Green Network Technologies and the Art of Trading-off

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

Abstract — **In this contribution, we focus on energy-aware devices able to reduce their energy requirements by adapting their performance. We propose an analytical model to accurately represent the impact of green network technologies (i.e., low power idle and adaptive rate) on network- and energy-aware performance indexes. The model has been validated with experimental results, performed by using energy-aware software routers and real-world traffic traces. The achieved results demonstrate how the proposed model can effectively represent energy- and network-aware performance indexes. Moreover, also an optimization procedure based on the model has been proposed and experimentally evaluated. The procedure aims at dynamically adapting the energy-aware device configuration to minimize energy consumption, while coping with incoming traffic volumes and meeting network performance constraints.**

Index Terms — **green networking; low power idle; adaptive rate.**

I. Introduction

In the last few years, Telecom operators, Internet Service Providers and public organizations reported statistics of network energy requirements and the related carbon footprint, showing an alarming and growing trend [1]. The Global e-Sustainability Initiative (GeSI) [2] estimates an overall network energy requirement of about 21.4 TWh in 2010 for European Telcos, and foresees a figure of 35.8 TWh in 2020 if no Green Network Technologies (GNTs) will be adopted. 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 [3]. 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 [4], [5]. In this respect, current green networking approaches range from novel traffic engineering and routing criteria, to the introduction of energy-aware equipment and network interfaces [5]. This paper focuses on analyzing and evaluating the impact of power scaling GNT on next-generation network devices. This is accomplished by adopting two basic techniques: *Adaptive Rate* (AR) and *Low Power Idle* (LPI)[1]. The former allows dynamically modulating the capacity of a link, or of a processing engine, in order to meet traffic loads and service requirements; the latter forces links or processing engines to enter low power states when not sending/processing packets and to quickly switch to a high power state when sending one or more packets. In such scenario, our main objective is to provide a novel analytical model based on classical concepts of queuing theory and able to capture the trade-off between energy- and network-aware performance metrics, when AR and/or LPI techniques are adopted in a network device. In order to validate the proposed model, we performed several tests by using real-world traffic traces, and compared the estimated performance indexes with experimental measurements, obtained with Component Off-The-Shelf (COTS) software (SW) routers. Moreover, we also consider a simple energy-aware optimization procedure based on the analytical model we propose. Such procedure aims at periodically adapting the energy-aware configuration of the device in order to minimize its power consumption, while meeting network Quality of Service (QoS) constraints and incoming traffic volumes.

The paper is organized as follows. Section II introduces AR and LPI capabilities and how they can impact on network performance. The proposed model is described in section III, and its validation results are in section IV. The optimization procedure based on the proposed model is explained in Section V. Section VI shows some performance evaluation results obtained with the optimization procedure. Finally, the conclusions are drawn in section VII.

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 [6]. The rest of this section is structured as follows. Sub-section A introduces how ACPI (Advanced Configuration and Power Interface) standards make AR and LPI capabilities accessible to the SW layer. Sub-section 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 [7] standard models AR and LPI functionalities by introducing two sets of energy-aware states, namely performance and power states (*P*- and *C*-states), respectively. 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

R. Bruschi is with the National Inter-University Consortium for Telecommunications (CNIT), Italy. (phone: +39-010-3532057; fax: +39-010-3532154; e-mail: roberto.bruschi@cnit.it)
R. Bolla, A. Carrega, and F. Davoli are with Department of Communications, Computer and Systems Science (DIST), University of Genoa, Via all'Opera Pia 13, 16145 Genova, Italy (e-mail: {raffaele.bolla, alessandro.carrega, franco.davoli}@unige.it).

[1] For the sake of clarity, even though LPI might be seen as a limiting case of AR, we prefer to explicitly distinguish the two techniques.

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 1, AR (Fig. 1c) obviously causes a stretching of packet service times while the sole adoption of LPI (Fig. 1b) introduces an additional delay in packet service, due to the wake-up times. Moreover, preliminary studies in this field [3] 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. 1d, 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 transitions (and wake-up times) are needed, and the HW can remain in a low consumption state for longer periods.

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
$\tau_{conf}(P_y)$	time to recover forwarding operation after the HW wakeup
$\tau_{setup}(P_y, C_x)$	setup time, $\tau_{setup}(C_x, P_y) = \tau_{on}(C_x) + \tau_{conf}(P_y)$
$\mu(P_y)$	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_{on} and τ_{off} periods
N	buffer size
λ	rate of batch arrival
β_j	probability that an incoming burst contains *j* packets
$X(z)$	Probability Generating Function (PGF) of batch sizes
β	average number of customers in a batch
P_n	stationary probability of having $n \in [0, N]$ packets in the queuing system
ρ	traffic utilization of the server, which can be expressed as $\rho = \lambda\beta/\mu$ in the case of infinite buffer
$B(s)$	Laplace transform of the customer service process
$V(s)$	Laplace transform of the vacation process due to setup times
T_I	average duration of server idle periods including τ_{off}
T_B	average duration of server busy periods including τ_{setup}
T_R	average duration of idle/busy renewal process

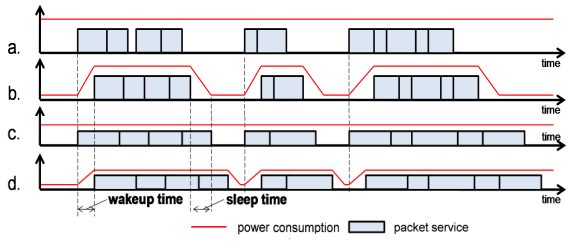

Fig. 1. Packet service times and power consumptions in the cases with: (a) no power-aware optimizations, (b) only LPI, (c) only AR, (d) AR and LPI.

III. THE ANALYTICAL MODEL

The model we propose aims to represent the behavior and the performance of an energy-aware network device, which includes LPI and AR capabilities. For the sake of simplicity, we 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 forwarding engine performance in terms of both packet service capacity and wakeup times of the server. Similarly to [8] and [9] as previously sketched, the service rate μ represents 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. A finite buffer, with a size equal to *N* packets, is assumed to be bound to the server for backlogging incoming traffic packets. The model notation is introduced in Table I. The rest of this section is organized as follows. Sub-section A introduces the main parameters to be considered in a device with AR and LPI capabilities. Subsection B shows the model representing the traffic incoming to the energy-aware device. Finally, the proposed analytical model is described in Sub-section C.

A. *Introducing the Energy-Aware Parameters*

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

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. In a similar way, each *P*-state can be related with active power consumption $\Phi_a(P_y)$, as well as with a packet processing capacity $\mu(P_y)$. As the P_y state becomes higher, both $\Phi_a(P_y)$ and $\mu(P_y)$ values decrease. From the considerations in section II, it is reasonable to assume the network device working at small time-scales by switching between a sleeping C_x state, when idle, and a running P_y state, when performing operations. For this reason, throughout the paper we do not explicitly indicate the dependency of parameters (e.g. $\Phi_a(\cdot)$, $\mu(\cdot)$, and $\tau_{on}(\cdot)$) on the *C*- and *P*-states. Fixed the state pair $\{C_x, P_y\}$, the system works with the renewal process representation shown in Fig. 2. The server has

infinitely many alternating busy $T_B^{(n)}$ and idle $T_I^{(n)}$ periods, where the index n denotes the order of the interval. During a generic $T_B^{(n)}$, the server is active and performing packet forwarding activities, and then has instantaneous power consumption equal to $\Phi_a(P_y)$. Afterwards, when it serves the last backlogged packet, it enters the $T_I^{(n)}$ period corresponding to the low-consumption C_x state.

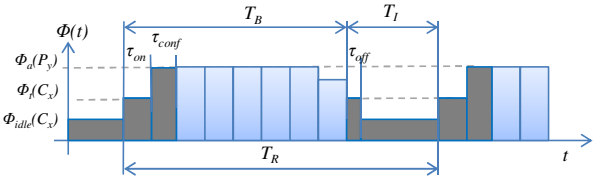

Fig. 2. Power consumptions during a renewal busy-idle cycle.

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 device 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 device 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 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 COTS 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 [10] [11]. However, as sustained more recently in [12] and [13], 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 exponentially distributed 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 regarded as the discrete version of a continuous Pareto probability distribution with parameter s). In more detail, we assume that incoming packet batches have the following probability mass function:

$$\beta_j = \begin{cases} \frac{1}{j^s\left(\sum_{i=1}^{j_{max}} \frac{1}{i^s}\right)} & j \le j_{max} \\ 0 & j > j_{max} \end{cases} \quad (2)$$

where β_j represents the probability that an incoming burst contains j packets, with $j \in [1, j_{max}]$. The average packet number in a batch, β, is then obtained as:

$$\beta = \frac{\sum_{i=1}^{j_{max}} \frac{1}{i^{s-1}}}{\sum_{i=1}^{j_{max}} \frac{1}{i^s}} \quad (3)$$

Thus, we obtain the Probability Generating Function (PGF) of batch sizes as:

$$X(z) = \sum_{j=1}^{\infty} \beta_j z^j = \sum_{j=1}^{j_{max}} \frac{z^j}{j^s\left(\sum_{i=1}^{j_{max}} \frac{1}{i^s}\right)} \quad (4)$$

C. The Proposed Queuing Model

The model we propose corresponds to a M^x/D/1/SET queuing system [14]. Packets arrive in batches with exponentially distributed 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. When the system becomes empty, the server is turned off. The system returns operative only when a batch of packets arrives. At this point in time service can begin only after an interval $\tau_{setup} = \tau_{on} + \tau_{conf}$ has elapsed.

The rest of this section introduces the analytical model and its specialization to our case. In sub-section III.C.1 we derive the PGF and the stationary probabilities of the M^x/D/1/SET queuing system; in sub-section III.C.2 we express the server's idle and busy periods. Then, we propose an approximation for the packet loss probability in the case of a finite buffer of size N and derive network- and energy-aware performance indexes in subsections III.C.3 and III.C.4.

1) The PGF and the stationary probabilities

In order to obtain the values of stationary probabilities P_n for $n \in [0, \infty]$, we exploit the PGF of the M^x/G/1 system as shown in [15] and [14]:

$$P(z; M^x/G/1) = (1-\rho)\frac{(1-z)B(\lambda - \lambda X(z))}{B(\lambda - \lambda X(z)) - z} \quad (5)$$

Under the assumption that service times are deterministic, we can express the Laplace transform of service times as:

$$B(s) = e^{-s/\mu} \quad (6)$$

Thus, we obtain that the PGF of the M^x/D/1 queuing system can be written as:

$$P(z, M^x/D/1) = (1-\rho)\frac{(1-z)e^{-\frac{\lambda}{\mu}(1-X(z))}}{e^{-\frac{\lambda}{\mu}(1-X(z))} - z} \quad (7)$$

By exploiting the stochastic decomposition results of Doshi [16] for the single unit arrival case and the results in [14] for bulk arrivals, the PGF of the M^X/G/1 queue with setup times turns out to be:

$$P_v(z) = \zeta(z) P(z, M^x/D/1) \quad (8)$$

where

$$\zeta(z) = \frac{1 - zV(\lambda - \lambda X(z))}{\left(\frac{1}{\beta} + \lambda\tau\right)(1 - X(z))} \quad (9)$$

is the PGF of the number of arrivals during the residual life of the vacation period, defined as an idle period plus a setup period τ_{setup} (for the sake of simplicity, throughout the rest of the paper we indicate τ_{setup} with τ). Since server setup times have constant durations equal to τ, we can express $V(s)$ as

$$V(s) = e^{-\tau s} \quad (10)$$

By using Eqs. (9) and (10) in Eq. (8), we can obtain the PGF of our M^x/D/1/SET system:

$$P_v(z) = (1-\rho)\frac{1 - ze^{-\lambda\tau[1-X(z)]}}{\left(\frac{1}{\beta} + \lambda\tau\right)[1 - X(z)]}\frac{(1-z)e^{-\frac{\lambda}{\mu}[1-X(z)]}}{e^{-\frac{\lambda}{\mu}[1-X(z)]} - z} \quad (11)$$

Remembering that the PGF is defined as:

$$P_v(z) = \sum_{n=0}^{\infty} P_n z^n \quad (12)$$

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

$$P_n = \frac{1}{n!}\frac{\partial^n}{\partial z^n} P_v(z)\Big|_{z=0} \quad (13)$$

Notwithstanding these coefficients can be obtained in closed form through simple derivation operations, we preferred to

evaluate such derivatives numerically, since numerical evaluation has a lower computational complexity than calculating the closed-form expressions of the derivatives at $z = 0$.

2) The server idle and busy times

Under server traffic utilization $\rho < 1$, a G/G/1 queuing system will become empty infinitely often. This obviously remains true also for our M^X/D/1/SET model. Hence, using classical principles of renewal theory, we can identify independent and identically distributed (iid) "cycles" of the form:

$$T_R^{(n)} = T_I^{(n)} + T_B^{(n)} \tag{14}$$

where $T_B^{(n)}$ is the n^{th} busy period, (corresponding to the "delay busy period" in [14], which includes the setup time), and $T_I^{(n)}$ is the n^{th} idle period. In more detail, both sequences $T_B^{(n)}$ and $T_I^{(n)}$ can be demonstrated to be iid. The average duration of idle and busy period are given by:

$$T_I = E\left\{T_I^{(n)}\right\} = \frac{1}{\lambda} \quad T_B = \frac{1}{\lambda}\frac{\rho}{1-\rho} + \frac{\beta\tau}{1-\rho} \tag{15}$$

We can obtain T_R as follows:

$$T_R = T_I + T_B = \frac{\frac{1}{\lambda}+\beta\tau}{1-\rho} \tag{16}$$

3) Network performance indexes

Starting from the stationary probabilities P_n obtained in sub-section III.C.1, as well as the idle and busy periods in sub-section III.C.2, we can easily derive a large set of network performance indexes. The mean value $\bar{L}$ of packets in the queuing system can be obtained by specializing the general expressions in [14] to our case of deterministic service time and Zipf-distributed packet batches:

$$\bar{L} = lim_{z\to 1} P'_v(z) = \frac{2\lambda\beta\tau+\lambda^2\beta^2\tau^2-\beta+\sum_{j=1}^{j_{max}}\beta_j j^2}{2(1+\lambda\beta\tau)} + \frac{\rho^2-\beta+\sum_{j=1}^{j_{max}}\beta_j j^2}{2(1-\rho)} \tag{17}$$

Using Little's law, the average waiting time $\bar{W}$ is:

$$\bar{W} = \frac{\bar{L}}{\lambda\beta} = \frac{2\tau+\lambda\beta\tau^2-\frac{1}{\lambda}+\frac{1}{\lambda\beta}\sum_{j=1}^{j_{max}}\beta_j j^2}{2(1+\lambda\beta\tau)} + \frac{\rho^2-\beta+\sum_{j=1}^{j_{max}}\beta_j j^2}{2\lambda\beta(1-\rho)} \tag{18}$$

It is worth noting that both the $P_v(z)$ function in Eq. (8) and the stationary probabilities P_n in Eq. (13) are referred to the M^X/D/1/SET queue with an infinite buffer. However, by assuming a low value of loss probability and similarly to [17], we can approximate the stationary probabilities of the finite buffer queuing system with the $\{P_0, P_1, \dots, P_N\}$ probabilities of the M^X/D/1/SET queue. In more detail, the average value of packet loss probability can be expressed through the following approximation:

$$P_{loss} = 1 - \sum_{n=0}^{N} P_n \tag{19}$$

The approximation might be used also to re-compute $\bar{L}$ and $\bar{W}$ for the finite buffer case. However, if P_{loss} is minute (as it actually turns out to be in most practical cases), Eqs. (17) and (18) already provide a good approximation.

4) The Energy Consumption

Recalling Fig. 2 and Eq. (1), we can express the average energy consumed in a renewal cycle as follows:

$$\tilde{\Phi} = \frac{[\Phi_a \cdot (T_B - \tau_{on}) + \Phi_t \cdot \tau_{on} + \Phi_{idle} \cdot T_I]}{T_R} \tag{20}$$

and by using Eqs. (15) and (16) in Eq. (20):

$$\tilde{\Phi} = \frac{\left[\Phi_a\left(\frac{1}{\lambda}\rho+\beta\tau-(1-\rho)\tau_{on}\right)+(1-\rho)\left(\tau_{on}\Phi_t+\frac{1}{\lambda}\Phi_{idle}\right)\right]}{\frac{1}{\lambda}+\beta\tau} \tag{21}$$

IV. Model Validation

In order to validate the proposed model, we took the multi-core Linux SW Router (SR) used in [8] as a term of comparison. This choice is mainly due to the fact that current HW routers do not include AR and LPI capabilities, and only their nominal and/or maximum power consumptions are reported in the datasheets.

The considered SW Router is equipped with several Gigabit Ethernet adapters with Receive-Side Scaling (RSS) support [18]. Eight cores, placed in two Xeon 5550 processors, perform all packet forwarding operations in a fully parallel and independent way among themselves. Each processor core 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 [8] suggest to use the values indicated in Table II for the τ_{on} parameter, and to fix $\tau_{conf} = \mu^{-1}$. In this scenario, our model represents the behavior of each single core, serving packets from reception interfaces. The parameters λ and β parameters are respectively the arrival rate and the average size of traffic batches processed by the considered core. For the sake of simplicity, we decided to show the validation results for a single processor core, receiving traffic from a single Gigabit Ethernet interface with a reception buffer size equal to 512 packets, and forwarding it towards another Gigabit Ethernet link. We performed the SW router experimentations and the proposed model estimation by using real-world traffic traces that are publicly available [19] and part of "A Day in the Life of the Internet" [20][2]. We used a 96-hour-long traffic trace divided into sequential time windows of 15 minutes. Thus, for each time window, we obtained energy- and network-aware performance indexes both with the SW router and with the proposed model. The SW router measurements were performed by using the test-bed composed by an Ixia N2X router tester [21] to reproduce traffic traces, and to measure packet losses and latency times with high accuracy levels, and an Agilent U2353A multifunction Data Acquisition (DAQ) device [22], to measure the processor power consumption. As far as the proposed model is concerned, for each time window, we used the λ, β, s, and j_{max} values calculated from the traffic trace. In detail, s and j_{max} parameters were obtained by least squares fitting of the Zipf distribution in Eq. (2) with the trace sample. The evolution of the traffic offered load over the time of the reference traffic trace is reported in Fig. 3 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 burst arrival rate and burst sizes. Fig. 4 reports the power consumption values estimated by the analytical model (AM), the values measured with the experimental test-bed, and the maximum estimation error in each time window. The AM estimation was obtained with Eq. (21).

The results in Fig. 4 outline the good accuracy level provided by the model. Moreover, they suggest that selecting too deep stand-by states may cause a rise on power consumption. This is simply caused by the non-negligible τ_{on} times to enter the deepest *C*-state. When the probability that burst inter-arrival time is larger than τ drops, the device

[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.

enters low-power sleeping states more and more rarely and for shorter periods, before waking up again. Figs. 5 and 6, respectively, show the average values of loss probability and packet latency times for both the SR and the AM, as well as the relative estimation error. The AM estimates of latency times were obtained with Eqs. (17) and (18), and loss probabilities were computed as in Eq. (19). Such results show that the proposed AM represents also network-aware performance indexes with a good accuracy level, since the errors are lower than 0.1% for loss probabilities, and lower than 2% for latency times. Regarding the AM complexity and execution times, the former depends linearly on the buffer size *N*, and the latter never exceeds 150 ms.

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

V. ENERGY-AWARE CONTROL

The proposed model can be viewed as an interesting estimation tool, which can be adopted for controlling AR and LPI capabilities in next-generation network devices. The model can be included in the optimization procedures in order to dynamically and optimally change the energy-aware device configuration with respect to the estimated traffic

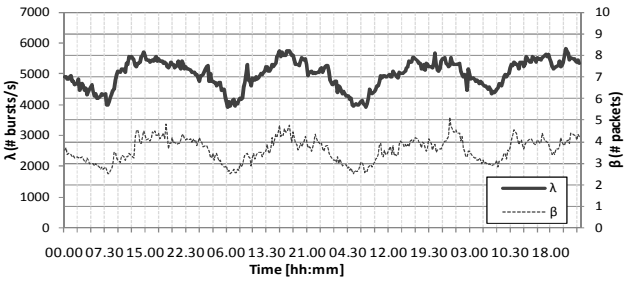

Fig. 3. Average values of λ and β measured in the traffic trace in [19].

load and performance requirements [9]. Such optimization procedures have to periodically select the optimal pair of $\{C_x, P_y\}$ states, which minimizes the device's power consumption while meeting the estimated load and network performance requirements (i.e., loss rate, latency times, etc.). Thus, given the estimate of incoming traffic load, in terms of λ, β, j_{max} and s, the optimization problem can be formalized as:

$$\begin{cases} min_{\{C_x,P_y\}}\ \tilde{\Phi}(\lambda,\beta) \\ \bar{W} \le W^* \\ P_{loss} \le P_{loss}^* \end{cases} \quad (22)$$

where W^* and P_{loss}^* are the maximum admissible average values of latency and loss probability, respectively. Thus, the optimization procedure starts by considering all the $\{C_x, P_y\}$ pairs that satisfy the performance constraints in Eq. (22). This is performed by estimating the average latency values with Eq. (18), and the loss probabilities through Eq. (19), by numerically obtaining the first *N* coefficients of the Taylor series expansion of the PGF in Eq. (13). For all the $\{C_x, P_y\}$ pairs satisfying the above constraints, we calculate the estimated value of the average power consumption with Eq. (21). The configuration guaranteeing the minimum

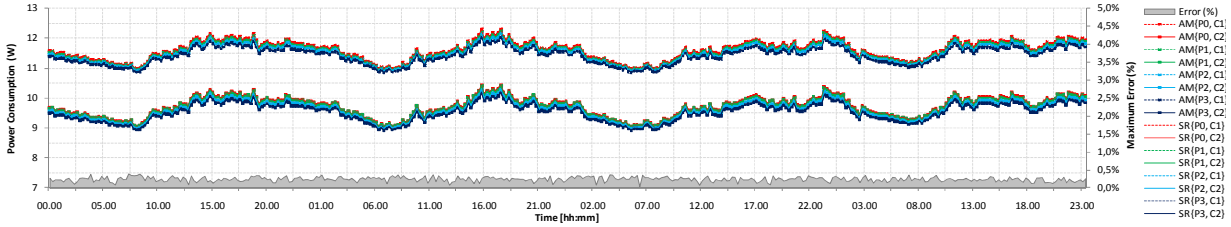

Fig. 4. Energy consumption estimated by the analytical model according to various configurations of *C*- and *P*-states, and maximum estimation error of the analytical model with respect to the SW router per each time slice.

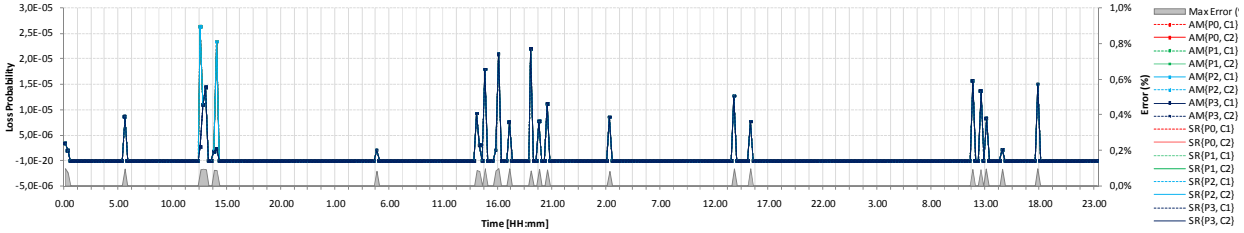

Fig. 5. Packet loss probability estimated by the AM and measured on the SR with respect to different *P*- and *C*-states, and maximum relative estimation error.

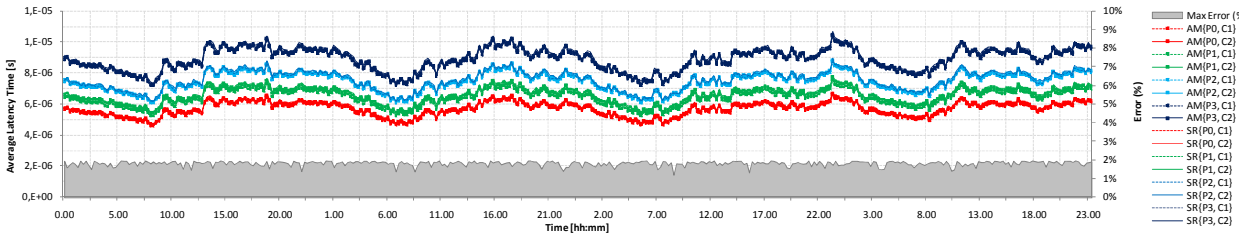

Fig. 6. Average packet latency estimated by the AM and measured on the SR with respect to different *P*- and *C*-states, and maximum relative estimation error.

consumption is finally selected.

VI. Performance Evaluation

In this section, we provide a performance evaluation of the optimization procedure introduced in section V. To this purpose, we exploited the same test-bed described in section IV. The optimization procedure works in every time window, by finding the best $\{C_x, P_y\}$ pair that minimizes energy consumption and satisfies the packet latency and loss constraints. The performance constraints of the model were $W^* = 10\,\mu s$, and $P^*_{loss} = 10^{-5}$. We decided to let the optimization procedure know the exact values of time window traffic parameters (λ, β, j_{max} and s) in advance. This choice gives us the chance of evaluating the optimization procedure performance in the absence of errors due to incoming traffic estimation. The results in Figs. 7 and 8 only depend on the accuracy level of the AM. We decided to take the $\{C_1, P_0\}$ SR configuration as a term of comparison, since it represents the most "conservative" case for network performance. Fig. 7 reports the power consumption for both cases considered, and underlines how the optimization procedure allows saving about 16-17% of energy with respect to the fixed $\{C_1, P_0\}$ configuration. Regarding network performance, Fig. 8 shows that the latency constraint is fully satisfied in all the time windows. Moreover, we reported in Fig. 8 also the performance of the $\{C_2, P_3\}$ configuration, which, contrarily to the optimization procedure results, overcomes the constraint in several cases. The measured values of packet loss probability confirm the fulfillment of network performance constraints.

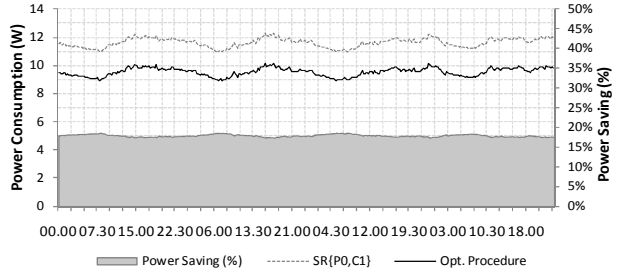

Fig. 7. SR energy consumption values with both the optimization procedure and the fixed pair $\{C_1, P_0\}$. The energy savings of the optimization procedure with respect to the $\{C_1, P_0\}$ pair are also reported.

VII. Conclusions

In this contribution, we focused on performance-adaptive network devices, able to save energy by scaling their traffic processing capacities through AR and LPI mechanisms. 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 validation results were performed by using a Linux-based SR with AR and LPI primitives and real-world traffic traces, and demonstrate how the proposed model can effectively represent energy- and network-aware performance indexes. Moreover, also an optimization procedure based on the model has been proposed and experimentally evaluated. The results show that such procedure can allow saving more than 16-17% of energy with respect to a device with only LPI capabilities enabled.

Acknowledgments

This work has been supported by the ECONET (low Energy Consumption NETworks) project, funded by the European Commission under the 7th Framework Programme (FP7) [23].

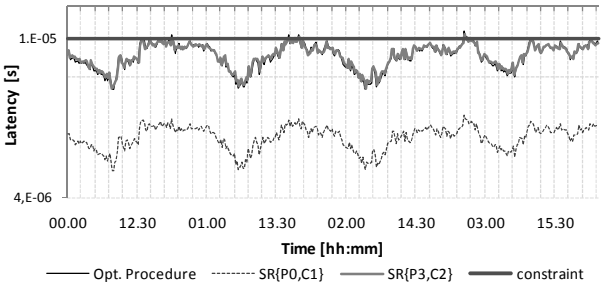

Fig. 8. SR loss probability values measured in each time window with both the optimization procedure and the fixed pair $\{C_1, P_0\}$.

References

[1] Bianco C., Cucchietti F. and Griffa G., "Energy Consumption Trends in the Next Generation Access Network — A Telco Perspective", Proc. of the 29th Int. Telecom. Energy Conf. 2007 (INTELEC 2007), Rome, Italy, Sept. 2007, pp.737-742.

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

[3] 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.

[4] 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..

[5] 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).

[6] 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.

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

[8] 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.

[9] Bolla R., Bruschi R. and Davoli F., "Energy-Aware Performance Optimization for Next-Generation Green Network Equipment", Proc. of the 2nd Workshop on Programmable Routers for Extensible Services of Tomorrow, in conjuction with ACM SIGCOMM'09 (PRESTO09), Barcelona, Spain, Aug. 2009.

[10] 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.

[11] Willinger W., Paxson V. and Taqqu, M., "Self-similarity and Heavy Tails: Structural Modeling of Network Traffic", in "A Practical Guide to Heavy Tails", Chapman & Hall, New York, 1998.

[12] 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.

[13] 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.

[14] 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.

[15] Tijms H. C., "The M^X/G/1 Queue" in "A First Course in Stochastic Models", John Wiley & Sons Ltd, Chichester, UK, 2003, pp. 360.

[16] Doshi B. T., "A Note on Stochastic Decomposition in a GI/G/1 Queue with Vacations or Setup Times", J. of Applied Probability, vol. 22, 419-428, 1985.

[17] Kim H. S. and Shroff N.B., "Loss Probability Calculations and Asymptotic Analysis for Finite Buffer Multiplexers", IEEE/ACM Trans. on Networking, vol. 9, no. 6, pp. 755-768. Dec. 2001.

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

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

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

[21] The Ixia IxN2X Router Tester, http://www.ixiacom.com/products/ixn2x/index.php.

[22] Agilent U2353A multifunction DAQ, http://cp.literature.agilent.com/litweb/pdf/5989-9923EN.pdf.

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