom Italia, GRNET and NASK as reported in [17], and shown in Figure 7.

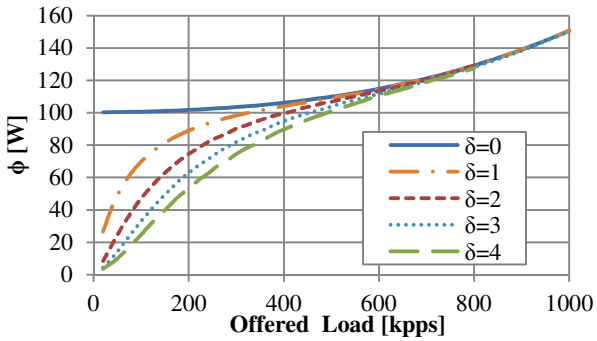

Figure 3. Average power consumption versus the incoming traffic load provided by the proposed optimization framework and according to different δ values.

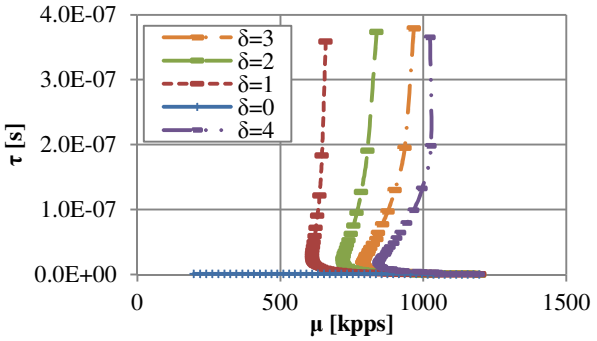

Figure 4. Locus of the extremal points provided by the proposed optimization framework and according to different δ values and incoming traffic loads.

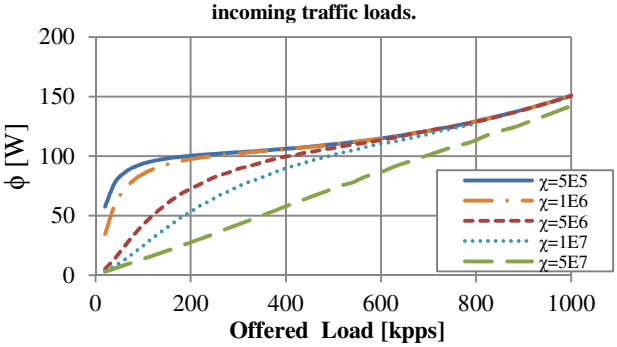

Figure 5. Average power consumption versus the incoming traffic load provided by the proposed optimization framework and according to different χ values.

The AR and LPI parameters have been fixed as follows: γ = 3, δ=4, and $\chi = 10^7$. Figures 8 and 9 show the normalized cumulative occurrences of the optimal values of μ and τ as obtained from the optimization framework and the probability distribution of link utilizations in Figure 7 for all the three considered network operators. Such results outline how the ranges of the selected values of μ and τ are similar in all the three cases, and relatively narrow (e.g., the selected μ values correspond to less than 13% of the available ones). The estimated power savings by the optimization framework obviously depend on the average link utilization; so that the gains of Telecom Italia case is a bit lower than the other two cases, since its reference network links exhibit higher utilization. However, the obtained figures are quite impressive since estimated average savings from 67% up to 83% with respect to the case of no energy optimizations.

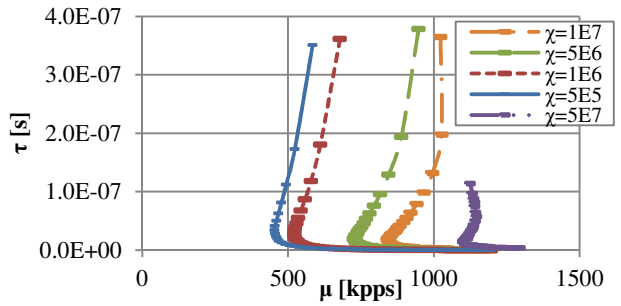

Figure 6. Locus of the extremal points provided by the proposed optimization framework and according to different χ values and incoming traffic loads.

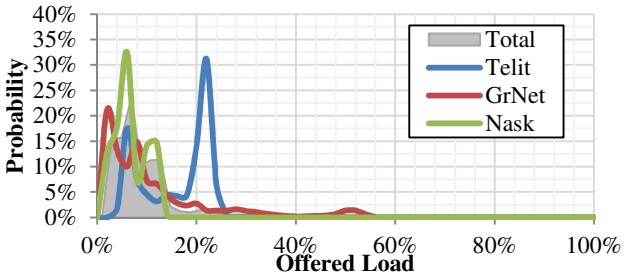

Figure 7. Probability distribution of the link utilization in some representative links of Telecom Italia, GRNET and NASK.

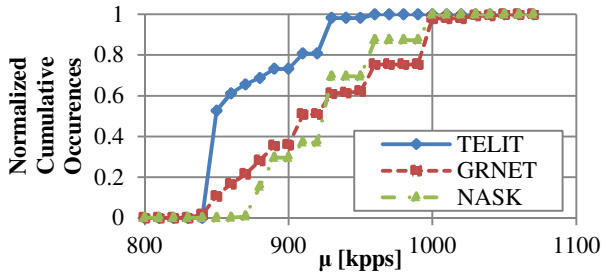

Figure 8. Normalized cumulative occurrences of the optimal values of μ for all the three considered network operators.

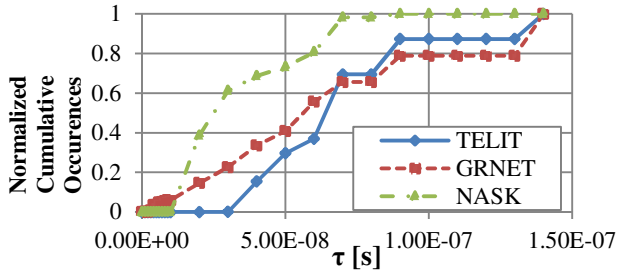

Figure 9. Normalized cumulative occurrences of the optimal values of τ for all the three considered network operators.

VII. CONCLUSIONS

We have considered the problem of choosing the optimal configuration of a network packet processor in order to minimize power consumption under a given performance constraint. The configuration is sought in terms of a couple of parameters that characterize the AR and LPI mechanisms: processing speed and wakeup time when exiting sleep mode. Instead of starting with a given (discrete) set of values for these parameters (as set by the manufacturer for a given piece of hardware), among which to seek the optimum for a certain traffic load, we have investigated the energy-performance tradeoff over a continuous parameter range. The loci of the optimal points give clear indications on the hardware settings that would best fit specific load values, which can be a valuable input to the electronic designer of network processing engines. We have characterized the opportunity of adopting DFS or DVFS techniques in terms of their potential energy saving in both AR and LPI. An indication of the optimal values that would be most likely to occur in real operator networks has also been given.

REFERENCES

[1] Cianfrani, A., Eramo, V., Listanti, M., Polverini, M., Vasilakos, A., "An OSPF-Integrated Routing Strategy for QoS-Aware Energy Saving in IP Backbone Networks," IEEE Trans. on Network and Service Manag. To appear.

[2] Bolla, R., Bruschi, R., Davoli, F., Cucchietti, F. "Energy efficiency in the Future Internet: A survey of existing approaches and trends in energy-aware fixed network infrastructures," *IEEE Comm. Surveys & Tutorials*, 13, 2 (2nd Qr. 2011), 223-244.

[3] Advanced Configuration and Power Interface Specification. [Online]. Available: http://www.acpi.info/DOWNLOADS/ACPIspec40a.pdf

[4] Ricciardi, S., Careglio, D., Fiore, U., Palmieri, F., Santos-Boada, G., Solé-Pareta, J., "Analyzing local strategies for energy efficient networking," Proc. of 1st W. on Sustain. Net., Valencia, Spain, May 2011.

[5] R. Bolla, R., Bruschi, R., Carrega, A., and Davoli F., "Green networking technologies and the art of trading-off," Proc. INFOCOM 2011 W. on Green Comm. and Net., Shanghai, China, April 2011, 301-306.

[6] Henzler, S. "Power Management of Digital Circuits in Deep Sub-Micron CMOS Technologies," Springer, Series in Advanced Microelectronics. 2007.

[7] Khandelwal, V., Srivastava, A. "Leakage Control Through Fine-Grained Placement and Sizing of Sleep Transistors," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Sys.*, 26, 7, July 2007, 1246-1255.

[8] De-Shiuan C., Shih-Hsin C., Shih-Chieh C., "Sleep Transistor Sizing for Leakage Power Minimization Considering Charge Balancing," *IEEE Trans. on VLSI Systems*, 17, 9, Sept. 2009, 1330-1334.

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

[10] Agarwal, K., and Nowka K., "Dynamic power management by combination of dual static supply voltages," *Proc. of the 8th Int. Symp. on Quality Electronic Design*, San José, CA, USA, March 2007, 82-95.

[11] Nedevschi S., Popa L., Iannaccone G., Wetherall D., Ratnasamy S. "Reducing Network Energy Consumption via Sleeping and Rate-Adaptation," *Proc. of the 5th NSDI Symp.*, San Francisco, CA, 2008, 323-336.

[12] Paxson V. and Floyd S., "Wide-area Traffic: The Failure of Poisson Modeling," IEEE/ACM Trans. on Net., 3, 3, 1995, 226-244.

[13] Salvador P., Pacheco A., Valadas R., "Modeling IP Traffic: Joint Characterization of Packet Arrivals and Packet Sizes Using BMAPs," *Computer Net.*, 44, 3, Feb. 2004, 335-352.

[14] Klemm A., Lindemann C., Lohmann M., "Modeling IP Traffic Using the Batch Markovian Arrival Process," *Computer Net.*, 54,2, Oct. 2003, 149-173.

[15] Choudhury G., "An M^X/G/1 Queueing System with a Setup Period and a Vacation Period," *Queueing Systems*, 36, 1-3, Jan. 200, 23–38.

[16] Garrett, M. "Powering Down," *Comm. of the ACM*, 51, 9, Sept. 2008, 42-26.

[17] The ECONET project, 2011. Deliverable 2.1. [Online] Available: https://www.econet-project.eu/Repository/DownloadFile/291

[18] Bolla R., Bruschi R., Carrega A., Davoli F., Suino D., Vassilakis C., Zafeiropoulos A., "Cutting the Energy Bills of Internet Service Providers and Telecoms through Power Management: an Impact Analysis," Computer Networks, to appear.