Theoretical and Technological Limitations of Power Scaling in Network Devices

Raffaele Bolla[(a)], Roberto Bruschi[(b)], Alessandro Carrega[(a)] and Franco Davoli[(a)]

[(a)] Department of Communications, Computer and Systems Science (DIST)
University of Genoa
Via Opera Pia 13, 16145 Genova, Italy
e-mail: {raffaele.bolla, alessandro.carrega, franco.davoli}@unige.it

[(b)] National Inter-University Consortium for Telecommunications (CNIT)
Research Unit of Genoa
Via Opera Pia 13, 16145 Genova, Italy
e-mail: roberto.bruschi@cnit.it

Abstract — ***The largest part of routers and switches, today deployed in production networks, has very limited energy saving capabilities, and substantially requires the same amount of energy both when working at full speed or when being idle. In order to dynamically adapt such energy requirements to the real device work load, current approaches foster the introduction of low power idle and power scaling primitives in entire devices, internal components and network interfaces. Starting from these considerations, we focus on power scaling, and we propose an analysis of the theoretical and technological limitations in adopting such kind of mechanisms. Thus, our contribution is twofold. On one hand, we performed several tests to identify the technological limitations in a software router based on off-the-shelf hardware, which already includes such capabilities. The results achieved show that the power scaling allows a linear trade-off between consumption and network performance, but the time to switch between two power states may cause a non negligible service interruption. On the other hand, regarding the theoretical limitations, we consider the trade-off between the benefit in dynamically adapting the power states within short time-scales and the overhead needed to choose and select the new power state.***

Keywords: Green Networking, Power Management, Dynamic Power Scaling.

I. INTRODUCTION

The energy efficiency issue is assuming ever-greater importance in most industrial sectors and research fields. Several studies suggested that the total energy consumption of electronics in the U.S., in 2006, was more than 70 trillion watt-hours per year (TWh/yr) of electricity, costing billions of dollars, and equivalent to at least 50 million metric tons of carbon dioxide emissions per year [1]. The energy wasted by telecommunication networks represents a non-negligible and continuously increasing share of such consumption.

Triggered by the increase in energy price, the continuous growth of costumer 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 also for wired networks and service infrastructures.

In the last years, a large set of telecoms, ISPs and public organizations around the world reported statistics of network energy requirements and the related carbon footprint, showing an alarming and growing trend. The European Commission DG INFSO report in [2] estimated European telcos and operators to have an overall network energy requirement equal to 14.2 TWh, which will rise to 35.8 TWh in 2020 if no green network technologies will be adopted. As shown in [3] and [4], energy consumption of the Telecom Italia network in 2006 was more than 2 TWh reaching 1% of the total Italian energy demand.

As described in [5] there are two main motivations to adopt "green" networking: an environmental one, for the reduction of wastes and relative CO_2 emissions; and an economic one, related to the reduction of costs sustained by operators to keep the network on and maintain the desired level of quality of service.

To this purpose, telecoms and service providers have begun requiring disruptive architectural solutions, protocols and innovative equipment in order to allow a better ratio of performance to energy consumption. This has inspired major ICT (Information and Communication Technologies) companies and research bodies to undertake different private initiatives focused on developing more sustainable data centres and network infrastructures.

Nevertheless, at present, current network devices and infrastructures do not implement power saving mechanisms in order to increase the energy efficiency. Networks and devices waste a huge quantity of energy, since they lack of any energy aware optimization. Indeed, they consume the same amount of energy when running at peak performance or when being idle.

The most important approaches to energy efficiency optimization in networking devices can be classified in two different categories: (*i*) minimizing the power consumption when no activities are performed (namely "idle" optimizations), and (*ii*) modifying the trade-off between network performance and energy when the hardware is active and performing operations (namely, "power state" optimizations). These two main kinds of power management policies are available in the largest part of COTS (Commercial Off-The-Shelf) processors and under rapid development in other hardware technologies (e.g., network processors, Application Specific Integrated Circuits - ASICs [6] and Field Programmable Gate Arrays -FPGAs). The IEEE 802.3az [7] task force considered both these mechanisms, but the current standard version includes only idle optimizations, given the technological and theoretical complexity of power scaling approaches. In

978-1-4244-8172-9/10/$26.00 ©2010 IEEE

fact, (*i*) changing frequency and/or voltage in silicon circuits generally requires a longer time period than entering low power idle states; and (*ii*) the adoption of power scaling mechanisms requires more control logic with respect to idle optimizations (for selecting the optimal power state among the available ones[1]). However, the quest for power scaling mechanisms is still open and many aspects need to be further analyzed. Starting from such considerations, the main objective of the paper is to explore the potential feasibility and the impact of utilizing the power scaling optimization in a network device. Our idea is to evaluate the limitations in adopting power-scaling mechanisms inside architectures and components of network devices. These kinds of power management support are generally realized at the hardware layer by changing the silicon operating frequency and voltage. Specific control applications termed *governors* provide the control logic, and are needed to dynamically configure such power profiles through specific interfaces.

The paper is organized as follows. Section II describes the Software Router (SR) architecture used in our tests. In Section III, we focus on the power scaling mechanism and how it is implemented in standard commodity processors. Section IV shows the numerical results about the relationship between power consumption and performance. The results about the power scaling mechanism are in Section V. In Section VI, the theoretical limitations due to the delay introduced by the frequency selection policy are described. Finally, conclusions and future work are highlighted in Section VII.

II. MULTI-CPU/CORE – MULTI-QUEUE ARCHITECTURE

The modern versions of SRs are founded on multi-core/cpu and COTS hardware and deploy a different architecture with respect to their commercial cousins. Each core included in a SR is an independent component that processes a certain share of incoming traffic. This way, the SR is represented as a set of Cores that work in parallel and independently.

In our tests, in addition to the multi-core processors, we use a specific network adapter that supports multiple Tx-Rings and Rx-Rings per network interface [8]. For example, the Intel PRO 1000 adapter (with MAC chip 82571 and higher) presents these characteristics and has been used in our tests.

As described in [9], the architecture with a multi-core processor and the multi-queue adapter can improve the forwarding performance significantly.

Figure 1 shows the SR architecture that includes these two components and provides:

- a number of CPUs/cores per Rx port that guarantee enough computational capacity to process incoming traffic;
- a number of Tx-Rings per interface equal to the number of CPUs/cores involved in traffic forwarding;
- multiple Rx-Rings per high-speed interface (ideally equal to the number of CPUs/cores needed to achieve the maximum theoretical packet rate).

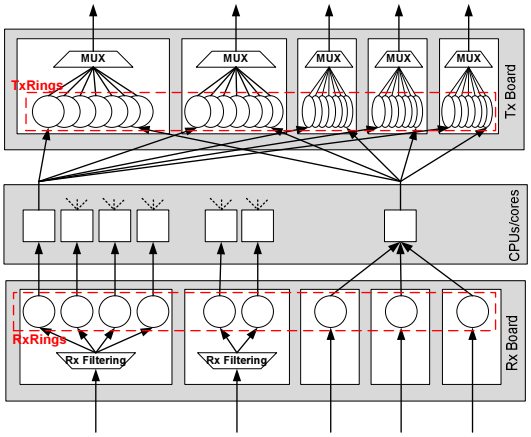

Figure 1. Overview of the SR Architecture.

In [10] and in [11], we already evaluated and modeled the impact of power management capabilities on network performance of new generation Linux SR platforms, founded on COTS multi-core processors and virtual I/O network interfaces [12]. Considering these previous works, and starting from their results, we analyze the impact of power scaling optimization with different tests on general-purpose hardware.

In our tests, we used this architecture with different processors (Xeon, Intel Core 2 and i5, AMD, etc.). The results with the different processors are very similar. In this paper we focus on the Intel Core i5 processor family that implements the latest power-saving mechanisms.

III. POWER SAVING INTERFACE

A PC-based SR generally includes support of the Advanced Configuration and Power Interface (ACPI). It provides a well-known interface to support dialogue between the hardware and the software layers in order to hide the heterogeneous power management mechanisms and details of different processors.

The ACPI standard abstracts the power-specific management capabilities of processors into two main different power saving mechanisms, namely performance and power states (P-States and C-States, respectively). These two mechanisms are essentially the ones described in the introduction, and P- and C-states correspond to idle and power optimization, respectively.

Considering the Intel i5 multi-core processor involved in our experiments, Table I reports the values of power saving and the transition times of the different C-states.

The C_0 power state is an active state where the CPU

[1] In idle optimization, the control logic is very simple, because, contrarily to power scaling, it works at run-time in two states only: "on" and "idle". If more idle states are available, the active one is periodically selected based on latency constraints.

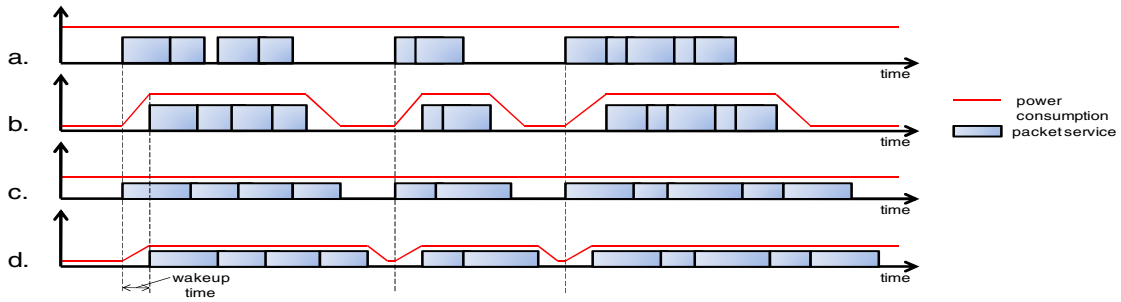

Figure 2. Power consumptions in the following cases: (a) no power-aware optimizations, (b) only idle logic, (c) only performance scaling, (d) performance scaling and idle logic

(central processing unit) executes instructions, while the power states from C_1 to C_n are idle states where the processor consumes less power and dissipates less heat.

Regarding P-states, the ACPI provides an equivalent frequency that represents the computing capacity of the core/processor given the specific P-state. Hence, the equivalent frequency is an abstraction behind which the operating energy is changed by altering the voltage, or throttling the clock. Thus, using P-states, a core can consume different amounts of power while providing different performance at the C_0 (running) state. ACPI allows the tuning of performance through P-states transitions. Unfortunately, due to issues in silicon electrical stability, the transition time between different P-states is generally very slow, especially if compared with usual time scales in network dynamics.

TABLE I. ENERGY SAVING AND TRANSITION TIMES FOR COTS PROCESSORS' C-STATES

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

The largest part of current CPUs can switch their performance state in about 2-3 ms as described in Section V. Due to these large P-state transition times, any closed-loop policies with tight time constraints are not feasible and cannot be adopted for optimizing power consumption inside network device architectures.

Table II shows the equivalent frequencies for the Intel core i5 processor used in our evaluations. These values are reported in the official Intel datasheet [13].

IV. TRADE-OFF BETWEEN PERFORMANCE AND POWER CONSUMPTION

This section describes the results about the relationship between power consumption and performance. In Figure 2 we show the general cases in using these power saving techniques. Reducing the power consumption with performance scaling and idle logic during no-activity periods can be very difficult when these periods are small (Figure 2d).

TABLE II. EQUIVALENT FREQUENCY FOR EVERY P-STATE ON INTEL CORE I5 PROCESSOR

P-State	*Frequency*	*P-State*	*Frequency*
P_{10}	2.67 GHz	P_5	2.00 GHz
P_9	2.54 GHz	P_4	1.73 GHz
P_8	2.40 GHz	P_3	1.60 GHz
P_7	2.27 GHz	P_1	1.47 GHz
P_6	2.14 GHz	P_0	1.20 GHz

The first evaluations regard the performance and power consumption with the specific hardware introduced in Section II. For each equivalent frequency, tests were run by analyzing the performance of forwarding in terms of throughput with their consumption. Figure 3 reports the results of our tests with CBR (Constant Bit-Rate) traffic for each different equivalent frequency. With low offered load there are no differences by changing the frequency. Considering the possible maximum throughput, the difference became purposeful passing 50% of the offered load.

Figure 4 shows the power consumption with the variation of the offered load according to the same scenario of Figure 3. In detail, these results suggest that we can reduce the power consumption by lowering the operating equivalent frequencies. However, we can note how the achievable energy savings mainly depend also on incoming traffic volumes. This is because C-state optimizations were enabled in the i5 processor under test: when the CPU goes idle, it enters the C1 state and saves about 70% of its energy. If the C1 optimization had been disabled, Figure 4 would have shown horizontal lines, representing power requirements of a CPU when active at a certain equivalent frequency.

Figure 4 demonstrates that it is possible to obtain energy savings by using only the C-State. If one wishes to obtain more energy savings, it is necessary to integrate these mechanisms with power optimization.

V. POWER SCALING OPTIMIZATION

As shown in Figures 3 and 4, power optimization can allow obtaining significant power saving without compromising performance, by considering the traffic load

dynamics on the Internet, where low utilization time bands can be easily identified. The main difficulties are how and when changing the equivalent frequency. Since the time required for the frequency scaling is non-zero, performance may be adversely affected if the change is done very frequently.

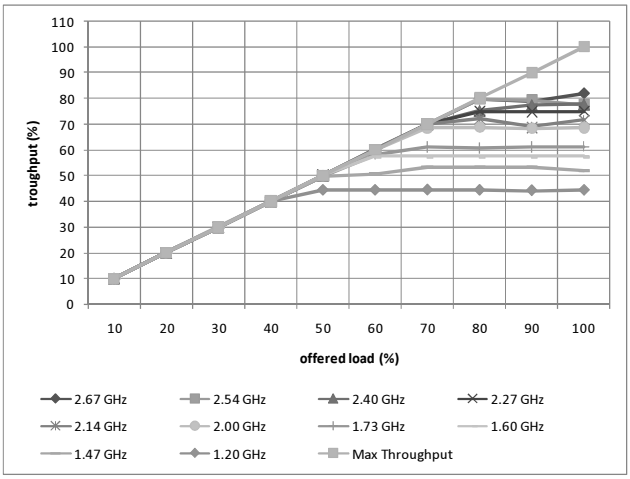

Figure 3. Throughput values according to different equivalent frequencies.

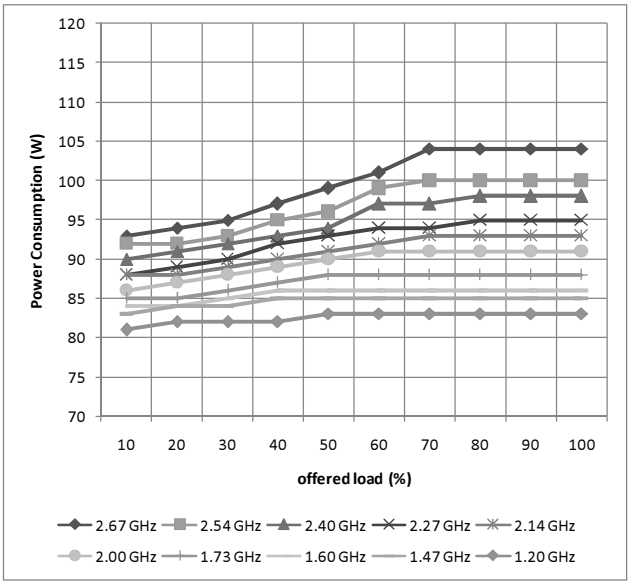

Figure 4. Power consumption values according to different equivalent frequency values.

In particular, the time spent on frequency scaling consists of two parts: (a) hardware time necessary to change the frequency; (b) time necessary to choose the next frequency.

These two delays can affect performance in terms of increased latency, as shown in Figure 5. The results are based on the tests done with the proposed architecture and related to the ones of Figures 3 and 4. It is interesting to observe how the latency changes based on what is the equivalent frequency before the scaling. For example, the result of frequency scaling from 2.00 GHz to 2.40 GHz is higher than the opposite.

Part b) of the time spent during the frequency scaling can be reduced considering the specific architecture described in Section II. The software (SW) governor defined in the ACPI standard executes the frequency selection phase. With the multi-core processor, a specific core is assigned to compute the frequency selection. In this way, as shown in Figure 6, the frequency selection period done by the SW governor does not affect the performance and the service in the forwarding engine is correctly active. It is important to remember that the specific SW governor changes the frequency periodically based on the information of the last periods and we do not focus in this paper on how the governor collects information related to CPU load, network load, etc., to choose the next frequency.

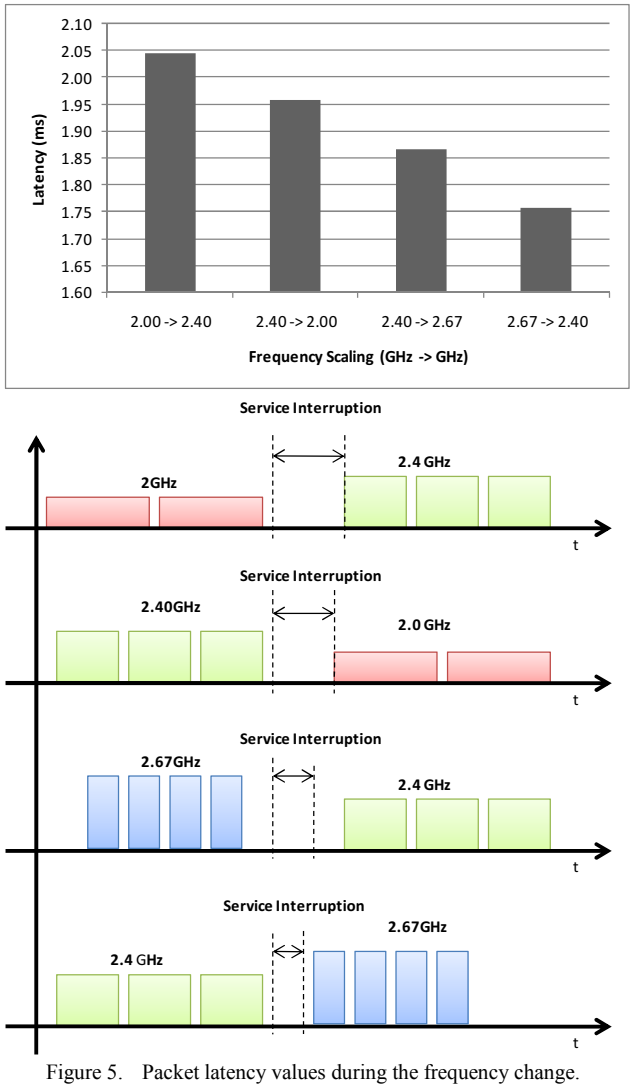

Figure 5. Packet latency values during the frequency change.

The period of frequency setting is done in the forwarding core and it causes the interruption of the service affecting the network performance. It is possible to reduce the period of service interruption with the multi-queuing network interface. This way, during the period of frequency setting for a specific core/processor, the associated queue of the NIC (Network Interface Card) is remapped to a different core/processor. The service interruption caused by the frequency scaling is minimized.

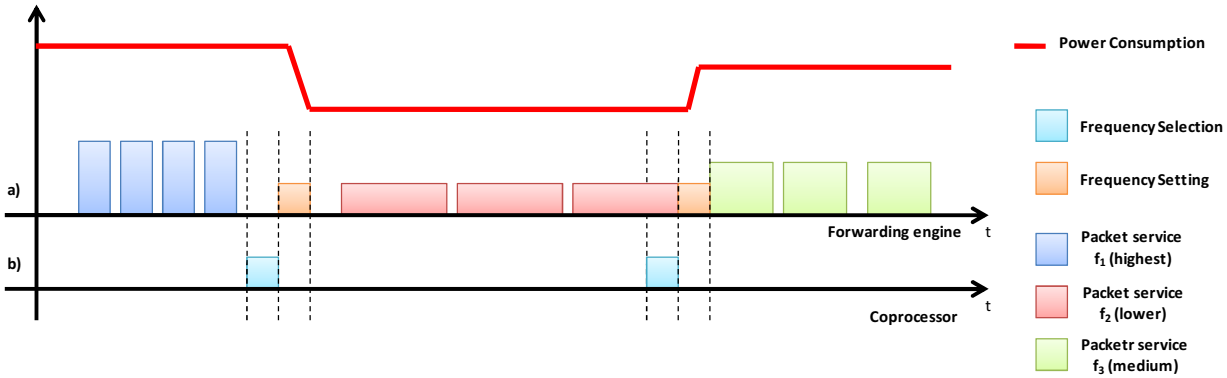

Figure 6. Delay time and Power consumptions when frequency scaling is enabled.

However, with the multi core solution it is necessary to use a specific CPU where to remap the queue, with an increase of the frequency selection time. For example, this can be a dedicated core/cpu, as shown in Figure 6b. Now the question is: how useful is the adoption of a specific core/processor for this job? The use of a specific CPU for this processing introduces additional power consumption that can be low if working in a very low power mode but, at any rate, it is non-zero. This problem introduces the theoretical limits on the power scaling mechanism that are described in the next Section.

VI. THEORETICAL LIMITATIONS ON THE POWER SCALING MECHANISM

The frequency selection is a critical operation that does not depend exclusively on the hardware limitations. Even considering a perfect device with instantaneous frequency scaling, the frequency selection time is non-zero. Therefore, the frequency selection has theoretical limitations, due to the fact that the logic behind this selection is not instantaneous; even if the best next frequency is chosen, a delay is introduced, which can increase the total power consumption (the energy spent in performing the calculation is directly proportional to its duration, which depends on the complexity of the operation, having fixed the speed of the CPU doing the computation).

The critical variable in this problem is the choice of the time-slice to execute the frequency selection phase. Obviously, decreasing the time-slice increases the operations of frequency selection that allow fitting the traffic profiles in the best possible way. The disadvantage is the resulting increase in energy consumption due to the operations of selection and frequency scaling.

In order to analyze the additional power consumption that can be introduced by varying the time-slice for the frequency selection, we consider an ideal model where the time necessary to execute the frequency scaling is zero, without service interruption, and the performance is always the best possible. The tests were done by using Internet traffic traces that are publicly available [14] and part of "A Day in the Life of the Internet" [15], on the same architecture described in Section III.

Figure 7 reports the total power consumption based on the variation of the period and the time spent for the frequency scaling operation. The variation of the time necessary to choose the best next frequency practically does not influence the total power consumption. On the contrary, the time-slice used in the frequency selection phase influences the total power consumption in a linear way. Only for the first value of the time window (1 ms), there is a significant influence of the power consumed during the frequency selection upon the total.

In Figure 8, we show the same situation, but in this case the frequency selection time is ideally zero and the total power consumption is influenced only by the frequency selection period (i.e., there is no additional energy consumption due to the governor for selecting the next frequency).

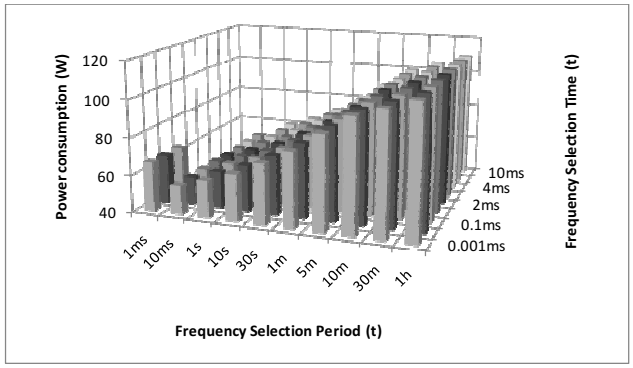

Figure 7. Theoretical power consumption with an ideal power scaling mechanism. The time necessary to choose the frequency linearly depends on the frequency selection time parameter.

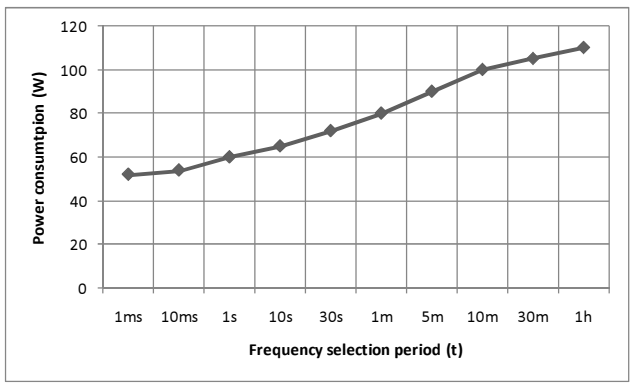

Figure 8. Theoretical power consumption with an ideal power scaling mechanism on the variation of the frequency scaling period. The time spent to choose the best next frequency is ideally zero.

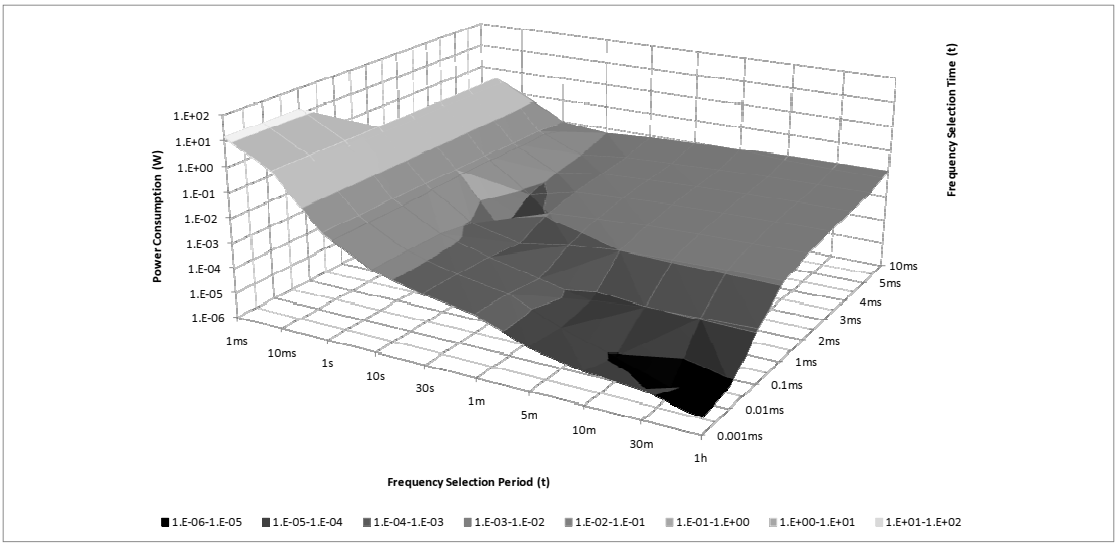

Figure 9. Increase in power consumption considering the time spent to choose the next frequency with respect to the case without frequency scaling.

The increase in power consumption due to the presence of the governor and to its complexity (represented by the frequency selection time) is shown in Figure 9. This figure, which shows the difference between the values in Figures 7 and 8, suggests that the smallest increase in energy consumption occurs when there are few frequency scaling operations. On the contrary, when the governor is applied too frequently and has a non-negligible complexity, energy gains of power scaling may be fruitless. Thus, there is a clear trade-off between the benefits of frequently switching the operating frequency, and the additional costs for estimating the new energy-aware configuration.

VII. Conclusions And Future Works

In this paper, we focused on the theoretical and technological limitations in adopting a power scaling mechanism. In detail, we compute several tests to identify the technological limitations in COTS hardware with power capabilities. The results show that the power consumption grows with the increase of the frequency. Power scaling allows a linear trade-off between consumption and network performance. The second part of this paper focuses on theoretical limitations and how the frequency selection can increase the power consumption. From the tests done with real Internet traffic data we note that, in order not to influence the power consumption, a SW governor is necessary, which runs on larger intervals and with a simple algorithm to choose the next equivalent frequency. Future work will aim to extend these studies by including a more detailed analysis about the idle state, and on how it is possible to use power scaling in a device with idle power mechanism without reducing performance.

References

[1] Research News, Berkeley Labs: ”Berkeley Lab Researchers Are Developing Energy-Efficient Digital Network Technology”, URL: http://www.lbl.gov/Science-Articles/Archive/EETD-efficient-networks.html.

[2] European Commission DG INFSO, "Impacts of Information and Communication Technologies on Energy Efficiency", Final Report, Sept.2008, http://ec.europa.eu/information_society/newsroom/cf/itemdetail .cfm ?item_id=4441.

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

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

[5] Bolla, R.; Bruschi, R.; Carrega, A.; “GreenSim: An open source tool for evaluating the energy savings through resource dynamic adaptation”, Proc. 2010 Internat. Symp. on Performance Evaluation of Computer and Telecommunication Systems (SPETCS 2010), Ottawa, Canada, July 2010 (to appear).

[6] Nedevschi, S.; Popa, L.; Iannacone, G.; Wetherall, D.; Ratnasamy, S.; “Reducing network energy consumption via sleeping and rate-adaption”, Proc. 5th USENIX Symposium on Networked Systems Design and Implementation, San Fransisco, CA, 2008, pp. 323-336.

[7] IEEE P802.3az Energy Efficient Ethernet Task Force, URL: http://www.ieee802.org/3/az/index.html

[8] Yi, Z.; Waskiewicz, P.J.; “Enabling Linux network support of hardware multiqueue devices”, Proc. 2007 Linux Symp., Ottawa, Canada, June 2007, pp. 305-310.

[9] Egi, N.; Greenhalgh, A.; Handley, M.; Iannaccone, G.; Manesh, M.; Mathy, L.; Ratnasamy, S.; “Improved forwarding architecture and resource management for multi-core software routers”, Proc. IFIP conference on Networking and Parallel Computing (NPC), 2009.

[10] Bolla, R.; Bruschi, R.; Ranieri, A.; “Green support for PC-based software router: Performance evaluation and modeling”, Proc. 2009 IEEE International Conference on Communications (ICC 2009), Dresden, Germany, June 2009.

[11] Bolla, R.; Bruschi, R.; Ranieri, A.; “Performance and power consumption modeling for green COTS software router”, Proc. 1st International Conf. 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] Intel® Core™ i5 Processor Datasheet, http://download.intel.com/design/processor/datashts/322164.pdf

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

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