Simulating the Power Consumption of Computer Networks

Ricardo Lent
Department of Electrical and Electronic Engineering
Imperial College, London SW7 2BT, UK
r.lent@imperial.ac.uk

***Abstract*—We introduce a high-fidelity simulation model to evaluate the power consumption of computer networks derived from measured data on routers. To this end, we design and use a power consumption profiling system to parameterize the electrical power consumption of network elements in terms of data traffic. A model validation done by comparing simulation outputs for a test case with actual measurements from an equivalent network testbed confirms the accuracy of the model. Finally, we apply the model to investigate power saving routing on a representative MPLS network.**

I. Introduction

The Internet consumes over 1000 billion KWh per year [1], making it a significant user of the world's energy production. The introduction of more efficient mechanisms can help to diminish the Internet's energy consumption per user. These mechanisms must consider not only the design of less power hungry electronic components, but also a more efficient use of the supporting infrastructure. A way to achieve energy efficiency is by introducing power-aware operating policies aimed to optimize energy usage in addition to other, more traditional, network metrics. In this respect, while it is reasonably well understood how the network characteristics and dynamics affect the quality of communications, there is a rather limited notion of how these dynamics affect the energy usage of routers, servers and end computers. Network simulators have been a useful tool to study computer networks in the past. However, existing simulators have very limited support for studying the power consumption of nodes in wired networks. Support for power consumption simulation has been mainly focused to mobiles (e.g., to estimate battery usage). New power consumption models are needed to evaluate wired networks for the accurate design and evaluation of "greener" network algorithms. Router and computer manufacturers usually specify the maximum power consumption of their products. Unfortunately, this information is insufficient to derive accurate simulation models of power consumption given that the actual consumption would depend on each device's workload.

In this paper, we tackle the development of an accurate power consumption simulation model for computer networks. We approach the problem in the following way. First, we design a power consumption profiling system (PCPS), which allows to measure the power consumption of one (or more) network elements under workload. We use the PCPS to experimentally characterize the traffic load-to-power consumption relationship of a set of network elements (PC-based routers, servers and end machines). The measured data allows to derive suitable simulation models of network elements under known traffic conditions. To evaluate the model, we then integrate it to an existing packet-level simulator and compare simulation predictions for a test network to actual measurements from a testbed. Finally, we apply the simulation model to investigate the power consumption of a MPLS network under different operating policies, some of which include switching off unused routers.

II. Power Consumption Profiling

We define power consumption profiling as the process of parameterizing a network node (or nodes) power consumption in terms of its (their) workload. We introduce the PCPS for this purpose. Because our main focus is on studying routers, we assume that the main node's workload comes from processing packets. However, the PCPS is general enough to be applicable to other cases.

A. *Profiler components*

PCPS uses a black-box approach where the packet processing rate-to-power relationship of an object is studied only in terms of the inputs and outputs without knowledge of the internal workings.

- *Workload generator* is a controllable traffic source that can produce packet streams with desired statistical properties (e.g., constant-bit rate, exponential, self-similar). The workload therefore is produced by one or more packet streams and directed to the object, and generated by one or more computers and from one or more network interfaces. Depending on the computing resources employed, more than one traffic source may be needed to produce higher workload to the object.
- *Data logger* allows to collect, store and report relevant information. In particular, it allows to relate power consumption to workload metrics (packet processing rate, CPU load, memory usage, etc.) The packet processing rate was obtained by periodically sampling the network interfaces' usage (`ifconfig` in Linux) for the number of packets in and out as well as the byte counts and by taking into account any possible counter overflow. Additional system metrics can be obtained from the Linux

/proc filesystem if needed. However, we limited this paper to traffic load. Note that the system is mainly useful to study power-workload in steady-state (i.e., under a constant traffic flow) because the data collection from distributed sources. On the other hand, the data logger can optionally offer data reporting by exposing stored data to information consumers (as a web service).

- *Power reader* is a system consisting of a power meter and a computer interface, which together can provide an instantaneous reading of the object's power consumption in a machine readable format. There are various alternatives to achieve this objective. The Advanced Configuration and Power Interface (ACPI) exposes power related information of various PC components (through /proc/acpi/* in Linux). However, it is mainly applicable to laptops and other mobile devices operating on a battery. Similarly, the Intelligent Platform Management Interface (IPMI), which was initiated by Intel as a standard, defines a set of interfaces that allow monitoring and managing a computing system. Unfortunately, there is limited power metering support in most installed hardware (in particular, older hardware) for both ACPI and IPMI, which in most cases only cover some components. It is expected that future computing systems will have a better power monitoring support. An alternative (and more accurate) way to measure the power consumption of a system is with the help of an external power meter (e.g., a multimeter with a computer interface).

B. Instrumenting a Power Reader

An initial approach was to employ commercially available power meters (model N67FU), which can measure power in the range 0.2 – 3120 W at one sample per second and with one decimal precision. The meter was coupled to a computer via optical recognition, which translates the pictures periodically taken from the meter's display with a webcam into a floating point number. We found existing optical character recognition (OCR) programs unsuited for the task of reading 7-segment style numbers. All open-source programs tested (Tesseract, GOCR and Ocrad) failed to read the numbers. Our approach was to develop our own 7-segment optical recognition program (DOR). First a level threshold (T) is applied to the picture to reduce the image to a 1-bit color depth image. DOR is calibrated with the expected location of each of the 7 segments for each digit, with each of them represented by a rectangular area. The presence or lack of bits in each of these areas is used to compute the reading with a maximum likelihood decision rule. Threshold T is dynamically adjusted with every reading to help DOR cope with lighting changes in the measuring room. Each reading takes about 300 ms to complete (including picture taking, pre-processing the image and computing the decision algorithm). The program was scheduled to run once per second to match the update rate of the power meter device. A second approach involves deploying a sensor network for power monitoring [2].

Regardless of the power reading approach, the data logger correlates the electrical power readings to the observed object's performance metrics. For monitoring and demonstration purposes, the power reader also stores a limited number of recent readings exposed to a variety of data consumers through a JSON interface.

III. Power Measurements

We implemented a PCPS prototype and conducted a series of measurements to evaluate the power consumption of common network nodes under different throughput conditions. The prototype is capable of running automated tasks, so can do extended experiments and measurements with limited supervision.

We limited the measurements to PC-based equipment which was available to conduct this study. We examined three main operating cases: when (i) an application transmits packets at a controllable rate in a busy-wait loop; (ii) an application receives packets in blocking mode and; (iii) the network layer receives (and drops) or forwards packets to other nodes (at the kernel level). We exclude results for non-blocking application operations (both for the sending and receiving cases) because of the great variability in CPU use (and power consumption) which is caused by interleaving processes in the O.S. For each testing case, we established one or more CBR traffic flows with small (16B) medium (128B) or large packet sizes (1024B).

The results are depicted in figures 1–2. For illustration purposes, we show the results obtained from two computer types both of which run Linux 2.6. The first computer type (PC1) was a single-core CPU Intel Pentium 4 at 2.4 GHz with a single 4-port server. The second one (PC2) was a dual-core CPU Intel E6300, with each core running at 1.86 GHz, and equipped with two 4-port servers. Each 4-port server is provided by a D-Link DFE-580TX adapter, controlled by the sundance driver in Linux. Each 4-port server increases 1–2 W the idle consumption level of the machine. When forwarding packets, we experimented with 1, 2 and 4 concurrent and physically separated flows (different interfaces and cables). These cases are labeled as "1F", "2F" and "4F" respectively in the figures' legends.

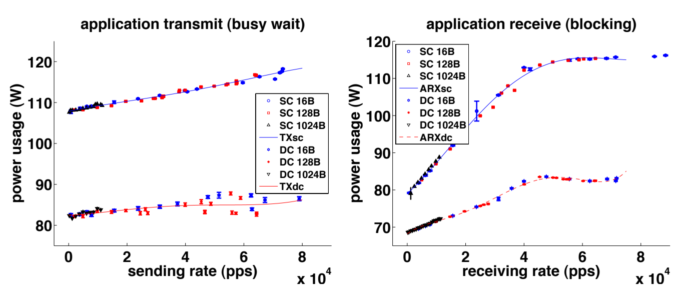

Fig. 1. Power Measurements

A polynomial fitting to the data is indicated in the figures (as lines). The fitting has the form: $Pm = \beta(\sum_{i=0}^{n} m_i \lambda^{n-i}) + \alpha$, where α is the idle router power consumption (i.e., switched on but without handling traffic) and β is the power variation range. Values for α and β for PC1 router were determined as

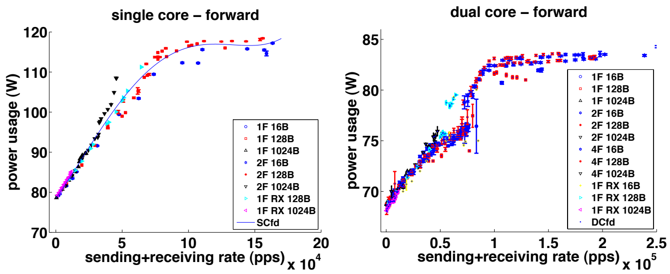

Fig. 2. Power Measurements

79.002 and 68.923, and for PC2 router, 36.012 and 9.8359 respectively. Up to 5th degrees produced reasonable errors (values for m_i can be found in Table I).

TABLE I

PC single core, 2 Eth quads	
operation mode	*polynomial coefficients*
transmit (busy-wait)	-1.16e-9 0.23e-3 105.76
receive (blocking)	2.71e-18 -4.54e-13 1.55e-8 7.3e-4 78.89
forward or layer-3 rx	1.5e-19 -4.4e-14 1.57e-09 5.2e-4 78.02
PC dual core, 2 Eth quads	
operation mode	*polynomial coefficients*
transmit (busy-wait)	3.12e-13 -2.61e-8 0.6e-3 79.34
receive (blocking)	1.9e-22 -3.3e-17 1.9e-12 -4.2e-08 6.3e-4 67.92
forward or layer-3 rx	2.64e-20 -1.3e-14 1.6e-09 5.95e-05 69.34

A. Estimation via Interpolation

Interpolating the data set can produce more accurate results than polynomial approximation in numeric calculations, which is the case of computer simulations. The measurements suggest that the main power consumption driver in a network node is the packet processing rate (rather that the bit processing rate, as seen from the results when using different packet sizes). This makes sense given the CPU overhead comes from processing packet headers. This situation considerably simplifies the construction of the model, which can be stated in the following terms when using linear interpolation. Consider a static array M, which is used to store K ($0 \leq k < K$) observed packet rates and power consumption pairs (λ_k, P_k), in ascending order by λ_k. M's first entry: $\lambda_0 = 0$ is when the router is idle. M's last entry: λ_{K-1} corresponds to the maximum attainable packet rate ($\lambda_{K-1} = \Lambda$). To determine the power consumption $P_i(t)$ at time t of node i operating at $\lambda_i(t)$ packets per second:

Algorithm 1 Calculate $P_i(t)$

Require: $0 \leq \lambda_i(t) \leq \Lambda, M$
  $i \leftarrow 1$
  while $\lambda_i(t) > M[i].\lambda$ **do**
    $i \leftarrow i + 1$
  end while
  $P_i(t) = (M[i].P - M[i-1].P)/(M[i].\lambda - M[i-1].\lambda) * (\lambda_i(t) - M[i-1].\lambda) + M[i-1].P$

IV. Simulation

To use the model, a simulator needs to calculate the packet processing rate at each node, which is straight forward to achieve in flow-based simulations. On the other hand, in packet-based network simulations, the simulator needs to compute the packet processing rate of each node by counting the packets observed during a certain time period. Counting can be achieved in the following way. Assuming that the simulated time is kept in variable t ($t \geq 0$, simulation run starting at 0) for each node i of interest ($0 \leq i \leq N$, where N is the number of nodes of interest from a power consumption standpoint), the following operations occur:

- Variables $E_i(t')$ and $P_i(t')$ represent for node i, the total energy consumption and power consumption respectively at time t' ($0 \leq t' \leq t$). These variables are initialized at $t = 0$ as $E_i(0) = 0$, $P_i(0) = \alpha$.
- To calculate the packet processing rate $\lambda_i(t')$, table W_i in node i keeps the packet processing times tp_j (timestamp for the $j - th$ arrival or departures) up to w seconds in the past (i.e., $t - w \leq tp_j \leq t$). This operation can be easily realized with a FIFO queue Q_i where the time of each new packet arrival or departure is pushed to the back of the queue while entries older than $t - W$ are popped out from its head. Therefore, $\lambda_i(t) = ||Q_i||/w$, where $||Q_i||$ is the length of the queue at time t. The net effect of this operation is to compute the average packet rate over the period of observation (w). The actual value of w would depend on the desired observation granularity and memory constraints (e.g., 100–1000 ms).
- The simulator would only need to calculate $\lambda_i(t)$ at the times of interest (t'), which correspond to the execution times of one of two possible events: (i) processing a user request (e.g., when the user queries the node's power consumption) or (ii) a packet arrival or departure. Therefore, t' is a subset of t in an event-driven simulation and is used as follows:

Algorithm 2 Update energy consumption at node i

Require: t, $t' \leq t$, $i \in n$
  compute $\lambda_i(t)$
  $E_i(t) \leftarrow E_i(t') + (t - t')P_i(t')$
  Update $P_i(t)$ as in Algorithm 1
  $t' \leftarrow t$

To evaluate the power consumption model, we integrated it to a packet network simulator. To validate the model, we deployed a network testbed (depicted in Figure 3) and constructed an equivalent simulation model. On both instances, we run the same set of experiments to evaluate the fidelity of the simulations by comparing its results to the observed behavior of the network testbed under equivalent testing conditions.

The test consisted in establishing two traffic flows at rates λ_0 and λ_1 to follow paths A, B, C, D and A, E, D respectively and measuring (and simulating) the power consumption of the three central routers (depicted inside the square in Figure 3).

All three PCs were dual-core machines exhibiting different power consumption.

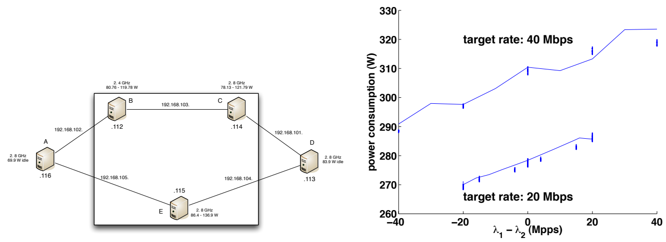

Fig. 3. Test network (left) and validation results (right)

Because of the addressing model, the simulation slightly differs from the testbed in that MPLS (with a null shim header) was employed to setup the two paths instead of static routing as in the testbed. However, the resulting packet flows are comparable. Flow rates were configured such that $\lambda_0 + \lambda_1 = \lambda_T$. λ_T was equal to 20 Mbps in one case and 40 Mbps in a second case. As depicted in Figure 3, the simulation results (continuous line) closely matched the observations (dots) for various values of λ_0 and λ_1. The x-axis represents the rate difference between the two flows, so that at the center both flows carry the same amount of traffic $\lambda_0 = \lambda_1 = \lambda_T/2$ and at the extremes only one flow carry all of the traffic λ_T.

A. Startup and Shutdown

In addition to the power consumption during regular operating periods, we measured the power usage during node's startup and shutdown (Figure 4 illustrates the case of PC1). For simulation purposes, we define periods τ_{up} and τ_{down} to represent the starting up and shutting down time intervals respectively. During these intervals, nodes are not able to receive or send packets in the simulator. However, they do incur in a (constant) consumption of α_{up} and α_{down} Watts, which accounts for the average power measured during that period. When off, nodes consume α_{off}. Values derived from the measured data are given in Table II.

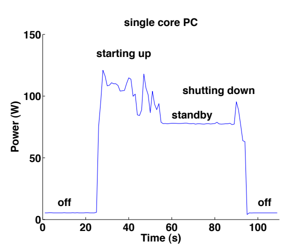

Fig. 4. Power usage during start up and shutdown.

TABLE II
PARAMETERS FOR STARTUP/SHUTDOWN

node	τ_{up}	α_{up}	τ_{down}	α_{down}	α_{off}
PC1	32.0s	101.34W	5.0s	65.37W	5.4W
PC2	30.0s	79.45W	6.0s	41.55W	2.2W

V. Application Scenario

We apply the simulation model previously described to investigate potential power savings in a scaled-down version of the GÈANT 2 network (as of February 2009 [3]).

Links are defined in the range of 1–8Mbps (rather than of a Gbps order as in the real network) and router power consumption is modeled after PC-based routers.

The network works as a 37-node MPLS network in the simulations for 20 additional nodes, which serve either as sources or destinations of traffic. Sources and sinks were arbitrarily connected to some of the 37 routers. Each MPLS router was associated to one of five available power consumption profiles measured from the testbed. Profiles were assigned in a round-robin fashion.

The network workload consists of 16 traffic flows of γ bps each. The study aims to evaluate the power consumption of the entire MPLS network for a range of γ values. While a discussion of power optimization and quality-of-service tradeoffs is beyond the scope of this paper, we provide a brief optimization example to illustrate the point and demonstrate a concrete application of the simulation model. We evaluate three operating policies in the network, which include a combination of switching off unused routers and re-routing flows to minimize a power-based total cost function (based on the shortest path) [4]. The estimated power consumption is depicted in Figure 5 and compares the cases of no optimization (all routers switched on routing based on the hop-count without link capacity considerations) with two cases where unused routers are switched off. In the first case, routing remains based on hop count, whereas in the latter case the routing cost aims to minimize the total MPLS network power usage. Routing optimization and switching off unused equipment can clearly save power as suggested by the simulation results. In terms of the packet delivery ratio, the power-based routing cost produced slightly better results to the others. This is because we avoided loading links beyond a preset threshold (about 80% the link capacity) when an power cost was used.

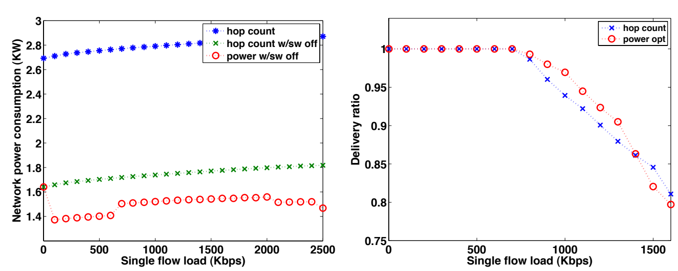

Fig. 5. Power consumption and packet delivery ratio vs. offered load

VI. Related Works

The literature offers a few power consumption models of network components. Notably, Tao et al. [5] proposed a methodology to evaluate the power consumption of switch fabrics in routers. Wang et al. developed an architectural-level power model applicable to different router microarchitectures

[6]. On the opposite order of magnitude, Baliga et al. [7] estimated the energy consumption of access networks and the Internet as a whole.

Another set of related works addressed ways to reduce energy consumption. One of the earliest works that suggested the need for energy efficiency in computer networks was done by Gupta and Singh [8], who although did not address a specific approach, discussed the impact of putting selected network elements to sleep, including routers, switches and interfaces. A related work proposed the use of on/off links in interconnection networks [9]. Another discussion was put forward by Christensen et al. on power management for communication networks [10]. The case of putting idle ports to sleep was further addressed by Ananthanarayanan and Katz [11], who proposed three opportunistic power reduction schemes specific to network switches. The use of rate adaptation in addition to sleeping [12] has also been suggested.

Chiaraviglio et al. [13] proposed an algorithm for switching off nodes and links to minimize energy consumption by using the total network power consumption the objective function of a shortest path-based algorithm. A comparable approach was proposed by Cardona et al. [14]. These two works assumed an arbitrary model for nodes' power consumption. An approach for guaranteed QoS while minimizing energy use was proposed by Qu and Potkonjak [15]

A discussion on the power consumption of p2p applications compared to their centralized versions was done by Nedevschi et al. [16]. An specific approach to reduce the energy consumption of BitTorrent was addressed by Blackburn and Christensen [17]. Server assignment optimization was addressed in [18].

Related works involving power consumption measurements of network devices are rather limited in the literature. Chabarek et al. [19] measured power consumption in commercial routers. However, they focused on a very narrow operating space as their paper had a different aim than ours. Another related work was done by Kansal and Zhao [20], who discussed energy profiling for application design. A close work to ours, but with a different aim and approach, was developed by Bolla et al., who also measured power consumption in PC-based routers.

VII. Conclusions

In this paper, we have addressed the problem of accurately simulating the power and energy consumption of computer networks. We approached the problem by creating simulation models from a data set that was measured on PC-based routers. To obtain the measurements, we designed and constructed a power profiling system that served to experimentally parameterize power consumption in terms of data traffic. Our measurements indicated a relatively fixed relationship between power consumption and router workload. However, we could expect variations in this relationship under different operating environments, hardware or software upgrades, or under different traffic patterns and packet handling policies. While we focused mainly on routers, the profiling system can be applied to a wider set of devices (e.g., to profile servers). A direct comparison between simulation results and measured data on a network testbed validated the fidelity of the models. A continuation of this work will address the design of power-efficient algorithms for computer networks, which will be evaluated with the tools elaborated in this paper.

Acknowledgment

This work was partially supported by FP7 FIT4Green.

References

[1] J. G. Koomey, "Estimating total power consumption by servers in the u.s. and the world", 2007.

[2] Ricardo Lent, "A sensor network to profile the electrical power consumption of computer networks", in *GreenCom3, Proceedings of the 3rd International Workshop on Green Communications*, 2010.

[3] Geant 2, "Geant website", *http://www.geant2.net*, .

[4] Ricardo Lent, "Electrical power usage monitoring system", http://san.ee.ic.ac.uk/epums.

[5] Terry Tao Ye, Luca Benini, and Giovanni De Micheli, "Analysis of power consumption on switch fabrics in network routers", in *In Proc. Design Automation Conference*, 2002, pp. 524–529.

[6] Hang-Sheng Wang, Li-Shiuan Peh, and Sharad Malik, "A power model for routers: Modeling alpha 21364 and infiniband routers", *IEEE Micro*, vol. 23, no. 1, pp. 26–35, 2003.

[7] J. Baliga, K. Hinton, and R. S. Tucker, "Energy consumption of the internet", in *Proceedings of the First International Workshop on Green Communications*, June 2009.

[8] Maruti Gupta and Suresh Singh, "Greening of the internet", in *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, New York, NY, USA, 2003, pp. 19–26, ACM.

[9] Vassos Soteriou and Li-Shiuan Peh, "Dynamic power management for power optimization of interconnection networks using on/off links", *High-Performance Interconnects, Symposium on*, vol. 0, pp. 15, 2003.

[10] Kenneth J. Christensen, Chamara Gunaratne, Bruce Nordman, and Alan D. George, "The next frontier for communications networks: power management", *Computer Communications*, vol. 27, no. 18, pp. 1758 – 1770, 2004, Performance and Control of Next Generation Communications Networks.

[11] Ganesh Ananthanarayanan and Randy H. Katz, "Greening the switch", Tech. Rep. UCB/EECS-2008-114, EECS Department, University of California, Berkeley, Sep 2008.

[12] Sergiu Nedevschi, Lucian Popa, Gianluca Iannaccone, Sylvia Ratnasamy, and David Wetherall, "Reducing network energy consumption via sleeping and rate-adaptation", in *NSDI'08: Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, Berkeley, CA, USA, 2008, pp. 323–336, USENIX Association.

[13] L. Chiaraviglio, M. Mellia, and F. Neri, "Energy-aware networks: Reducing power consumption by switching of network elements".

[14] J.C.C. Restrepo, C.G. Gruber, and C.M. Machuca, "Energy profile aware routing", in *Proceedings of IEEE International Conference on Communications Workshops*, June 2009, pp. 1–5.

[15] Gang Qu and Miodrag Potkonjak, "Energy minimization with guaranteed quality of service", in *ISLPED '00: Proceedings of the 2000 international symposium on Low power electronics and design*, New York, NY, USA, 2000, pp. 43–49, ACM.

[16] Sergiu Nedevschi, Sylvia Ratnasamy, and Jitendra Padhye, "Hot data centers vs. cool peers.", in *HotPower*, Feng Zhao, Ed. 2008, USENIX Association.

[17] J. Blackburn and K. Christensen, "A simulation study of a new green bittorrent", in *COIN-ACOFT*, 2007.

[18] Luca Chiaraviglio and Ibrahim Matta, "GreenCoop: Cooperative Green Routing with Energy-efficient Servers", Tech. Rep. BUCS-TR-2009-034, CS Department, Boston University, November 24 2009.

[19] Joseph Chabarek, Joel Sommers, Paul Barford, Cristian Estan, David Tsiang, and Steve Wright, "Power awareness in network design and routing", in *In Proc. IEEE INFOCOM*, 2008.

[20] Aman Kansal and Feng Zhao, "Fine-grained energy profiling for power-aware application design", *SIGMETRICS Perform. Eval. Rev.*, vol. 36, no. 2, pp. 26–31, 2008.