

An Analytical Model for Designing and Controlling New-Generation Green Devices

Raffaele Bolla, Roberto Bruschi, Alessandro Carrega, and Franco Davoli

Abstract—In this paper, we focus on energy-aware devices able to reduce their energy requirements by adapting their performance. We consider the device to be able to save energy through two main energy-aware primitives, namely, low power idle and power scaling. In such an environment, we propose a novel and original model for accurately representing how the joint usage of the previously cited primitives can impact on both energy consumption and network performance. As shown by the results achieved, the proposed model can be effectively applied in order to design and control energy-aware hardware of next-generation network devices.

Index Terms—green networks; low power idle, power scaling.

I. INTRODUCTION

RECENTLY, triggered by the increase in energy price, the continuous growth of customer population, the spreading of broadband access, and the expanding number of services being offered by telecoms and Internet Service Providers (ISPs), the energy efficiency issue has become a high-priority objective for wired networks and service infrastructures [1] [2]. To support the new generation network for a rapidly increasing customer population, telecoms and ISPs need an ever larger number of devices, with sophisticated architectures able to perform more and more complex operations in a scalable way. Notwithstanding these trends, it is well known that networks, links and equipment are provisioned for busy or rush hour load, which typically exceeds their average utilization by a wide margin [4]. While this margin is generally filled in rare short time periods, the overall power consumption in today's networks remains more or less constant with respect to different traffic utilization levels [1].

The key of any advanced power saving criteria certainly resides in dynamically adapting resources, provided at network, link or equipment levels, to current traffic requirements and loads. In this respect, current green networking approaches range from novel traffic engineering and routing criteria [3], to the introduction of energy-aware equipment [4], components [5] and network interfaces [6] [7].

Groundbreaking works in this area, like the one carried out by the 802.3az task force [8], already outlined that two power management primitives, namely Low Power Idle (LPI) and Power Scaling (PS) (or adaptive link rate if adopted in link

interfaces [8]), can be effectively used to save energy in active network devices. These power management mechanisms are far from being new in general purpose hardware, since they are a key feature in today's processors across all market segments, and are easily configurable through the Advanced Configuration Power Interface (ACPI). They respectively allow (i) minimizing power absorption when no activities are performed, and (ii) modifying the trade-off between performance and energy consumption when the hardware is active and performing operations. Moreover, the use of these two primitives is not exclusive, and, in the authors' opinion, may be jointly applied to fine tune the performance of devices, while minimizing their energy consumption [9] [10].

In this paper, we want to move these concepts significantly forward. Our main objective is providing a novel analytical model able to capture not only the trade-off between energy- and network-aware performance metrics, but also to accurately evaluate the mutual influence of LPI and PS and how they can impact the overall device behavior. To this purpose, the model explicitly takes different network- and energy-aware metrics into account. The proposed analytical model is based on classical concepts of queuing theory, and it can represent an interesting estimation tool, which well suits a large set of network switching architectures, technologies and components (from green link interfaces to packet processing engines). For these reasons, the proposed tool can be effectively adopted inside optimization procedures for the design and the dynamic control of next-generation green networking equipment.

The paper is organized as follows. LPI and PS, and how they are implemented in general purpose hardware through ACPI are introduced in section II. Then, we describe the proposed model in section III. Section IV reports numerical results, while the conclusions are in section V.

II. ENERGY-AWARE SILICON AND NETWORK PERFORMANCE

Nowadays, the largest part of current network equipment does not include LPI and PS capabilities, but power management is a key feature in today's processors across all market segments. In general purpose hardware, the ACPI standard models the LPI and PS primitives through the introduction of two sets of states, the so-called C- and P-states, respectively. Both these states can be individually employed and tuned for each core in the largest part of today's processors. Regarding the C-states, the C_0 power state is an active power state where the CPU executes instructions, while the C_1 through C_n power states are processor sleeping or idle

R. Bolla, R. Bruschi, A. Carrega and F. Davoli are with the Department of Communications, Computer and Systems Science (DIST), University of Genoa, Via all'Opera Pia 13, 16145 Genoa, Italy (phone: +39 010 353 2075; fax: +39 010 353 2154; e-mail: {raffaele.bolla, roberto.bruschi, alessandro.carrega, franco.davoli}@unige.it).

states, where the processor consumes less power and dissipates less heat. On the other hand, as the sleeping power state (C_1, \dots, C_n) becomes deeper, the transition between the active and the sleeping state (and vice versa) requires longer time. For example, as outlined in Table I, the transition between the C_0 and the C_1 states needs just only few nano-seconds, while 50 μ s are required for entering the C_3 state.

While in the C_0 state, ACPI 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. Using P states, a core can consume different amounts of power while providing different performance at the C_0 (running) 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. Current multi-core processors generally provide a certain number of feasible C and P states depending on the processor's technology [9]. Each processor's core can be individually configured for using a pair of C and P states independently of the other cores in the same processor. Unfortunately, due to issues in silicon electrical stability, the transition time between different P-states is generally slow, especially if compared with usual time scales in network dynamics: 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 policies with tight time constraints (where operating frequencies are throttled at packet- or flow-levels' time scale) are not feasible and cannot be adopted for optimizing power consumption inside network device architectures. The ACPI standard requires specific control applications, named governors, which are needed to dynamically configure power profiles in terms of C- and P-states through the ACPI standard interfaces. In more detail, the specific objective of such SW governors is to periodically optimize the configuration of ACPI devices with respect to their expected performance and computational load.

III. THE ANALYTICAL MODEL

The proposed model aims at representing the behavior and the performance of an energy-aware network device (or at least some of its sub-components), which includes LPI and PS primitives. For the sake of simplicity, we refer to LPI and PS primitives by using the ACPI terminology (i.e., in terms of C- and P-states). Our approach considers a network device substantially as a traffic forwarding engine that processes incoming packet headers at a finite capacity μ . The selection of different P- and C-states is supposed to impact on the forwarding engine performance in terms of both the packet service capacity and wakeup times. In detail, let $\{C_1, \dots, C_X\}$ and $\{P_0, P_1, \dots, P_Y\}$ be the set of sleeping and performance states available in the device, respectively. Each sleeping state is thought to be bound with both a different value of idle power consumption $\Phi_{\text{idle}}(C_x)$ and a different transition time $\tau_t(C_x)$, needed to enter the idle state and to wake-up from it. Let us suppose that a deeper sleeping state is characterized

TABLE I – INDICATIVE ENERGY SAVINGS AND TRANSITION TIMES FOR COTS PROCESSORS' C-STATES

C-State	Energy Saving with respect to the C_0 state	Transition Times
C_0	0%	-
C_1	70%	10 ns
C_2	75%	100 ns
C_3	80%	50 μ s
C_4	98%	160 μ s
C_5	99%	200 μ s
C_6	99.9%	unknown

both by lower power consumption and by a larger transition period.

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 y index is higher, both the $\Phi_a(P_y)$ and the $\mu(P_y)$ values decrease. For the sake of simplicity, in the rest of this paper we do not indicate explicitly the dependency of the $\Phi_a(\cdot)$ and the $\mu(\cdot)$ functions on their arguments.

Network devices are assumed working at small time scales by switching between a sleeping C_x $x \in [1, X]$ state, when idle, and a running P_y $y \in [0, Y]$ state, when performing operations. At larger time-scales, a governor can optionally modify the pair $\{C_x, P_y\}$ of selected states in order to meet the expected performance requirements and computational loads. When the governor changes the selected sleeping state C_x or the performance state P_y , it causes the device to work with a different trade-off between network performance and power consumption. For example, if a deeper sleeping state is selected, then the idle power consumption $\Phi_{\text{idle}}(C_x)$ will decrease, but forwarded packets will experience a larger latency, since transition times between C_0 and C_x will increase. In case of selection of a slower P state, the active power consumption $\Phi_a(P_y)$ decreases, and the packet processing capacity is reduced: this obviously causes the raise of packet service and latency times, as well as of packet loss probability.

Starting from the estimated incoming traffic load and from the device configuration (i.e., the selected pair of C and P states), the main objective of the model is two-fold: it has to accurately represent the energy consumption, as well as to estimate the packet forwarding performance in terms of packet latency times and loss rates. To this purpose, we model the packet computation engine of the network device as a single server queuing system with maximum service rate μ .

Similar to [9], the μ service rate is thought to represent the device capacity (i.e., the maximum number of packet headers that can be processed per second), and then it can be tuned by changing the operating P_y state. In detail, if the operating clock frequency increases (and then the index y of P state gets closer to 0), then an increasing number of packet headers can be processed per time unit. 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. A finite buffer, with size equal to N packets, is assumed to be associated to the server for backlogging incoming traffic.

Regarding the incoming traffic process, 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 [11]. However, as sustained more recently in [12], a Batch Markov Arrival Process (BMAP) can effectively estimate the network traffic behavior.

Starting from these considerations, we decided to model incoming traffic through a BMAP with LRD batch sizes. In other words, we assume to receive groups (or batches, or bursts) of j packets at exponential inter-arrival times with average value equal to $1/\lambda$. Coming back to our model, the sizes j of packet batches are supposed to follow Zipf's law:

$$\beta_j = \begin{cases} \frac{1}{j^s (\sum_{i=1}^{j_{\max}} \frac{1}{i^s})} & j \leq j_{\max} \\ 0 & j > j_{\max} \end{cases} \quad (1)$$

where β_j represents the probability that an incoming burst contains j packets, with $j \in [1, j_{\max}]$.

The average number of packets in a batch, β , is then obtained as:

$$\beta = \frac{\sum_{i=1}^{j_{\max}} \frac{1}{i^s}}{\sum_{i=1}^{j_{\max}} \frac{1}{i^s}} \quad (2)$$

Synthesizing, under the assumed hypothesis, in the proposed model the packet processing engine corresponds to a $M^x/D/1/N$ queueing system, where customers arrive in batches at Markov inter-arrival times with average rate λ , and are served by a single server at a fixed rate μ .

Before entering the analytical model's details, let us define:

- P_n as the stationary probability of having $n \in [0, N]$ packets in the queueing system;
- the traffic utilization ρ of the server, which in the case of infinite buffer can be expressed as:

$$\rho = \frac{\lambda\beta}{\mu} \quad (3)$$

The rest of this section aims at describing the proposed analytical model in detailed. Specifically, sub-sections III.A and III.B show the derivation of the “internal” parameters of the $M^x/D/1/N$ queueing system. Then, sub-sections III.C and III.D aim at obtaining network- and energy-aware performance indexes, respectively.

A. Deriving the stationary probabilities

In order to obtain the values of stationary probabilities P_n for $n \in [0, N]$, we derive the probability generating function for the related $M^x/D/1$ queueing system with infinite buffer. To this purpose, we firstly exploit the probability generating function for $M^x/G/1$ as shown in [13]:

$$P(z) = (1 - \rho) \frac{(1-z)[1-\lambda\alpha(z)(1-G(z))]}{1-z-\lambda\alpha(z)(1-G(z))} \quad (4)$$

where $|z| < 1$, and $G(z)$ and $\alpha(z)$ are defined as in the following:

$$G(z) = \sum_{j=1}^{\infty} \beta_j z^j \quad (5)$$

$$\alpha(z) = \int_0^{\infty} e^{-\lambda[1-G(z)]t} [1 - B(t)] dt \quad (6)$$

where $B(t)$ is the cumulative distribution function of service times. Thus, by recalling the hypothesis regarding deterministic service times, as well as the Zipf's batch distribution (Eq. 1), we can re-write the $G(z)$ and the $\alpha(z)$ functions as:

$$G(z) = \sum_{j=1}^{j_{\max}} \frac{z^j}{j^s (\sum_{i=1}^{j_{\max}} \frac{1}{i^s})} \quad (7)$$

$$\alpha(z) = \int_0^{\infty} e^{-\lambda[1-G(z)]t} \left[1 - \mathbb{1}\left(t - \frac{1}{\mu}\right) \right] dt = \int_{\frac{1}{\mu}}^{\infty} e^{-\lambda[1-G(z)]t} dt = \frac{1}{\lambda[1-G(z)]} \left[1 - e^{-\frac{\lambda}{\mu}(1-G(z))} \right] \quad (8)$$

where $\mathbb{1}(\cdot)$ is the Heaviside step function, and represents the cumulative distribution function of our deterministic service times $\frac{1}{\mu}$. By exploiting Eqs. 7 and 8, we obtain the probability generating function for a $M^x/D/1$ queueing system as follows:

$$P(z) = (1 - \rho) \frac{(1-z)e^{-\frac{\lambda}{\mu}(1-G(z))}}{e^{-\frac{\lambda}{\mu}(1-G(z))} - z} \quad (9)$$

Then, we can obtain the state probabilities P_n by calculating the Taylor series' coefficients of the $P(z)$ function:

$$P_n = \frac{1}{n!} \frac{\partial^n}{\partial z^n} P(z) \Big|_{z=0} \quad (10)$$

These coefficients can be obtained in closed form through simple derivation operations. Table II reports the P_n expressions obtainable for the first 3 values. It is worth noting that both function $P(z)$ in Eq. 8, as well as the stationary probabilities P_n in Table II are referred to the $M^x/D/1$ queue with an infinite buffer. However, assuming a low value of loss probability and similarly to [14], we can approximate the stationary probabilities of the finite buffer queueing system with the $\{P_0, P_1, \dots, P_N\}$ probabilities of the $M^x/D/1$ queue.

B. The server idle and busy times

When $\rho < 1$, a single-server queueing system is known to empty infinitely often. This obviously remains true also for our $M^x/D/1/N$ model. Hence, stochastic processes such as server's idle and busy periods are regenerative processes: they regenerate each time a packet batch arrival finds the system empty. Using classical principles of renewal theory, we can identify independent and identically distributed (iid) “cycles” of the form:

$$T_R^{(n)} = T_I^{(n)} + T_B^{(n)} \quad (11)$$

where $T_B^{(n)}$ is the n^{th} busy period, and $T_I^{(n)}$ is the n^{th} idle period. In more detail, both the $\{T_B^{(n)}\}$ and the $\{T_I^{(n)}\}$ sequences can be demonstrated to be iid. Thanks to the memory-less property of the exponential distribution of batch inter-arrival times, the inter-arrival time distribution when the last packet in the queue is served (and then the server enters a new idle period) is still:

$$P\{T_I^{(n)} \leq t\} = 1 - e^{-\lambda t} \quad (12)$$

TABLE II – CLOSED-FORM EXPRESSIONS FOR THE STATIONARY PROBABILITIES P_n IN A $M^x/D/1$ QUEUEING SYSTEM WITH ZIPF BATCHES.

	Stationary Probabilities
P_0	$1 - \rho$
P_1	$(1 - \rho) \left(e^{\frac{\lambda}{\mu}} - 1 \right)$
P_2	$(1 - \rho) e^{\frac{\lambda}{\mu}} \frac{-\lambda + \mu \left(e^{\frac{\lambda}{\mu}} - 1 \right) \sum_{i=1}^{j_{\max}} \frac{1}{i^s}}{\mu \sum_{i=1}^{j_{\max}} \frac{1}{i^s}}$
...	...

and then:

$$T_I = E\{T_I^{(n)}\} = \frac{1}{\lambda} \quad (13)$$

Let T_R and T_B be the average value of the renewal cycle periods and of the busy periods, respectively. Considering that the traffic utilization ρ can be expressed as:

$$\rho = \frac{T_B}{T_R} = \frac{T_B}{T_B + T_I} \quad (14)$$

we can obtain T_B as follows:

$$T_B = \frac{1}{\lambda} \frac{\rho}{1-\rho} \quad (15)$$

and consequently:

$$T_R = \frac{1}{\lambda} \frac{1}{1-\rho} \quad (16)$$

C. Network performance indexes

Starting from the stationary probabilities P_n $n \in [0, N]$ obtained in sub-section III.A, as well as the idle and busy periods in sub-section III.B, we can easily derive a large set of network performance indexes. The packet loss probability can be derived through the following approximation:

$$P_{loss} = 1 - \sum_{n=0}^N P_n \quad (17)$$

The mean value \bar{S} of packets in the queuing system is approximately equal to:

$$\bar{S} = \sum_{n=0}^N n \cdot P_n \quad (18)$$

while the average waiting time can be found by applying Little's law:

$$\bar{L} = \frac{\bar{S}}{\lambda \beta} \quad (19)$$

Beyond these classical parameters of network performance, we can introduce also the average burstiness index, which can be defined as the ratio between the average sizes of out-going and incoming packet bursts. In more detail, it can be expressed as:

$$\bar{B} = \frac{\mu T_B}{\beta} \quad (20)$$

Thus, recalling Eqs. 3 and 16, we obtain that:

$$\bar{B} = \frac{\mu \rho}{(1-\rho)\lambda \beta} = \frac{1}{(1-\rho)} \quad (21)$$

D. The Energy Consumption

As previously sketched, we assume the considered device to be configured for working with a fixed pair of C- and P-states $\{C_x, P_y\}$. Thus, the instantaneous power requirements can be expressed as follows:

$$\Phi = \begin{cases} \Phi_{idle}(C_x) & \text{if no operations are executed} \\ \Phi_a(P_y) & \text{otherwise} \end{cases} \quad (22)$$

where $\Phi_{idle}(C_x)$ represents the energy requirement of the considered device when it resides in the low-energy C_x state ($x > 0$), while $\Phi_a(P_y)$ is the power consumption when it is active at the P_y performance state.

Let us recall the renewal process representation of server's busy and idle periods (section III.B). As shown in Fig. 1, we know that during the $T_B^{(n)}$ periods, with $n = 0, 1, \dots, \infty$, the server is active and performing packet processing/forwarding activities, and then it has an instantaneous power consumption equal to $\Phi_a(P_y)$. Afterwards, when it serves the last backlogged packet, it enters the $T_I^{(n)}$ period, and it enters the low-consumption C_x state. However, as shown in Table I,

transitions between the active state, C_0 , to the C_x state are not instantaneous, and a transition time τ_t is required.

When new packets are received, the device has to wake-up by exiting the C_x state, and returning to the active one (and this requires an additional τ_s period).

Furthermore, depending on the specific device architecture and implementation, an additional time τ_s is required to setup and to suitably configure the packet elaboration process. For instance, as shown in [9], in Linux based SW routers, τ_s represents the computation time of HW interrupts that are used to trigger the packet forwarding process.

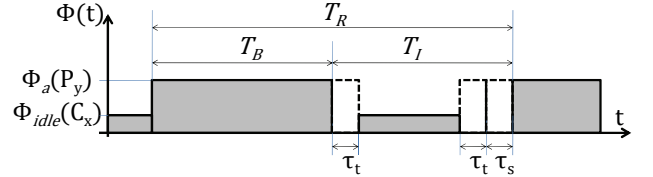


Fig. 1. Power consumptions during a renewal busy-idle cycle.

During a generic $T_I^{(n)}$, the instantaneous power requirement will be equal to $\Phi_{idle}(C_x)$ only if and when the device really resides in the C_x state. On the contrary, during the C_0 - C_x transitions and setup times, the power consumption is supposed to be $\Phi_a(P_y)$. Thus, we can express the energy wasted during a generic $T_I^{(n)}$ as follows:

$$E_I(T_I^{(n)}) = \begin{cases} \Phi_a(P_y) T_I^{(n)} & \text{if } T_I^{(n)} < \vartheta \\ \Phi_a(P_y) \vartheta + (T_I^{(n)} - \vartheta) \Phi_{idle}(C_x) & \text{if } T_I^{(n)} > \vartheta \end{cases} \quad (23)$$

where $\vartheta = 2\tau_t + \tau_s$.

We can obtain the average value \bar{E}_{idle} of the energy wasted during idle periods by exploiting the probability distribution function of idle period durations:

$$\bar{E}_{idle} = \int_0^\infty E_I(t) P\{T_I^{(n)} = t\} dt \quad (24)$$

then, by exploiting Eqs. 13 and 25:

$$\bar{E}_{idle} = \Phi_a(P_y) \int_0^\vartheta \lambda t e^{-\lambda t} dt + \int_\vartheta^\infty \lambda [\vartheta \Phi_a(P_y) + (t - \vartheta) \Phi_{idle}(C_x)] e^{-\lambda t} dt \quad (25)$$

we obtain the following expression:

$$\bar{E}_{idle} = \frac{1}{\lambda} [\Phi_{idle}(C_x) e^{-\lambda \vartheta} + \Phi_a(P_y) (1 - e^{-\lambda \vartheta})] \quad (26)$$

Regarding busy periods, the derivation of the average energy consumption can be obtained in a simpler way:

$$\bar{E}_B = E\{\Phi_a(P_y) T_B^{(n)}\} = \frac{1}{\lambda} \frac{\rho}{1-\rho} \Phi_a(P_y) \quad (27)$$

Therefore, the average steady-state power consumption $\bar{\Phi}$ of the considered device working at $\{C_x, P_y\}$ states can be finally obtained through the average consumption in a renewal period T_R . In detail, $\bar{\Phi}$ is equal to:

$$\bar{\Phi} = (1 - \rho) [\Phi_{idle}(C_x) e^{-\lambda \vartheta} + \Phi_a(P_y) \left(\frac{1}{1-\rho} - e^{-\lambda \vartheta} \right)] \quad (28)$$

IV. NUMERICAL RESULTS

In order to provide a complete performance evaluation, we decided to use a large data set from the real-world, and experimental devices as terms of comparison. Regarding the experimental devices, we used a Linux Software router, whose energy-aware capabilities were already evaluated and analyzed in [9]. This choice is mainly due to the fact that

current commercial/HW routers do not include or do not give the possibility to tune power management capabilities. On the contrary, in Linux Software routers, these capabilities are easily accessible through the ACPI interface. For the sake of simplicity, we used only a single processor core, which includes 4 C-states (including the C_0 one), and 4 P-states that provide different performance levels in terms of power consumption, forwarding capacity and transition times. Tables III and IV report the parameters for the C- and P- states, respectively.

TABLE III – POWER CONSUMPTIONS AND TRANSITION TIMES OF THE DEVICE’S C-STATES.

C_j state	$\Phi_{idle}(C_j)$	τ_t
C_1	10 Watt	10 ns
C_2	8 Watt	100 ns
C_3	7 Watt	50 μ s

TABLE IV – POWER CONSUMPTIONS AND FORWARDING CAPACITIES OF THE DEVICE’S P-STATES.

P_i state	$\Phi_a(P_i)$	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

Regarding incoming traffic, the tests were done by using a 96-hour long Internet traffic trace that is publicly available [15] and part of the “A Day in the Life of the Internet” [16]¹.

The evolution of the traffic offered load over the time of the used traffic trace is reported in Fig. 2, which exhibits the classical night-and-day profiles. The minimum traffic load is during the first hours of the morning (e.g., from 3:00 to 6:00), while rush hours are at 11:00 and 14:00.

Fig. 3 shows the traffic offered load decomposition in burst arrival rates and burst sizes, which were calculated by starting from the packet level trace, and averaging the data in different time slices. Each time slice has a fixed duration equal to 15 minutes. It is interesting to underline how an increase in incoming traffic volume is due to the rise of both burst arrival rate and burst sizes. Besides the average values of λ and β , in every time slice we obtained the estimation of all the parameters (i.e., s and j_{max}) of the batch probability distribution. These values were obtained by a least-squares fitting of the Zipf distribution in eq. 1 with the measured samples in the slice. The performance evaluation tests were performed in all the 15-minute time-slices, by estimating and/or measuring the energy consumption and the network performance with both the proposed analytical model and the software router. For the analytical model, we used the above-mentioned parameters estimated from the traffic trace, while for the Software Router we directly re-produced the packet level trace. Regarding other configuration parameters, the setup time τ_s has been fixed to a packet service time ($1/\mu$), and the buffer size N is fixed to 1000. Fig. 4 reports the average energy consumption estimated by the proposed model, and the relative estimation error with respect to the Software Router. The tests were performed according to all

possible combinations of P and C states. Fig. 5 shows the packet loss probability of the model and of the Software Router according to the C_1 state and different P-states. In the same testing conditions, Fig. 6 reports the average packet latency times. The results obtained outline how the P_3 state provides too slow performance for rush hours, and how the model slightly over-estimates packet losses and under-estimates packet latencies with respect to the Software Router. All these results demonstrate how the proposed model can provide a precision level suitable to be effectively adopted for representing the performance of energy-aware HW capabilities in the presence of real Internet traffic.

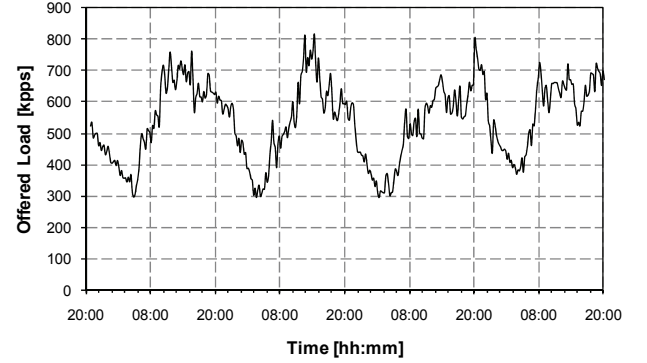


Fig. 2. Offered load in terms of packets per second measured in the traffic trace in [15].

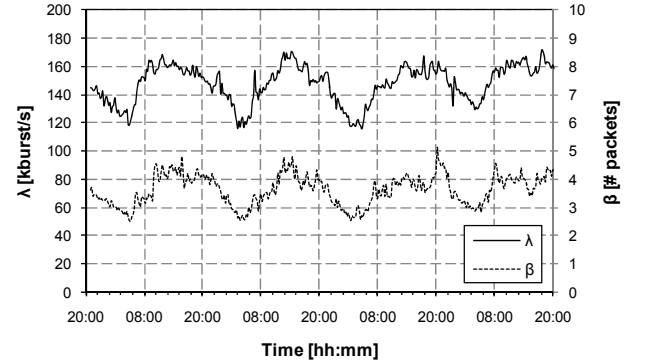


Fig. 3. Average values of λ and β measured in the traffic trace in [15].

V. CONCLUSION

In this contribution, we focused on performance-adaptive network devices, able to save energy by scaling their traffic processing capacities. In detail, we proposed a novel analytical model able to capture the impact of power management capabilities on network performance metrics. The analytical framework considers stochastic incoming traffic at packet level with LRD properties. The model explicitly takes a large set of features and parameters into account, like, for example, the transition times to enter a low-consumption sleeping state, or packet latencies and loss probabilities.

REFERENCES

- [1] Bianco, C.; Cucchietti, F.; Griffa, G., “Energy consumption trends in the next generation access network — a telco perspective,” Proc of the 29th Internat. Telecom. Energy Conf., 2007. (INTELEC 2007), Rome, Italy, Sept. 2007, pp. 737-742.
- [2] ITWales, “Green evangelist to call for big changes in computer use to aid environment at ITWales conference,” Nov. 2007, <http://www.itwales.com/997539.htm>.

¹ In order to meet the forwarding capacities of the Software Router, we increased the traffic volumes in the original trace by a scaling factor of 30.

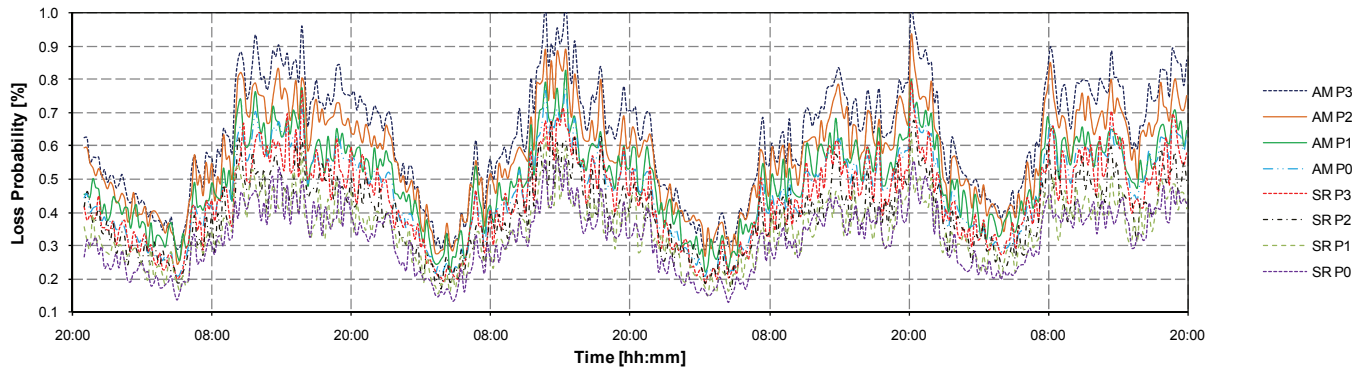


Fig. 4. Energy consumption estimated by the analytical model according to various C- and P-states, and maximum estimation error of the analytical model with respect to the SW router per each time slice (2nd y-axis).

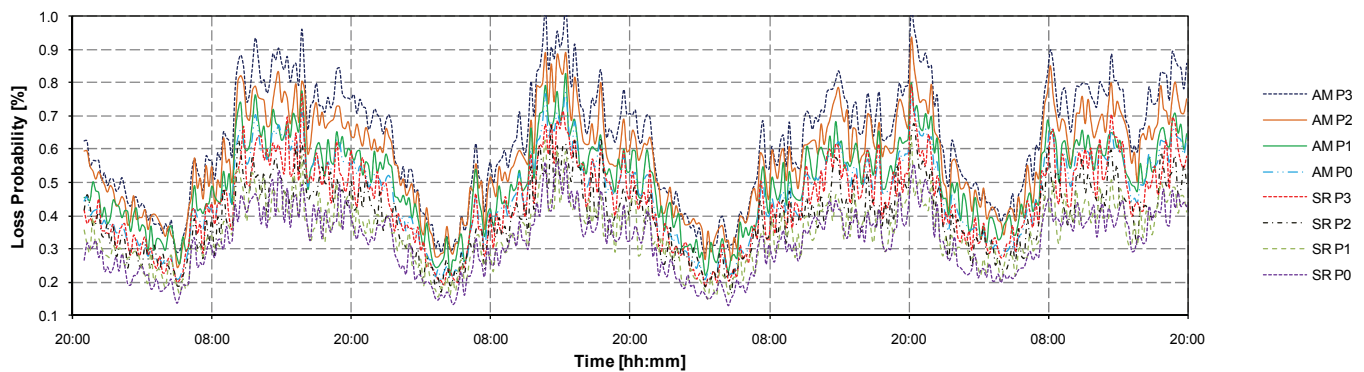


Fig. 5. Packet loss probability estimated by the model (AM), and measured on the SW router (SR) with respect to different P states and the C₁ state.

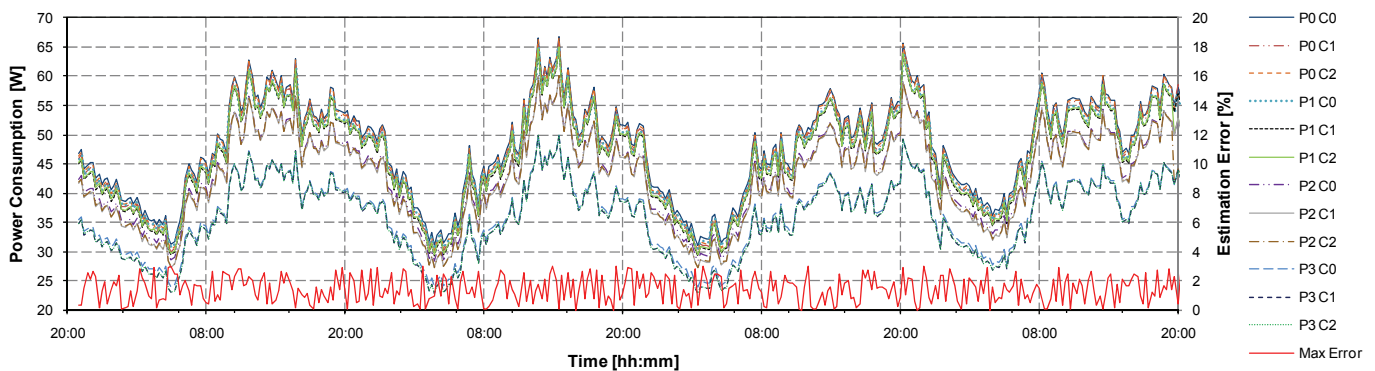


Fig. 6. Average packet latency estimated by the model (AM), and measured on the SW router (SR) with respect to different P states and the C₁ state.

- [3] Cardona Restrepo, J. C.; Gruber, C. G.; Mas Machuca, C.; "Energy Profile Aware Routing," Proc. Green Communications Workshop in conjunction with IEEE ICC'09 (GreenComm09), Dresden, Germany, June 2009.
- [4] Nedeveschi, S.; Popa, L.; Iannaccone, G.; Wetherall, D.; Ratnasamy, S.; "Reducing network energy consumption via sleeping and rate-adaptation," Proc. of the 5th USENIX Symposium on Networked Systems Design and Implementation, San Francisco, CA, 2008, pp. 323-336.
- [5] Noguera, J.; Kennedy, I.O., "Power Reduction in Network Equipment Through Adaptive Partial Reconfiguration," Proc. of the 2007 Internat. Conf. on Field Programmable Logic and Applications (FPL 2007), Aug. 2007, pp. 240-245.
- [6] Gupta, M.; Singh, S.; "Using Low-Power Modes for Energy Conservation in Ethernet LANs," Proc. of the 26th IEEE Conf. on Computer Comm. (Infocom07), Anchorage, Alaska, USA, pp. 2451-2455, May 2007.
- [7] Gunaratne, C.; Christensen, K.; Suen, S.; Nordman, B.; "Reducing the Energy Consumption of Ethernet with an Adaptive Link Rate (ALR)," IEEE Trans. Computers, vol. 57, no. 4, pp. 448-461, Apr. 2008.
- [8] Institute of Electrical and Electronics Engineers (IEEE), IEEE 802.3az Energy Efficient Ethernet Task Force, <http://grouper.ieee.org/groups/802/3/az/public/index.html>.
- [9] Bolla, R.; Bruschi, R.; Ranieri, A.; "Green Support for PC-based Software Router: Performance Evaluation and Modeling", Proc. of the 2009 IEEE Internat. Conf. on Communications (ICC'09), Dresden, Germany, June 2009. Best paper award.
- [10] Bolla, R.; Bruschi, R.; Ranieri, A.; Davoli, F.; "Performance Constrained Power Consumption Optimization in Distributed Network Equipment," Proc. Green Comm. Workshop in conjunction with IEEE ICC'09 (GreenComm'09), Dresden, Germany, June 2009.
- [11] Paxson, V.; Floyd, S.; "Wide-area Traffic: The Failure of Poisson Modeling", IEEE/ACM Trans. Networking, vol. 3, no. 3, pp. 226-244, June 1995.
- [12] Salvador, P.; Pacheco, A.; Valadas, R.; "Modeling IP Traffic: Joint Characterization of Packet Arrivals and Packet Sizes Using BMAPs", Computer Networks, vol. 44, no. 3, pp. 335-352, Feb. 2004.
- [13] Tijms, H. C.; "The Mx/G/1 Queue" in "A first course in Stochastic Models," John Wiley & Sons Ltd, Chichester, UK, 2003, pp. 360.
- [14] Kim, H. S.; Shroff, N. B.; "Loss Probability Calculations and Asymptotic Analysis for Finite Buffer Multiplexers," IEEE/ACM Trans. Networking, vol. 9, no. 6, pp. 755-768, Dec. 2001.
- [15] MAWI Working Group Traffic Archive, Sample Point F, available at <http://mawi.nyu.edu/mawi/samplepoint-F/20080318/>.
- [16] "A Day in the Life of the Internet" project, website available at <http://www.caida.org/projects/ditl/>.