Theoretical and Applied Informatics
ISSN 1896–5334
Vol.24 (2012), no. 4
pp. 257–270
DOI: 10.2478/v10179-012-0016-1

Optimising Server Energy Consumption and Response Time

EROL GELENBE, RICARDO LENT

Intelligent Systems and Networks Group
Department of Electrical and Electronic Engineering
Imperial College, London SW7 2BT, UK
{e.gelenbe,r.lent}@imperial.ac.uk

Received 5 October 2012, Revised 30 October 2012, Accepted 15 November 2012.

Abstract: The predicted annual growth of energy consumption in ICT by 4% towards 2020, despite improvements and efficiency gains in technology, is challenging our ability to claim that ICT is providing overall gains in energy efficiency and Carbon Imprint as computers and networks are increasingly used in all sectors of activity. Thus we must find means to limit this increase and preserve quality of service (QoS) in computer systems and networks. Since the energy consumed in ICT is related to system load,]this paper discusses the choice of system load that offers the best trade-off between energy consumption and QoS. We use both simple queueing models and measurements to develop and illustrate the results. A discussion is also provided regarding future research directions.

Keywords: Energy Consumption, Quality of Service, Optimisation, Computer Systems

1. Motivation and Problem Setting

CO_2 emissions due to energy consumption by ICT by 2020 is expected to exceed 1.4 Billion Tonnes. Thus it is particularly important to achieve energy savings in the ICT sector whose annual increase in electricity consumption is of the order of 4%, despite the significant improvements in the efficiency of hardware technologies that are being achieved year after year. Such issues are relevant both for communication systems such as base stations for mobile phones, and routers in networks, as well as for personal computers, office servers, and for large data centres. Indeed, by 2020 telecommunication systems are predicted to consume some 25% of the energy that is used in ICT systems, while data centres will account for some 18% of the total, and the rest will be caused by personal devices, office computers and servers.

The increasing importance of the Internet and the great interest in reducing energy consumption in all areas of human activity, increase the importance of making computer

systems and networks energy efficient. The improvement of a network's efficiency while respecting the users' QoS needs and the prior service level agreements (SLA) has been explored experimentally [31], while it has also been discussed for Cloud Computing [7]. Early work on energy savings in the Internet [32] addressed routing to aggregate traffic along a few routes, and putting certain nodes and devices to sleep; a network-wide and link layer approach were discussed and later work [33] designed energy saving algorithms for Ethernet links. In [9] components were powered on/off in combination with a multicommodity network-flow for traffic assignment. Online technique were proposed in [43] to spread load through multiple paths, based on a step-like model of power consumption as a function of the processing rate and on nodes that automatically adjust operating rate to utilization. Rate-adaptation was suggested in [36] link utilization and delay. In [12] active links and routers are selected to minimize power consumption via simple heuristics, and in [13] specific backbone networks are discussed, and the potential overall energy savings in the Internet are evaluated in [14]. In [31] the reduction of power consumption in wired networks is studied together with QoS constraints, while the Cognitive Packet Network (CPN) routing algorithm [29], [30] using energy awareness and QoS is also considered. Much work has also been done for wireless networks, including via Topology Control (TC) [38], [34], [40] that adjusts transmission power and the range of each node while preserving the connectivity of source-destination pairs. In [8] it is indicated that the radio transceiver consumes almost the same amount of energy in transmit, receive and idle mode, so that switching off the transceiver results in significant energy savings. In [44] two algorithms for energy conservation running on top of existing ad hoc routing protocols are discussed and the effect of turning off nodes on latency and packet loss are also considered. In [28] energy aware routing for ad hoc networks is introduced using CPN using a metric called path availability that is proportional to the remaining lifetime of battery power at the nodes. In [9] a model for router power consumption is presented and evaluated on two widely used routers, indicating that the base system is the largest power consumer. Other work [36] has focuses on the relation between the hardware processing rate, traffic and power consumption. Recent work has also considered the link between energy consumption and packet travel delay in networks [20], [1], while the links between network routing and energy consumption in packet networks [21], [22] has also been studied using queueing network models [19].

The simplest means of energy savings is to turn off computers and network units that are not being used, provided one can restart them rapidly when requests for computation or communication services do arrive. In the ideal case, a service system would only be consuming energy during its busy periods. However, turning a system on and off will in itself consume energy. Thus this paper attempts to take a holistic view so as to find the best system operating point with regard to a composite metric that includes both energy consumption and QoS. For a system represented as a single server queue whose power

consumption is ω Watts when it is in operation, with a load factor or processor utilisation $\rho = \lambda E[S]$, where λ is the average arrival rate of jobs and $E[S]$ is the average service time, the average power consumption in watts, under ideal operation when energy is only used when jobs are being processed, is:

$$\Pi = \omega\rho, \tag{1}$$

and the energy consumption per job in Joules would then be:

$$J_{job} = \Pi/\lambda = \omega E[S]. \tag{2}$$

However, it is not easy to wake up a server instantaneously as soon as a job arrives for processing, and then to turn it off right after the processing ends. There will always be some energy consumption even if the processor is idle, and the waking up delay will also have to be taken into account. Furthermore, both putting the system to sleep and waking it up will cause additional energy expenditure.

Thus in this paper we will first consider the case when the processor is constantly on, and we will consider a cost function that includes both the response time to jobs, and the energy that is consumed per job on average. We obtain the optimum value of load which minimises the cost function. We will also use measurements on a system with a synthetic workload to estimate the power consumption parameters of the system, the average processing time per job, and finally we validate the theoretical results regarding the optimum load that minimises the cost function. Then we study the case when multiple such systems are being operated in parallel and we need to share a flow of jobs to the system so as to optimise a composite cost function similar to the previous one. Again, we derive the optimum share of load that must be assigned to each of the subsystems. All of the results are based both on analysis using simple queueing models, and on measurements of a system with a synthetic workload.

2. Combining Energy & QoS

The simple formulae given above do not take into account the fact that the power consumption will depend on the load [7], [35], and andy further controls such as putting a server to sleep and waking it up will take time and consume additional energy. Without taking into account the power needs of complex cooling equipment that is needed for large systems, a simple but fairly realistic power consumption relation for current processing units, that has been experimentally validated, has the form [23]:

$$\Pi = A + B\rho, \tag{3}$$

where A is the power consumption of the processing unit when it is idle. A very efficient processor might have a very small value of A, and B would correspond to the rate of

increase in power consumption as more more cores are turned on as the load increases. Unfortunately, for much of the current equipment A is still a significant part (often more than 50%) of the total processor power consumption when it is idle. This includes the fact that the memory system and the peripheral equipment and network connections need to be powered even when no jobs are being processed, and that the operating system can remain active (and hence contributes to the energy consumption) even when there are no external jobs that need to be processed. From π we obtain an expression for the energy consumption per job:

$$J_{job} = \frac{A}{\lambda} + BE[S], \tag{4}$$

which would justify the principle of concentrating computation on a small number of processing units in order to minimise the power consumption per job. However if one also wishes to consider the resulting quality of service (QoS) then it would be reasonable to examine the simple cost function:

$$\begin{aligned} C_{job} &= \frac{aE[S]}{1-\lambda E[S]} + bJ_{job}, \\ &= \frac{aE[S]}{1-\lambda E[S]} + \frac{bA}{\lambda} + bBE[S], \end{aligned} \tag{5}$$

where a and b are the relative importance that is being placed on the QoS for a job, and the energy consumption per job, and the QoS is represented in (6) by the average response time formula for a job, assuming Poisson arrivals and exponential service times for jobs in a single server queue.

2.1. Experimental Validation

To validate the energy-QoS metric and optimum load model, we conducted a series of experiments using jobs executing on a server class system having a quad-core Intel Xeon 3430 (8M cache, 2.4 GHz), 2 GB RAM, single 150 GB SATA hard drive, and 2 on-board Gigabit Ethernet interfaces. The system runs Linux (Ubuntu) with CPU throttling enabled with the *ondemand governor*, which dynamically adjust the cores' frequency depending on load. A client machine is attached to the server through a fast Ethernet switch to generate the workload, and the client machine also measures the system's power consumption.

We measured power consumption when it is idle, i.e. when it has no external jobs to execute, to be $A = 69.5$ Watts, which corresponds to the value of A in equation (4).

We then launched a sequence of synthetic jobs, where each job consisted in calculating the real number π, using Machin's formula, to a desired level of precision. This level of precision was used to provide a wide range in the execution time and workload from one job to the next by choosing the precision at random with a uniform distribution in

the range of 10 to 50 thousand digits. The job also included sending the results back to the client through the network connection. By recording the start and completion times of each job at the server, we measured the *average* job processing time to be 6.4235s, exclusive on any waiting or queueing times at the server.

We then varied the number of jobs arriving to the server, i.e. the rate λ. The resulting measured average response time for a job is shown in Fig. 1. as a function of load (ρ), together with the theoretical average response time R for a job, predicted using a Poisson arrival process and an exponentially distributed service time:

$$R = \frac{E[S]}{1 - \lambda E[S]}, \tag{6}$$

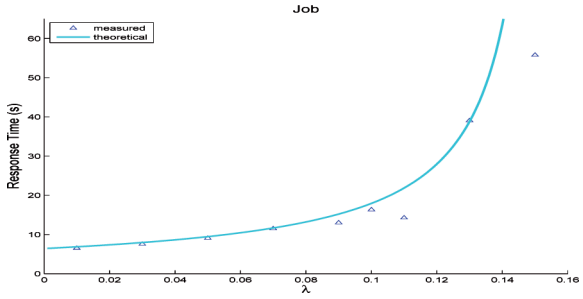


Fig. 1. Comparison of measured response time as a function of load $\rho = \lambda E[S]$, against the theoretical results from (6)

Then we measured the average energy consumed by a single job from observations obtained from serving a large number of jobs (1000), the average power consumption and the total running time of the experiment. The value of B was measured to be 13.24 Watts per job on average. The measured value of J_{job} and the calculated results from (4) using the experimentally estimated values of A and B are shown in Fig. 2.

3. Optimising Energy and QoS

A simple analysis allows us to compute the value of the arrival rate λ that minimises the composite cost function C_{job}. It is given by:

$$\lambda^* = \frac{1}{E[S]} \frac{\sqrt{\frac{bA}{a}}}{1 + \sqrt{\frac{bA}{a}}} \tag{7}$$

We therefore see that the optimum setting of the load $\rho^* = \lambda^* E[S]$ will depend just on A, the idle power consumption, and on the ratio b/a which is the relative importance

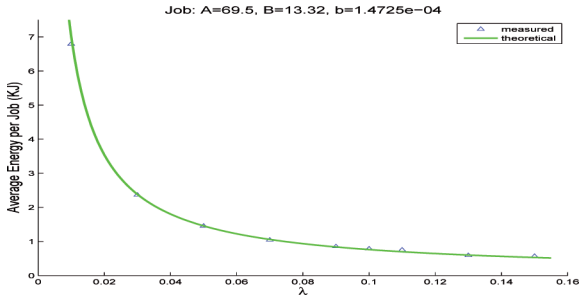


Fig. 2. Measured energy consumption per job (in KJoules) as a function of load ρ compared to the value predicted by the expression (4) using the experimentally measured values $B = 13.24$ and $A = 69.5$ Watts. The parameter b in (4) has been set to $b = 1.4725 \times 10^{-}04$ which is the inverse of the maximum energy consumption per job (in Joules) that was measured during the experiments

that we put on energy consumption with respect to the average response time of jobs, as given by the formula:

$$\rho^* = \frac{\sqrt{\frac{bA}{a}}}{1 + \sqrt{\frac{bA}{a}}} \tag{8}$$

The expression (8) gives us a simple rule of thumb for selecting system load for optimum operation, depending on how we weigh the relative importance of energy consumption with respect to average response time or how fast we are getting the jobs done.

We can also see that ρ^* is an increasing function of the ratio bA/a. In particular if we set $x = bA/a$, we have:

$$\frac{\partial \rho^*}{\partial x} = \frac{1}{2\sqrt{x}[1 + \sqrt{x}]^2} \tag{9}$$

This gives us the specific way in which the optimum load on the system should increase as the system's idle power consumption, and/or the relative importance that we place on energy, increase.

The optimum value of the load factor ρ^* for different values of a when b is normalised at $b = 1$ is shown in Fig. 3. On the other hand, Fig. 4 shows the resulting values of C^*_{job}.

In Fig. 5 we have also $b = 1.4725e-04$, which is the inverse of the maximum energy measured during the experiments. Four cases are illustrated. To represent the different levels of importance that my be attributed to the delay, a has been varied between $0.01 \times b$ to $100 \times b$.

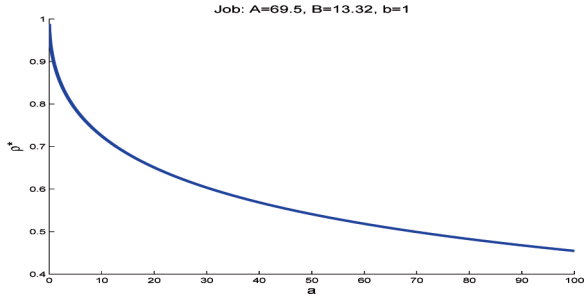


Fig. 3. The value of ρ that minimises the overall cost function C_{job} per job, given as a function of the relative importance of the average response time in the cost function, when we set $b = 1$

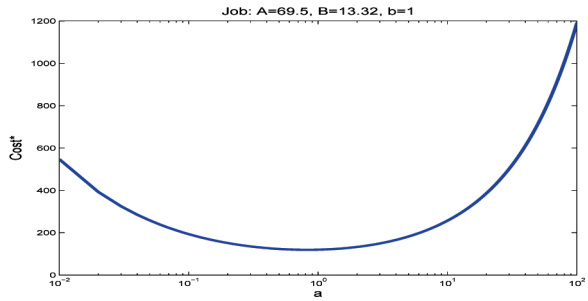


Fig. 4. The minimum value of the cost function C^*_{job} obtained by setting the load to its optimum value ρ^* for different values of the relative importance of the average response time a, when we set $b = 1$

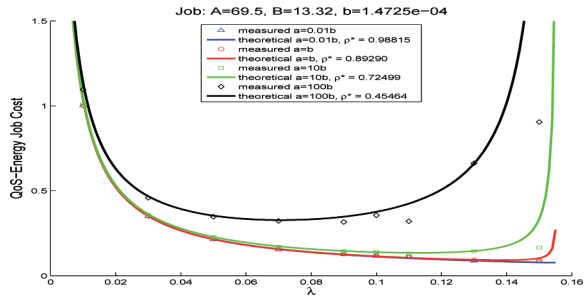


Fig. 5. Comparing theoretical predictions and experimental measurements for the overall cost as a function of load

4. Load Sharing in N Sub-Systems

In many cases, a data centre is composed of N heterogeneous sub-systems, and our analysis can provide guidelines about how to share load among them, where each one has an energy profile described by the parameters A_i and B_i and with different processing capacity. The base system will be system 1 in the sense that $B_1 \leq B_i$ for $i \neq 1$. The execution of a job on system i will take on average time $E[S_i] = E[S_1]/\sigma_i$ where σ_i is the speed-up factor for system $i \neq 1$ with respect to the base system, and we *do not* imply that the base system is the slowest one.

With a flow of λ jobs per unit time as whole and we assign a fraction p_i of the flow to the $i-th$ sub-system, and denote $\lambda_i = p_i\lambda$ with:

$$1 = \sum_{i=1}^{N} p_i \tag{10}$$

The cost function of interest then becomes:

$$\begin{aligned} C_{job} &= \sum_{i=1}^{N} p_i \{ \frac{aE[S_i]}{1-\lambda_i E[S_i]} + bJ^i_{job} \} \\ &= \sum_{i=1}^{N} p_i \{ \frac{aE[S_i]}{1-\lambda_i E[S_i]} + \frac{bA_i}{\lambda_i} + bB_i E[S_i] \} \end{aligned} \tag{11}$$

is minimised with regard to the flows assigned to each sub-system, by computing:

$$\begin{aligned} \frac{\partial C_{job}}{\partial p_i} &= E[S_i](bB_i + \frac{a}{(1-\rho_i)^2}) \\ &\quad -E[S_1](bB_1 + \frac{a}{(1-\rho_1)^2}), 2 \leq i \leq N \end{aligned} \tag{12}$$

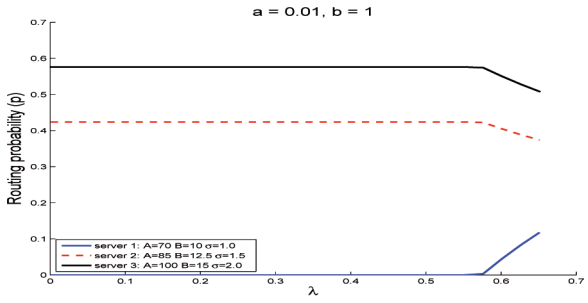
where $\rho_i = p_i\lambda E[S_i]$, so that to minimise the cost function we need to set:

$$\rho_i = 1 - \sqrt{\frac{a}{\frac{a\sigma_i}{(1-\rho_1)^2} + b[B_1\sigma_i - B_i]}} \tag{13}$$

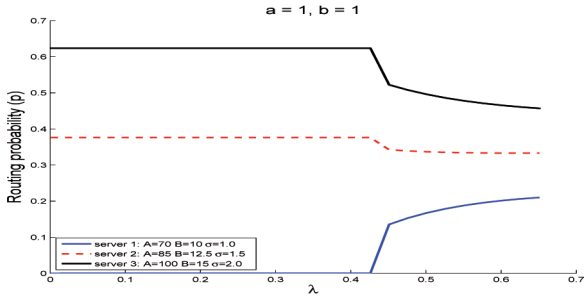
where $\sigma_i = E[S_1]/E[S_i]$ is the speed-up factor of running a job on system i with respect to system 1.

As a numerical example, consider three systems with speed-up factors $\sigma = 1, 1.5, 2.0$ (i.e, the second system is 50% faster than the first and the third system twice as fast). Assume that these systems have idle power consumption $A = 70, 85, 100$ watts respectively and let the B values be $B = 10, 12.5, 15$ watts per unit time; in this case

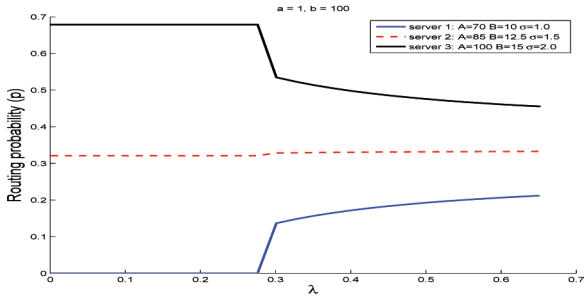
higher computing power also implies higher power consumption. From (13) with constraint (10) we obtain the optimum routing probabilities for the three systems as a function of the job arrival rate λ, as shown in Fig. 6. Despite its higher power consumption, the faster system is preferred at lower loads because it can produce the lowest energy consumption per job. By assigning a greater weight to the response time via the parameter a in (12), the routing probabilities to the other two systems will increase as expected. Setting $b = 1$, Fig. 7 shows how the *minimum* value of the cost function that combines Energy and Delay, with the optimum sharing of load among the three sub-systems, will vary for various values of load λ as a function of a.



(a) a=0.01



(b) a=1



(c) a=10

Fig. 6. Optimum load-sharing probabilities that minimize the Energy-QoS cost function for different values of a and b

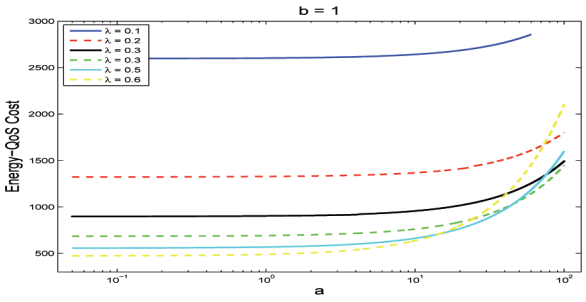


Fig. 7. The minimum (optimum) value of the Energy-QoS Cost as a function of a and b

5. Conclusions

This paper has shown, both through simple mathematical models and via experiments, that the load in a computer system can be tuned so that an optimum trade-off is achieved between the response time the system provides and the energy cost per job that is being executed. The analysis allows us to obtain a simple analytical expression for the optimum system load. The approach is then extended to the problem of load sharing among multiple sub-systems that may have different energy consumption and response time characteristics.

Much work remains to be done in these areas. A popular way to reduce energy consumption is to turn systems on when the load is sufficiently high, and then setting them to sleep at low loads; this leads to behaviours similar to those of unreliable systems [18] where the effect performance of the system is diminished by this on-off behaviour, while jobs arriving to an empty system which wake it up would in turn result in further service delays [17], in addition to the energy cost of putting a system to sleep and then waking it up again. In the context of packet networks, such patterns of activity have to be shown also to be quite disruptive in terms of packet loss, delay and jitter [31] so that they may be useful if the sleep-wake cycles can be well synchronised with fluctualtions in load [27]. Thus it will be important to see how they can be designed and optimised to maximise the benefit to energy savings, and minimise the harm to QoS of such schemes.

Another interesting direction will be to consider optimum task assignment [2] of jobs to processors or systems so that both execution times and energy savings are part of the relevant cost function. Inevitably, in large distributed environments such as the Grid or Cloud Computing, the information about loads and sub-system or system characteristics will be imperfect and incomplete [24] and the information arriving to a given user from various sources needs to be synchronised and fused [25]. Thus another important direction for research is to study situations in which decision need to be taken with incomplete, imperfect or delayed information, both from an analytical and theoret-

ical perspective and using measurements on real systems. We can also study situations where different classes of users would be accessing different classes of systems, so that the system as a whole is optimising multiple classes of criteria simultaneously [4], [26]. Thus there is a wealth of research problems that need to be addressed in these areas and we hope that this paper will create interest for this new area of systems performance engineering.

References

1. O.H. Abdelrahman, E. Gelenbe: *Packet delay and energy consumption in non-homogenous networks*. The Computer Journal 55(8):950-964, 2012.
2. J. Aguilar, E. Gelenbe: *Task assignment and transaction clustering heuristics for distributed systems*. Information Sciences 97 (1): 199-219, 1997.
3. M. Alonso, S. Coll, V. Santonja, J.-M. Martínez, P. López, J. Duato: *Power-aware fat-tree networks using on/off links*. 3rd International Conference on High Performance Computing and Communications (HPCC 2007), Houston, TX, pp. 472-483, 2007.
4. V. Atalay, E. Gelenbe: *Parallel algorithm for colour texture generation using the random neural network model*. International Journal of Pattern Recognition and Artificial Intelligence 6 (2& 3): 437-446, 1992.
5. J. Baliga, K. Hinton, R. S. Tucker: *Energy consumption of the Internet*. Optical Internet, 2007 and the 2007 32nd Australian Conference on Optical Fibre Technology. COINACOFT 2007. Joint International Conference on, June, pp. 1-3, 2007.
6. J. Baliga, R. Ayre, K. Hinton, W. V. Sorin, R. S. Tucker: *Energy consumption in optical IP networks*. Lightwave Technology, Journal of, 27, 2391-2403, 2009.
7. A. Berl, E. Gelenbe, M. D. Girolamo, G. Giuliani, H. D. Meer, M. Q. Dang, K. Pentikousis: *Energy-efficient cloud computing*. The Computer Journal, 53, 1045-1051, 2010.
8. A. Boukerche, X. Cheng, J. Linus: *A performance evaluation of a novel energy-aware data-centric routing algorithm in wireless sensor networks*. Wireless Networks, 11, 619-635, 2005.
9. J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, S. Wright: *Power awareness in network design and routing*. INFOCOM 2008. The 27th Conf. Computer Comms. IEEE, April, pp. 457-465, 2008.
10. J. S. Chase, D. C. Anderson, P. N. Thakar, A. M. Vahdat, R. P. Doyle: *Managing energy and server resources in hosting centers*. SOSP'01: Proceedings of 18th ACM Symposium on Operating systems principles, October, pp. 103-116, 2001. ACM.

11. G. Chen, W. He, J. Liu, S. Nath, L. Rigas, L. Xiao, F. Zhao: *Energy-aware server provisioning and load dispatching for connection-intensive internet services*. NSDI'08: Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation, April, pp. 337-350, 2008. USENIX Association.

12. L. Chiaraviglio, M. Mellia, F. Neri: *Reducing power consumption in backbone networks*. Communications, 2009. ICC '09. IEEE International Conference on, June, pp. 1-6, 2009.

13. L. Chiaraviglio, M. Mellia, M., F. Neri: *Energy-aware backbone networks: A case study*. Communications Workshops, 2009. ICC Workshops 2009. IEEE International Conference on, June, pp. 1-5, 2009.

14. L. Chiaraviglio, D. Ciullo, E. Leonardi, M. Mellia: *How much can the Internet be greened?* GLOBECOM Workshops, 2009 IEEE, 30 2009-Dec. 4, pp. 1-6, 2009.

15. D. Economou, S. Rivoire, C. Kozyrakis, P. Ranganathan: *Full-system power analysis and modeling for server environments*. Workshop on Modeling, Benchmarking and Simulation (MoBS), June 2006.

16. X. Fan, W.-D. Weber, L. A. Barroso: *Power provisioning for a warehousesized computer*. ISCA '07: Proceedings of the 34th annual international symposium on Computer architecture, June, pp. 13-23, 2007. ACM.

17. E. Gelenbe, R. Iasnogorodski: *A queue with a server of walking type*. Ann. Inst. Henri Poincare, XVI, 63-73, 1980.

18. E. Gelenbe, S. Tripathi, D. Finkel: *"Performance and reliability of a very large distributed system"*, Acta Informtica, Vol. 23, 643-655, 1986.

19. E. Gelenbe, R. R. Muntz: *Probabilistic models of computer systems - part I*. Acta Informatica, 7, 35-60, 1976.

20. E. Gelenbe: *"Search in unknown random environments"*, Phys. Rev. E 82: 061112 (2010) -– Published December 7, 2010.

21. E. Gelenbe, C. Morfopoulou: *"A Framework for Energy Aware Routing in Packet Networks"*, The Computer Journal 54 (6): 850-859, 2011, doi: 10.1093/comjnl/bxq092.

22. E. Gelenbe, C. Morfopoulou: *"Power savings in packet networks via optimised routing"*, ACM/Springer Mobile Networks and Applications 17:152-159, 2012.

23. E. Gelenbe, R. Lent: *Trade-offs between energy and Quality of Service*. The Second IFIP Conference on Sustainable Internet and ICT for Sustainability – SustainIT 2012, IEEE, Oct. 4-5, 2012, Pisa, Italy.

24. E. Gelenbe, G. Hebrail: *A probability model of uncertainty in data bases*, Proceedings of the second international conference on data engineering, pp. 328-333, 1986.

25. E. Gelenbe, K. Sevcik: *Analysis of update synchronization for multiple copy data bases*. IEEE Transactions on Computers 28 (10): 737-747, 1979.

26. E. Gelenbe, J.M. Fourneau: *Random neural networks with multiple classes of signals*. Neural Computation 11 (4): 953-963, 1999.

27. E. Gelenbe, C. Rosenberg: *Queues with slowly varying arrival and service process*. Management Science, 36(8), 928-937, 1990.

28. E. Gelenbe, R. Lent: *Power-aware ad hoc cognitive packet networks*. Ad Hoc Networks, 2, 205-216, 2004.

29. E. Gelenbe: *Cognitive packet network (CPN)*. U.S. Patent 6, 804, 20, 2004.

30. E. Gelenbe: *Steps towards self-aware networks*. Communications of the ACM, 52, 66-75, 2009.

31. E. Gelenbe, S. Silvestri: *Reducing power consumption in wired networks*. Computer and Information Sciences, 2009. ISCIS 2009. 24th International Symposium on, METU, North Cyprus Campus, 14-16 September, pp. 292-297, 2009. IEEE Digital Library.

32. M. Gupta, S. Singh: *Greening of the Internet*. Computer Communication Review, 33, 19-26, 2003.

33. M. Gupta, S. Singh: *Dynamic ethernet link shutdown for energy conservation on ethernet links*. Communications, 2007. ICC '07. IEEE International Conference on, June, pp. 6156-6161, 2007.

34. X. Jia, D. Li, D. Du: *QoS topology control in ad hoc wireless networks*. INFOCOM 2004. 23rd Annual Joint Conf. IEEE Computer and Comms. Societies, March, pp. 1264-1272 vol. 2, 2004.

35. R. Lent: *Power measurements of processors for routing*. Technical report. ISN Group, EEE Dept., Imperial College, London, UK, 2010.

36. S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, D. Wetherall: *Reducing network energy consumption via sleeping and rate-adaptation*. NSDI'08: Proc. 5th USENIX Symposium on Networked Systems Design and Implementation, Berkeley, CA, USA, pp. 323-336, 2008. USENIX Association.

37. C. Panarello, A. Lombardo, G. Schembra, L. Chiaraviglio, M. Mellia: *Energy saving and network performance: a trade-off approach*. e-Energy 2010 – First International Conference on Energy-Efficient Computing and Networking, April, pp. 41-50, 2010. ACM.

38. R. Rajaraman: *Topology control and routing in ad hoc networks: A survey*. SIGACT News, 33, 60-73, 2002.

39. J. Restrepo, C. Gruber, C. Machuca: *Energy profile aware routing*. Communications Workshops, 2009. ICC Workshops 2009. IEEE International Conference on, June, pp. 1-5, 2009.

40. P. Santi: *Topology control in wireless ad hoc and sensor networks*. ACM Comput. Surv., 37, 164-194, 2005.

41. V. Soteriou, L.-S. Peh: *Design-space exploration of power-aware on/off interconnection networks*. IEEE International Conf. on Computer Design, San Jose, CA, pp. 510-517, 2004. IEEE Computer Soc.

42. B. Urgaonkar, P. Shenoy, A. Chandra, P. Goyal, T. Wood: *Agile dynamic provisioning of multi-tier Internet applications*. ACM Trans. on Autonomous and Adaptive Systems (TAAS), 3, 1-39, 2008.

43. N. Vasic, D. Kostic: *Energy-aware traffic engineering*. Technical report. EPFL, 2008.

44. Y. Xu, J. Heidemann, D. Estrin: *Adaptive energy-conserving routing for multihop ad hoc networks*. Research Report 527. USC/Information Sciences Institute, 2000.