Energy-Aware Performance Optimization for Next-Generation Green Network Equipment

Raffaele Bolla, Roberto Bruschi, Franco Davoli, Andrea Ranieri
DIST, University of Genoa
Via all'Opera Pia 13,
16139, Genoa, Italy

{raffaele.bolla, roberto.bruschi, franco.davoli, andrea.ranieri}@unige.it

ABSTRACT

Besides a more widespread sensitivity to ecological issues, the interest in energy-efficient network technologies springs from heavy and critical economical needs, since both energy cost and network electrical requirements show a continuous growth, with an alarming trend over the past years. In this contribution, we explore and try to evaluate the feasibility and the impact of power management policies able to well suit a heterogeneous set of highly modular architectures, generally used for developing today's network equipment. The proposed policies aim at optimizing the power consumption of each device component with respect to its expected network performance. Finally, in order to provide an experimental evaluation of the proposed ideas, we applied such power management policies to a new generation SW router platform, and we evaluated it with real traffic traces.

Categories and Subject Descriptors

C.2.6 [**Routers**]: Energy-aware Routers

General Terms

Design, Experimentation, Measurement, Performance.

Keywords

Green networks, Router Power Management, SW Router.

1. INTRODUCTION

Today, energy efficiency can be considered as one of the biggest challenges in a large part of industrial and research fields. This arises from the need of reducing the operating, manufacturing and energy related expenses of enterprises, industries as well as residential buildings, while keeping an eye on targets for the reduction of greenhouse gas emissions.

Recent and official studies estimated that ICT industry accounts for approximately 2% of global CO2 emissions, overcoming even the carbon footprint of aviation. In detail, focusing on telecommunication networks, they are estimated to produce about 0.6% of the global CO2 emissions. Today, fixed and mobile network infrastructures have enormous and heavily increasing requirements in terms of electrical energy. For example, as shown in [1] and in [2], energy consumption of the Telecom Italia network in 2006 has reached more than 2TWh (about 1% of the total Italian energy demand), increasing by 7.95% with respect to 2005, and by 12.08% to 2004. Another explanatory example is represented by British Telecom, which absorbed about 0.7% of the total UK's energy consumption in the winter of 2007, making it the biggest single power consumer in the nation [3]. As outlined in [4], similar trends can be generalized to a large part of the other telecoms and service providers, since they essentially depend on data traffic volume increase, which appears to follow the Moore's law, and new services being offered. To support new generation network infrastructures and related services for a rapidly increasing customer population, telecoms and service providers need an ever larger number of devices, with sophisticated architectures able to perform more and more complex operations in a scalable way.

Besides a more widespread sensitivity to ecological issues, the interest on energy efficient networking springs from heavy and critical economical needs, since both energy cost and network electrical requirements show a continuous growth, with an alarming trend over the past years. Networks, links and devices are provisioned for busy or rush hour load, which typically exceeds their average utilization by a wide margin. While this margin is generally reached rarely and over short time periods [5], the overall power consumption in today's networks remains more or less constant with respect to different traffic utilization levels. Against such flat energy wastes, the specific challenge for telecoms, network equipment manufacturers and the networking research community nowadays mainly regards the introduction of innovative criteria and technologies, able to save energy by dynamically adapting network capacities and resources to current traffic loads and requirements. Despite some interesting scientific contributions (e.g., [6], [7], [8] and [9] among others), green networking performance and optimization remains an open and very interesting issue.

Our ideas are based on the introduction, the exploitation and the control of power management capabilities (i.e., sleeping and rate adaptation) inside architectures and components of network equipment. In this respect, our approach starts by considering the two main kinds of power management hardware support, today available in the largest part of COTS processors and under rapid development in other hardware technologies (e.g., network processors, ASIC [9] and FPGA). These power management technologies respectively allow to minimize power consumption when no activities are performed (namely, "idle" optimizations), and to modify the trade-off between performance and energy when the hardware is active and performing operations (namely, "power state" optimizations). These kinds of power management support are generally realized at the hardware layer by powering

PRESTO'09, August 21, 2009, Barcelona, Spain.
Copyright 2009 ACM 978-1-60558-446-1/09/08...$10.00.

off sub-components, or by changing the silicon operating frequency and voltage. Specific control applications, namely governors, are needed to dynamically configure such power profiles through the ACPI standard interfaces. In more detail, the specific objective of such SW governors is to optimize the configuration of network devices, in terms of operating frequency and voltage, with respect to their expected performance.

In [10] and in [11], we already evaluated and modeled the impact of power management capabilities on network performance of new generation Linux SW router platforms, founded on COTS multi-core processors and virtual I/O network interfaces [12]. The obtained results clearly highlight that power management mechanisms introduce a nearly linear trade-off between maximum forwarding performance (in terms of number of forwarded packets per second) and power consumption according to CPU/Cores frequency scaling.

In this paper, we want to move significantly forwards these concepts by showing how these capabilities can be used and dynamically controlled, and by exploring how equipment architectural modularity can be exploited for boosting energy efficiency in next generation network devices.

In more detail, we extend and refine the optimization policy proposed in [13], by including it in a power management governor for SR. This policy provides an explanatory example on how network equipment modularity can be exploited for accurately and separately tuning the tradeoff between power consumption and forwarding. The proposed policy is founded on a simple analytical framework able to optimize power consumption of a network device with respect to its expected forwarding performance. Even if the framework is designed for well suiting a heterogeneous set of distributed architectures (ranging from crossbar and line-cards based devices to multi-core SRs), we will mainly focus on SRs. This is because they already include advanced power management support, and they can be easily modified (i.e., open source SW).

The paper is organized as follows. The optimization policy to be used in the power management governor is introduced in Section 2, while Sections 3 and 4 show the applied analytical model and the optimization procedure, respectively. Section 5 shows the prototypic architecture of the proposed energy governor. In section 6, a performance evaluation is reported. Conclusions are in Section 7.

2. THE POWER MANAGEMENT POLICY

We assume that the distributed router is composed by C hardware components, each one supposed to manage a certain share of the overall forwarded traffic. Moreover, components are assumed to independently switch on different P-states, which provide a different tradeoff between performance and power consumption for each HW element. Finally, every element is obviously supposed to increase its maximum forwarding capacity, as a more power intensive state is selected.

The main objective of the optimization policy proposed here, is to minimize the power consumption of a network device, while maintaining a certain performance level. In this respect, the methodology to be used implies the definition of a suitable cost function, to capture the tradeoff between performance and power consumption, which must then be minimized with respect to the operating parameters. Since the time scales at which the hardware can be switched among different clock frequencies are typically longer than those at the packet- and flow-level, the optimization cannot be realized as a closed-loop control with tight timing constraints. Therefore, we will treat the problem as a parameter-adaptive optimization one, where the expected value of the cost function is periodically minimized over a finite horizon, on the basis of updated information on average values of traffic volumes and requirements.

Starting from these considerations, we suppose to divide a day in different time slices, during which the link traffic loads have an almost similar statistical behavior. During each time slice, the power management governor is thought *i*) to estimate the statistical features of incoming traffic for each device component (e.g., by using the data collected in the same time slice of previous days), and, then, *ii*) to adopt a suitable power configuration for each device element to optimize its trade-off between the expected forwarding performance and energy waste. In this way, the overall optimal configuration of the modular device is achieved by optimally and individually setting the working frequency values of each HW component $f_c \in F_c$ (where F_c is the set of admissible working frequencies for the element c).

We introduce a cost function Φ that represents the overall power consumption of the distributed router, and that can simply be expressed as the sum of power consumptions of the individual elements.

$$\Phi(f_1, \dots, f_C) = \sum_{c=1}^{C} \Phi_c(f_c) \qquad (1)$$

We consider Φ as the cost function of our optimization problem, and we minimize its values in a constrained domain, where the minimum performance bounds we want to assure to forwarded traffic are satisfied. We fix a single set of performance constraints regarding the maximum values of packet loss rates p_c^* for each router component. Given the distributed equipment architecture, these values can be easily used to determine the loss probability of forwarded traffic flows (as will be shown in sub-section 3.3). Thus, we can formulate our optimization problem as follows:

$$\begin{cases} \min_{f_1,\dots,f_C} \Phi(f_1, \dots, f_C) \\ p_c(f_c) \le p_c^* \;\; \forall c \in [1, C] \end{cases} \qquad (2)$$

In order to find the optimal router configuration, we have to find the frequency array $\{\hat{f}_1, \dots, \hat{f}_C\}$, which guarantees the minimum value of Φ that respects the performance bounds.

3. THE ANALYTICAL MODEL

Our approach does not aim to describe equipment architecture and performance in detail, but to provide a generic analytical framework, which well suits in a large set of equipment/ networking scenarios.

Starting from these considerations, we provide an overall model composed by two simple sub-models that represent traffic and single components' behavior, respectively.

The traffic sub-model represents the traffic offered to the equipment in terms only of average traffic matrix on a per port basis, related variance and maximum deviation. The single components' model takes into account how the forwarding/processing capacity of a component scales according to the working frequencies and traffic loads. Obviously, to correctly apply the overall optimization policy, the equipment architecture and the components' interaction must be given.

3.1 Traffic model

Our approach consists of dividing a day in different time slices, in order to collect and to separately estimate traffic loads, and to calculate an optimized router configuration for each one of such time periods. Such approach allows working on data that can be easily provided to routers, and it does not require high computational capacity or memory usage. Moreover, owing to the 24 hours' time scale variability of traffic load dynamics on Internet links, in many cases rush hours and low utilization time bands can be easily identified. For the sake of simplicity and without loss of generality, we drop the time slice index in the rest of this paper.

In detail, we assume to collect periodical samples, for each network component, inside the same time slice of the instantaneous traffic offered load $\lambda_c^{(i)}(t)$ on each network interface $i \in I_c$. Then, such offered load samples are used to estimate three main statistical parameters, namely: λ_c, the average value of traffic offered load; σ_c^2 , the variance of traffic offered load; π_c , the peak value of traffic offered load.

3.2 Equipment component model

Since our main aim is to provide a high level model for network equipment components supporting power saving mechanisms, we only model some basic aspects, which can easily be adapted to different component HW technologies and architectures.

As previously sketched, we assume that each equipment component c can work at different internal clock frequencies $f_c \in F_c$, where the set F_c includes a limited number of frequency values at which component HW circuits can correctly work[1]. Each component is supposed to rise its packet processing capacity and its power consumption, as the working frequency value increases. Moreover, we have also to take into account active stand-by optimizations, which allow reduce power consumption when no activities or operations are performed. Therefore, we completely characterize the router element c by considering the following three parameters:

- $\mu_c(f_c)$: the maximum service rate when working at f_c;
- $\Phi_{idle}(f_c)$: average power consumption when no activity is performed inside the router element working at f_c;
- $\Phi_{active}(f_c)$: average power consumption when the router element c performs operation at the clock frequency f_c.

It is reasonable to suppose that both the set of values of working clock frequencies F_c and the related μ_c, Φ_{idle}, and Φ_{active} parameters could be directly provided by the manufacturers in the component's datasheet. However, even if their specific values depend on the component HW, μ_c, Φ_{idle}, and Φ_{active} are supposed to be monotonic increasing functions with respect to the frequency f_c. The limit scenarios, where Φ_{idle} or Φ_{active} are constant according to f_c, correspond to the lack of "idle" or "power state" management mechanisms, respectively. Starting from the previous definitions, we can write the average power consumption Φ_c of a network equipment component, working at the frequency f_c, as the weighted sum of the Φ_{idle}, and the Φ_{active} consumptions:

$$\Phi_c(f_c) = p_{idle}\Phi_{idle}(f_c) + (1 - p_{idle})\Phi_{active}(f_c) \qquad (3)$$

The "idle probability" p_{idle} corresponds to the probability that the component's packet processing unit is not running any forwarding operations, and $1 - p_{idle}$ is the probability that such operations are being performed. The idle probability p_{idle} can be related to the ratio between the maximum packet processing capacity and the traffic offered load, in terms of packets per second that require header processing. In particular, by assuming that the packet processing unit in a component can be modeled as a single server[2] queuing system, with finite buffer and generic arrival/service time distributions (i.e., a G/G/1/N queuing model), we can express its idle probability as follows:

$$p_{idle} = 1 - \frac{\lambda_c[1-p_c(f_c)]}{\mu_c(f_c)} \qquad (4)$$

Since we cannot obtain a closed-form expression for the loss probability of a generic queuing system, we provided in [13] the upper and the lower bound approximations, namely $p_c^{max}(f_c)$ and $p_c^{min}(f_c)$, respectively. In more detail:

$$p_c^{min}(f_c) = \max\left\{0, \frac{\lambda_c - \mu_c(f_c)}{\lambda_c}\right\} \qquad (5)$$

$$p_c^{max}(f_c) = e^{-\frac{[\mu_c(f_c)-\lambda_c]^2}{2\sigma_c^2+\frac{2}{3}M_c[\mu_c(f_c)-\lambda_c]}} \qquad (6)$$

Where $M_c = \max_{\forall i \in I_c}\left\{\pi_c^{(i)} - \lambda_c^{(i)}\right\}$.

While the p_c^{max} parameter is useful to estimate the performance constraint in Eq. 2, the p_c^{min} one can be used to find an upper bound for the average power consumption Φ_c. In particular, starting from Eq. 4, we can determine the following upper bound for the idle probability:

$$p_{idle} \geq 1 - \frac{\lambda_c[1-p_c^{min}]}{\mu_c(f_c)} \qquad (7)$$

and, consequently, also for the power consumption $\Phi_c(f_c) \leq \widetilde{\Phi}_c(f_c)$:

$$\widetilde{\Phi}_c(f_c) = \Phi_{idle}(f_c) + \frac{\lambda_c[1-p_c^{min}]}{\mu_c(f_c)}[\Phi_{active}(f_c) - \Phi_{idle}(f_c)] (8)$$

Finally, in order to solve the optimization problem defined in Section 2, we can approximate the $\Phi_c(f_c)$ and the $p_c(f_c)$ parameters with the relative upper-bounds $\widetilde{\Phi}_c(f_c)$ and $p_c^{max}(f_c)$.

3.3 The equipment architecture

In order to correctly apply the optimization policy for each component, we have to know, and to estimate the statistical features (in terms of λ_c, M_c and σ_c^2) of the traffic share incoming to that element. The overall device architecture and how components exchange traffic among themselves must be known. This because if the device's data plane is composed by one or more components working in series, the loss probability of the first components in the chain may affect the λ_c of the last ones.

This is certainly the case of architecture based on switching matrix and line-cards (usually adopted by high-end commercial equipment). In fact, each line-card can be thought at least as a single device component, and the forwarding process for a single packet usually involves two line-cards (i.e., the reception and the transmission ones). In such scenario, the solution to be used in order to correctly evaluate the λ_c parameters consists of a simple recursive optimization procedure, which is diffusely adopted in

[1] The HW available frequency values are generally a certain multiple of a "base frequency".

[2] It possible to extend the p_{idle} definition to multi-server queueing system case, without the proposed model losing its generality.

"multi-stage" architectures. For this reason and for the paper's space limit, we do not deepen such procedure here.

New generation SW routers, founded on multi-core processors and COTS hardware [12], deploy a rather different architecture with respect to their commercial cousins. In more detail, as outlined in [10], [11] and [12], each Core included in a SW router can be considered as an independent component that entirely process a certain share of the incoming traffic (i.e., the traffic incoming from the network interfaces (NICs) bounded to that Core).

In this respect, a SW router can be thought like a set of Cores working in a parallel and fully independent manner. In such case, the λ_c of a component will not be affected by other ones, and no recursive optimization procedure is needed. Therefore, we can decompose our optimization problem on a per-component basis:

$$\min_{f_1,\dots,f_C} \Phi(f_1, \dots, f_C) = \textstyle\sum_{c=1}^{C} \min_{f_c} \Phi_c(f_c) \tag{9}$$

Beyond Cores' independence, new generation SR routers can exploit a further freedom of degree by dynamically change the Cores-NICs bindings. In more detail, as outlined in [12] and in Fig. 1, this gives the opportunity of re-distributing among the SR Cores the traffic offered load, incoming from different NICs, and to consequently find the most energy-aware SR configuration (e.g., using a limited number of Cores at maximum speed, or a larger number of Cores at lower speed). In this respect, [10] and [11] report an energy-aware performance analysis of SRs based on different HW platforms, and show that the minimum power consumptions are generally related to the configurations where a larger number of Cores is active at lower speeds.

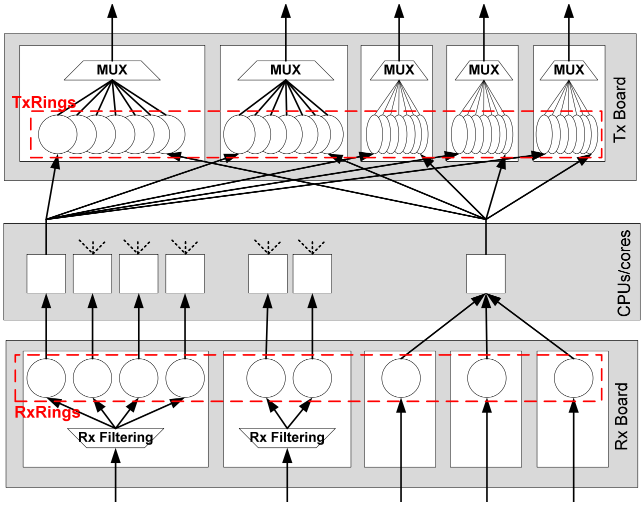

Figure 1. Overview of the Multi-Core SR architecture with multi-queueing NICs as introduced in [12].

Starting from these considerations, we extended the original optimization mechanism in [13] in order to analyze which SR configuration (in terms of Core-NIC bindings) is the most energy-aware one. To this purpose, we perform an exhaustive search on a subset of all the possible Core-NIC bindings[3]. For each analyzed configuration, we extract the λ_{c}, σ_c^2 , π_{c} parameters from the statistical features of traffic load incoming in the bounded NICs.

3.4 Closed-loop enhancements

A further enhancement to the original optimization procedure in [13] consists of the addition of a closed-loop control, which can be exploited for raising the components' operating frequencies in case of an unexpected increase of incoming traffic loads happens. In more detail, this enhancement is simply realized by enabling the Traffic model (see sub-section 3.1 and section 5) to collect data about packet losses at each device component. If the loss rate of one or more components exceeds the p_c^* for a certain time period, the optimization framework is immediately invoked, and a new and more conservative re-configuration of operating frequencies performed.

4. MINIMIZATION PROCEDURE

For both distributed architectures introduced in sub-section 3.3, the core step of the procedure is the power consumption minimization of each single component. Thus, in order to find the optimal component configuration, we select the sub-set $\tilde{F}_c$ of working frequencies f_c that respect the performance constraint in Eq. 2. To determine $\tilde{F}_c$, we exploit Eq. 12 as follows:

$$\tilde{F}_c = \{f_c \in F_c \colon\ p_c^{max}(f_c) < p_c^*\} \tag{10}$$

Thus, for $\forall f_c \in \tilde{F}_c$, we numerically evaluate Eq. 14 in order to select the frequency f'_c, which guarantees the minimum value of $\widetilde{\Phi}_c(f_c)$. The set of analyzed frequency values is reduced by evaluating the loss probability constraint. In more detail, by inverting Eq. 12, and by substituting the p_c^{max} with the loss constraint p_c^*, we can obtain the minimum capacity μ_c^* that assures the constraint fulfillment:

$$\mu_c^* = \lambda_c - \tfrac{1}{3} M_c \ln(p_c^*) + \tfrac{1}{3}\sqrt{{M_c}^2 \ln^2(p_c^*) - 18\sigma_c^2} \tag{11}$$

Thus, we consider only the f_c values, for which $\mu_c(f_c) \geq \mu_c^*$.

In order to solve the minimization problem and to obtain the minimum $\widetilde{\Phi}_c(f_c)$ value, we use a "brute force" approach (exhaustive search) on the subset of frequency values satisfying the loss probability constraint. However, since the number of working frequencies $|F_c|$ is generally very low (i.e., it does not exceed 10 values in the largest part of HW components and technologies), and since the proposed model is characterized by a very low computational complexity, finding the minimum is feasible even by performing an exhaustive search.

5. THE PROTOTYPIC GOVERNOR

In order to validate and evaluate the performance of the proposed energy-aware optimization framework, we decided to implement a prototypic application, namely "governor" for dynamically managing the trade-off between network performance and power consumption. This governor is basically thought to work on a Linux Multi-Core SW Router [12][14], since it is an open source platform, and already includes HW power management capabilities. As shown in Fig. 2, the governor is composed by three main modules, namely Traffic Estimator, Optimization Framework and ACPI sockets, respectively.

The Traffic Estimator is responsible for the continuous monitoring and sampling of equipment registers regarding the current traffic load (e.g., rates of received packets, loss rates, etc.). As shown in the sub-section 3.1, this modules processes the collected information for evaluating the λ_c, σ_c^2 , π_{c} for each time slice in a day. The average sampling time of traffic load registers is 30 sec, while the values of λ_{i}, σ_i^2 , π_{i} are recomputed every 10 minutes. These times are obviously re-configurable.

[1] [3] this subset corresponds to binding guaranteeing that Core has a forwarding capacity larger than $\lambda_{\mathrm{c}} + \sigma_c^2$

The traffic load estimation are used to feed the Optimization Framework module, where the analytical models described in sub-sections 3.2 and 3.3 are applied in order to find the optimal energy aware device configuration.

In more detail, the Optimization Framework is fundamentally composed by two sub-modules, namely “Equipment” and Component “models”, which are iteratively used for obtaining the optimized device configuration. Starting from the traffic offered load estimation, the Equipment model is responsible for deciding candidate values of clock frequency f_c and estimated traffic parameters (e.g., incoming traffic for different NIC-Core bindings) for every SR Core.

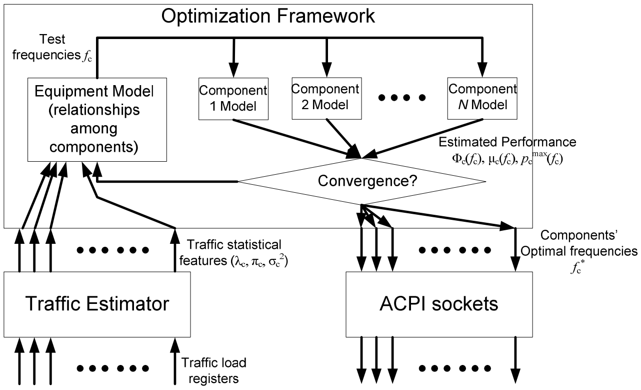
Optimization Framework
Test frequencies f_c
Equipment Model (relationships among components)
Component 1 Model
Component 2 Model
Component N Model
Estimated Performance $\Phi_c(f_c)$, $\mu_c(f_c)$, $p_c^{max}(f_c)$
Convergence?
Traffic statistical features (λ_c, π_c, σ_c^2)
Components' Optimal frequencies f_c^*
Traffic Estimator
ACPI sockets
Traffic load registers

Figure 2. Prototypic governor scheme.

Then, starting from the previous candidate values, C different instances of component model are used in order to estimate the average power consumption (Eq. 9) and the maximum loss probability (Eq. 7) for each device component.

In this way and by exploiting the minimization criterion in section 4, for all the possible bindings between Cores and NICs, the Equipment model finds, the frequency values $f_1, \dots, f_C$, which minimizes the device’s power consumption while respecting the performance constraints. The final optimal frequencies will correspond to the ones providing the lowest estimated energy consumption.

Finally, such frequency values are effectively transmitted to the HW by the ACPI sockets’ module.

6. PERFORMANCE EVALUATION

In order to evaluate the proposed optimization mechanism, we decided to use the multi-Core SW router architecture already adopted in [12].

In detail, the used HW platform includes two dual core Xeon processors. The choice of validating the proposed optimization policy on a SW router platform was simply driven by the fact that only such king of HW platforms already include power management capability. Each core can independently work at four operating frequencies, namely 3.0, 2.667, 2.333 and 2.0 GHz. Table I shows Φ_{idle}, Φ_{active} and μ_c values at different available f_c referred to a single Core the selected HW platform, while Fig. 3 summarizes the forwarding performance and power requirements for all the possible combinations of Cores’ frequency values. The SW router includes 4 Gigabit Ethernet interfaces. Each Core can dynamically be bound to each Gigabit port.

Regarding the traffic load generation, we used the Agilent N2X router tester to emulate the traffic offered to each ingress link. In order to thoroughly emulate such offered load, we exploited statistical features (in terms of burstiness and inter-burst periods at the packet level) collected from real traffic traces, captured at the packet level by monitoring the GARR network [15] during 30 day periods.

Table I. Power consumptions and service rate per core and f_c per Core

f_c [MHz]	Φ_{idle} [W]	Φ_{active} [W]	μ_c [kpps]
3000	20.5	45.75	985
2667	18	40	890
2333	15.75	33.75	770
2000	13.5	29	670

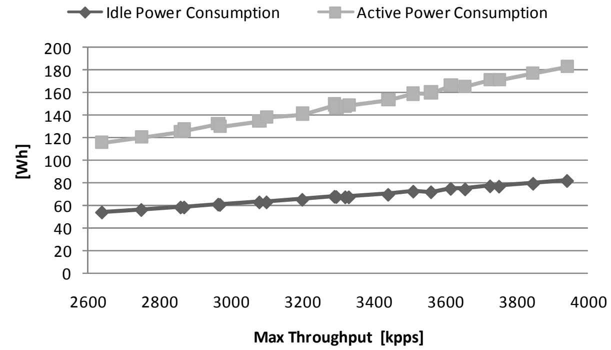

Figure 3. Maximum forwarding capacity, idle and active power consumptions for all the possible combination of Cores’ frequency values.

The optimization framework works on 144 daily time slices, each one with duration equal to 10 minutes. Fig. 4 shows the average values, the deviation, of overall traffic load offered to 4 Gigabit links of SR. The loss probability constraint was fixed at 0.01%.

Fig. 4 also shows the maximum forwarding capacity of the configurations, which are dynamically optimized by the proposed framework every 10 minutes. Such values were measured by emulating the SW router behavior for 10 days.

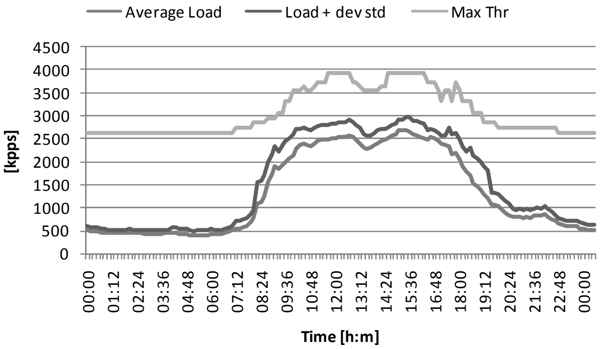

Figure 4. Offered load daily statistics (in terms of average value and deviation), and maximum forwarding capacity at the “optimal” frequencies.

Fig. 5 reports the measured power consumption of all the 4 Cores[4] estimated power consumptions of a single core at different operating frequencies in all the time slices, and the consumption values when no optimization frameworks are adopted. In more detail, we compared the energy requirements of our proposal with a SR that includes only the ACPI idle optimization, and runs all the Cores at the 3.0GHz operating frequency.

Thus, Fig. 5 outlines that the proposed framework allows to effectively reduce the overall power wasting of the SR: on a per day basis, we obtain an overall power consumption equal to 13.77 kW, while a SR with only idle optimization wastes about 16.69 KW. Thus, the resulting energy saving is about 30% with respect to the same HW platform with only the idle optimization.

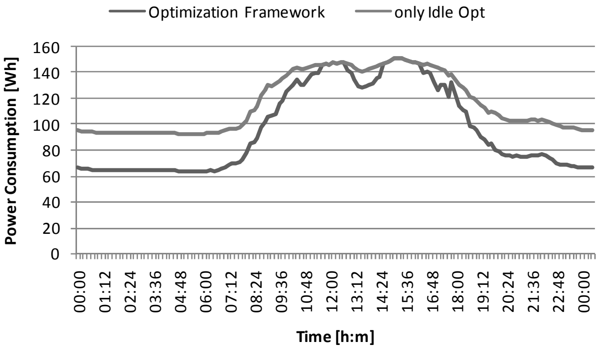

Figure 5. Average power consumption of the proposed optimization framework vs SR energy-aware performance with only the idle power optimization.

7. CONCLUSIONS

In this contribution, we demonstrated the feasibility and showed an explanatory example of green equipment for next generation networks. In more detail, we extended and evaluated the governor policy introduced in [13], which can effectively be adopted to dynamically optimize power consumption of a modular network device with respect to its expected forwarding performance. In addition, we showed how these optimization capabilities can be introduced in a SR, and how a governor can effectively developed.

Moreover, we showed that the proposed approach suits properly different equipment architectures (i.e., multi-core SW routers and crossbar-based devices). However, we mainly focused on SR architectures, since they already include power management capabilities, and they can be easily modified (thanks to the availability of source code).

The benchmarking results, obtained with a multi-core COTS SW router and real link traffic statistics, point out that the proposed optimization mechanism provides interesting levels of power saving (about 30% with respect to the use of idle states only).

8. REFERENCES

[1] C. Bianco, F. Cucchietti, G. Griffa, "Energy consumption trends in the next generation access network - a telco perspective," *Proc. 29th Int. Telecomm. Energy Conf (INTELEC 2007)*, Rome, Italy, Sept. 2007, pp.737-742.

[2] Telecom Italia Website, "The Environment", URL: http://www.telecomitalia.it/sostenibilita2006/English/B05.html

[3] BT Press, "BT announces major wind power plans," Oct. 2007, http://www.btplc.com/News/Articles/Showarticle.cfm?ArticleID=dd615e9c-71ad-4daa-951a-55651baae5bb.

[4] 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.

[5] Sprint IP Monitoring Project. URL: http://ipmon.sprint.com/.

[6] Noguera, J.; Kennedy, I.O., "Power Reduction in Network Equipment through Adaptive Partial Reconfiguration," *Proc. of the 2007 Int. Conf. on Field Programmable Logic and Applications (FPL 2007)*, Aug. 2007, pp. 240-245.

[7] Gupta, M.; Singh, S.; "Using Low-Power Modes for Energy Conservation in Ethernet LANs," *Proc. of the 26th Annual IEEE Conf. on Computer Communications (IEEE INFOCOM 2007)*, Anchorage, Alaska, USA, May 2007.

[8] Nedevschi, S.; Popa, L.; Iannaccone, G.; Wetherall, D.; Ratnasamy, S.; "Reducing network energy consumption via sleeping and rate-adaptation," *Proc. 5th USENIX Symp. on Networked Systems Design and Implementation*, San Francisco, CA, 2008, pp. 323-336.

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

[10] R. Bolla, R. Bruschi, A. Ranieri, "Green Support for PC-based Software Router: Performance Evaluation and Modeling", Proc. *2009 IEEE International Conference on Communications* (ICC 2009), Dresden, Germany, June 2009, to appear.

[11] R. Bolla, R. Bruschi, A. Ranieri, "Performance and Power Consumption Modeling for Green COTS Software Router", *Proc. 1st International Conference on COMmunication Systems and NETworkS* (COMSNETS 2009), Bangalore, India, Jan. 2009.

[12] Bolla, R.; Bruschi, R., "PC-based Software Routers: High Performance and Application Service Support," *Proc. of the ACM Sigcomm Workshop on Programmable Routers for Extensible Services of Tomorrow* (PRESTO'08), Seattle, WA, USA, Aug. 2008, pp. 27-32.

[13] Bolla, R.; Bruschi, R.; Davoli, F.; Ranieri, A.; "Performance Constrained Power Consumption Optimization in Distributed Network Equipment", *Proc. of the First Int. Workshop on Green Communications* (GreenComm09) co-located with IEEE ICC 2009, Dresden, Germany, June 2009, to appear.

[14] K. Argyraki, S. A. Baset, B. G. Chun, K. Fall, G. Iannaccone, A. Knies, E. Kohler, M. Manesh, S. Nedveschi, S. Ratnasamy, "Can Software Router Scale?," *Proc. of the ACM Sigcomm Workshop on Programmable Routers for Extensible Services of Tomorrow* (PRESTO'08), Seattle, WA, USA, Aug. 2008, pp. 21-26.

[15] The GARR NoC, https://www.noc.garr.it/GINS/svg/weathermap.php

[2] Note that we obtained the Cores' power consumption by measuring the power consumption of the entire SR, and subtracting the average consumptions of mainboard and NICs.