Power Control and Optimization in ICT

Giuseppe Iazeolla, Alessandra Pieroni

Department of Enterprise Engineering, Tor Vergata University, Via del Politecnico, 1 00133 Rome, Italy
Department of Enterprise Strategy and Applied Technology, Guglielmo Marconi University, Via Plinio, 44 00193 Rome, Italy

Abstract. The power management of ICT systems, i.e. data processing (Dp) and telecommunication (Tlc) systems, is becoming a relevant problem in economical terms. Dp systems totalize millions of servers and associated subsystems (processors, monitors, storage devices, etc.) all over the world, that need to be electrically powered. Dp systems are also used in the government of Tlc systems, which, besides requiring Dp electrical power, also require Tlc-specific power, both for mobile networks (with their of cell-phone towers and associated subsystems: base stations, subscriber stations, switching nodes etc.) and for wired networks (with their routers, gateways, switches etc.).
ICT research is thus expected to investigate into methods to reduce Dp and Tlc-specific power consumption. However, saving power may turn into waste of performance, in other words, into waste of ICT Quality of Service (QoS).
This paper investigates the Dp and Tlc power management policies that look at compromises between power-saving and QoS.

Keywords: ICT systems, data processing systems, server farms, telecommunication systems, energy management policies.
PACS: 07.05.Tp; 89.20.Ff; 88.05.Bc.

1. INTRODUCTION

The growth in ICT energy consumption is driven by the growth of demand for greater data processing (*Dp*) and larger access to telecommunications (*Tlc*), within almost every organization.

This growth has a number of important implications, including [3]:

- increased energy costs for business and government,
- increased emissions, including greenhouse gases, from electricity generation
- increased strain on the existing power grid to meet the increased electricity demand,
- increased capital costs for expansion of data center capacity and construction of new data centers,
- increased capital costs for expansion of wired and wireless access to communications.

Made 100 the total electrical power consumption for the ICT, around 14% is taken by the mobile *Tlc*,

74% by the wired *Tlc* and the remaining 12% by the *Dp* technology.

Dp, however, is only apparently the less powered sector, since *Tlc* is itself a *Dp* consumer, and so any effort to reduce the *Dp* power consumption may produce cascade effects that also reduce the *Tlc* one. Studying ways to save *Dp* power in thus central to any study for ICT power control and optimization. In the US, power absorbed by data centers is estimated in more than 100 Billion kW, for an expenditure of $ 8

Billions a year, that corresponds to the expenditure in electricity of about 17 Million homes [14].

This same US-local data-center problem becomes a global one when seeing at power consumption by web companies, say Google, Yahoo etc. The number of Google servers will reach an estimated 2,376,640 units by the end of 2013 [22].

Assuming a busy server absorbs around 240 W of power, Google will need about 600 MW of electrical power by the end of year 2013.

In *Tlc* systems, about 90% of *Tlc-specific* power consumption is concentrated in the routers. The links only absorb 10%. Current routers consume between 0.01 and 0.1 W/Mbps [8].

IT research is thus expected to investigate into methods to reduce power absorbed by*Dp* and *Tlc systems*. To do that, one may decide to adopt policies to periodically switch-*off Dp* servers *or Tlc* routers when they are in an *idle*-state.

Such policies, however, are to be sufficiently intelligent not to degrade the system Quality of Service (QoS). Indeed, returning an *off*-server or an *off* router to its *on* state requires spending a non-negligible amount of *setup*-time that makes the server or router slower to respond to customer requests. This may turn into low-quality services such as low response to web queries, unsatisfactory VoIP communications and streaming of data. Any research in power management should thus look at compromises between power-saving and QoS.

In this paper, Sect.2 studies *Dp* power management policies and Sect.3 *Tlc* power management policies.

2. POWER MANAGEMENT IN *DP* SYSTEMS

Data centers have become common and essential to the functioning of business, communications, academic, and governmental systems.

During the past years, increasing demand for *Dp* services has led to significant growth in the number of data centers, along with an estimated doubling in the energy used by servers and the power and cooling infrastructure that supports them.

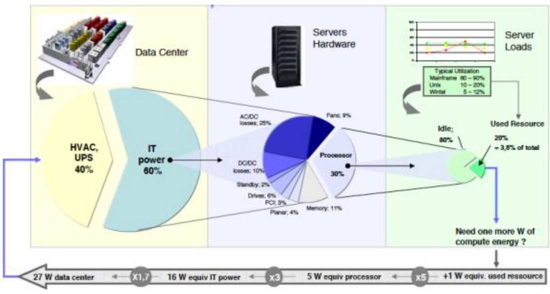


FIGURE 1. Data Center energy consumption sources [4].

Figure 1 illustrates the way energy is spent in data centers. Heating and services for Ventilation and Air Conditioning/Backup (HVAC/UPS) absorb around 40% of electrical energy and *Dp* services the remaining 60%. The latter is in turn divided between AC/DC losses (25%), DC/DC losses (10%), Fans, Drives, PCI, etc., and Memory consumptions (for a total 35%) and the remaining 30% is consumption in server processors. In other words, the processors consumption totalizes 0.30x0.60, i.e. 20% of total data center consumption. Such an amount, even though apparently negligible with respect to the total, is the main cause of the remaining 80%. Thus, any effort to reduce the processors 20% may produce cascade effects that also reduce the remaining 80%.

Figure 2 shows that a 1W savings at servers component level (processor, memory, hard disk, etc) creates a reduction in data center energy consumption of approximately 2.84W.

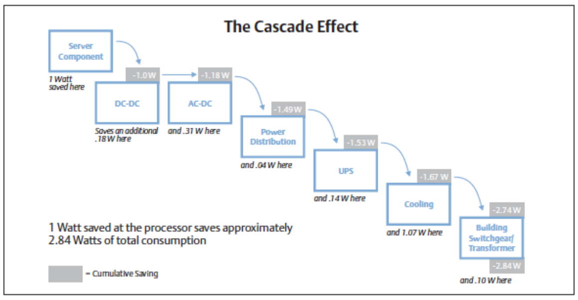


FIGURE 2. Cascade effect of energy saving in data centers [11].

For this reason, any research in ICT power saving should concentrate on policy to reduce *Dp* consumption at server components level.

Data centers can be seen as composed of a number of servers that can be organized into *single-farms* or a *multi-farms*. In the following, Sect.2.1 sees at power saving policies in the *single-farm* case, and Sect.2.2 at the *multi-farm* one.

2.1 Energy Saving in *Single-Farm* Data Centers

Most of power absorbed by the servers of a farm is wasted, since servers are *busy* (i.e. making processing work) only 20% to 30% of the time, on average. So, energy saving requires the adoption of management policies to avoid powering the servers when they are not processing. In other words, policies to decide in which state (*idle* or *off*) to keep the servers when not busy. Two types of server management policies will be considered: *static* and *dynamic* policies.

2.1.1 Energy Saving with Static Policies

One may assume a *busy*-server in the *on* state absorbs around 240W, an *idle*-server about 160W and an *off*-server 0W. So why not to keep in the *idle* state or in the *off* state the servers when not busy? Just since switching a server from *off* to *on* consumes a time-overhead. Thus, a power-saving policy may result in a time-wasting problem. As a consequence, the servers may lose performance (e.g. increased *response time* to the incoming jobs, lower throughput of communication packets, etc.) and its service may become unacceptable to customers.

To turn *on* an *off* server, we must first be put the server in *setup* mode. During the setup period, the server cannot process jobs. The time spent in setup is called *setup time*. In [7] the authors consider server farms with a setup cost. Setup costs always take the form of a time delay, and sometimes there is also a power penalty, since during that entire period the server consumes the same power as being in the *on* state.

In [7] three different policies are studied to manage server farms: *On/Idle* policy, *On/Off* policy and *On/Off/Stag* policy.

Under the *On/Idle* policy, servers are never turned *off*. All servers are either *on* or *idle*, and remain in the *idle* mode when there are no jobs to serve. Assume the farm consists of *n* servers, if an arrival finds a server *idle* it starts serving on the *idle* server. Arrival that find all *n* servers *on* (busy), join a central queue from which the servers pick jobs when they become *idle*.

The *On/Off* policy consists instead of immediately turning *off* the servers when not in use. As said above, however, there is a setup cost (in terms of time-delay and of additional power penalty) for turning *on* an *off* server.

Finally, the *On/Off/Stag* policy is the same as the *On/Off* one, except that at most 1 server can be in setup at any point of time. This policy is known as the "staggered boot up" policy in data centers, or "staggered spin up" in disk farms [7, 12, 13].

Figure 3a compares the *On/Off* and *On/Idle* policies for an example case.

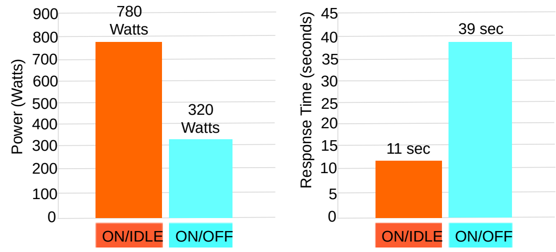


FIGURE 3a. Experimental results with 4 servers, utilization = 30% average setup time = 200sec, average job size (service time) =7sec [14].

The *On/Idle* policy proves to be better in terms of *response time*, because the incoming jobs do not suffer by *setup time* delays, but involves a larger amount of power waste with respect to the *On/Off* policy, since of the amount of power an *idle* server absorbs.

Figure 4 compares the three server management policies in a farm consisting of k=10 servers, when the average *setup time* changes from 1 to 100sec and the average processing load λ (i.e. average job arrival rate) from 1 to 7 job/sec. The mean job size (service time) is assumed of 1 sec.

Comparison is on the basis of the resulting mean response time E[T] to the incoming jobs and the average power consumption E[P].

In the *On/Idle* case, when λ is low, there is no waiting and thus the mean response time E[T] is of about the mean job service time (1sec) and increases for increasing λ.

A similar trend can be observed for the *On/Off/Stag* policy, since:

$$E[T]_{ON/OFF/STAG} = E[T]_{ON/IDLE} + E[\text{setup time}]$$

as shown in [7].

For the *On/Off* policy, instead, the response time curve follows a *bathtub* behavior.

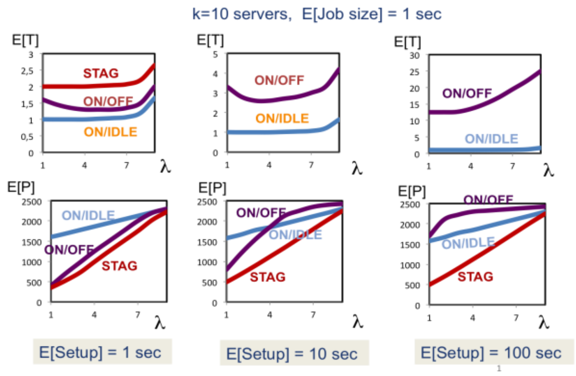


FIGURE 4. Effects of server management policies and *setup time* on response time and power consumption [14].

When the load λ is low, the mean response time is high since almost every arrival finds servers in the *off* state, and thus every job incurs the *setup time*. For medium loads, servers are less frequently switched to the *off* condition and thus jobs are more likely served by available servers in the *on* state, and do not incur in setup times. For high loads, finally, the mean response time increases due to large queueing in the system.

For the power consumption, one can show [7] that $E[P]_{ON/OFF/STAG} < E[P]_{ON/OFF}$, since at most one server can be in setup for the *On/Off/Stag* policy. There also results $E[P]_{ON/OFF} < E[P]_{ON/IDLE}$, since servers are turned *off* in the *On/Off* case. However, for loads λ above the medium, there results $E[P]_{ON/OFF} > E[P]_{ON/IDLE}$ for medium *setup time* i.e. E[setup time] = 10sec, while for large *setup time* i.e. E[setup time] = 100sec there always results $E[P]_{ON/OFF} > E[P]_{ON/IDLE}$, since of the large amount of power wasted in turning servers *on* in the *On/Off* policy.

TABLE 1. Synthetic view of the On/Off, On/Idle and On/Off/Stag power optimization policies

	Response Time	*Power Consumption*
On/Idle	*small response times*	*High waste of power*
On/Off	*Medium response times*	*Medium waste for low setup times, high for increasing setup*
On/Off/ Stag	*Large response times*	*Low waste of power*

Table 1 gives a synthetic comparison of the three considered policies in terms of *response time* and *power consumption.*

In conclusion, any reduction in power consumption is paid by an increase in response times. So, why not to adopt queueing disciplines that minimize average *response times*? The SPTF (Shortest Processing Time First) [25] or SJF (Shortest Job First)[24] queueing discipline is known to perform better that the common FIFO (First In First Out). Its use can then reduce the amount to pay in terms of *response time* to obtain a given power saving.

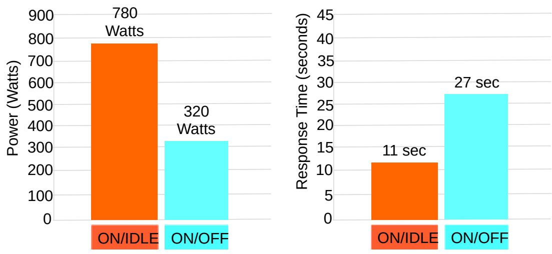


FIGURE 3b. Experimental results with same parameters of Fig. 3a except SPTF or SJF queueing discipline.

Figure 3a (that illustrates the FIFO queueing case) shows that to reduce the power consumption from 780 to 320W we have to pay an increase from 11 to 39 sec in average response time.

Figure 3b illustrates that if the SPTF discipline is instead used; the debt to pay in response time is much smaller (from 39 to 27 sec) as proved in our simulations studies [23].

2.1.2 Energy saving with dynamic policies

In any practical situation, the load λ changes over the time, according to a given pattern λ(t). One should then find policies that adapt themselves to changing load patterns. This is not the case of the policies introduced in Sect. 2.1.1, which are somehow static in nature and remain efficient only for given values of λ, while becoming inefficient for other values. Looking, for example, at the Figure 4 case with E[setup] = 10sec one can see that for changing values of λ there are situations in which the *On/Off* policy consumes less power than *On/Idle* and vice versa.

For this reason, two adaptive versions of the *On/Off* and *On/Idle* policies are known in literature, respectively called *DelayedOff* and *LookAhead*, which *dynamically* adapt themselves to changing loads [5].

The *DelayedOff*policy is an improvement of the *On/Off*. According to *DelayedOff*, when a server goes idle, rather than turning *off* immediately, it sets a timer of duration $\mathbf{t_{wait}}$ and sits in the idle state for $\mathbf{t_{wait}}$ seconds. If a request arrives at the server during these $\mathbf{t_{wait}}$ seconds, the server goes back to the *busy* state (with zero setup cost); otherwise, the server is turned *off*.

The *LookAhead* policy is an improvement of the *On/Idle*. Under such a policy, the system fixes an optimally chosen number *n** of servers maintained in the *on* or *idle* states. According to the standard *On/Idle*, if an arrival finds a server *idle* it starts serving on the *idle* server. Arrivals that find all *n** server *on* (busy), join a central queue from which servers pick jobs when they become *idle*.

The optimal t_{wait} and the optimal *n** of the two policies respectively are chosen to minimize a new metric called *ERP (Energy-Response time Product)* and defined as:

$$ERP = E[P]\ E[T].$$

Minimizing *ERP* can be seen as maximizing the performance per Watt, with performance been defined as the inverse of the mean response time [5].

In the *LookAhead*policy, *n** changes as a function of time. Indeed, the policy calculates *n*(t)* for each time t basing on the forecast of the load $\lambda(t)$ at time t.

Figure 5illustrates the auto-scaling capabilities of the *LookAhead* and *DelayedOff* policies [5], with respect to the conventional *On/Off*, for poissonian arrivals with $\lambda(t)$ changing sinusoidally with time (period = 6 hrs).

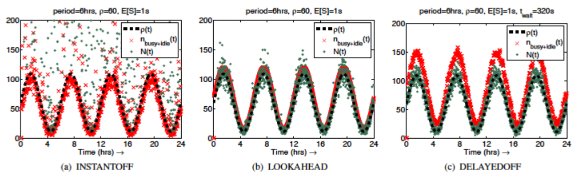


FIGURE 5. Effects of dynamic policies with respect to the static On/Off [5].

The left graph refers to the *On/Off* policy (called *InstantOff*), the middle graph to the *LookAhead* and the right one to the *DelayedOff*. The dashed line denotes the varying load at time t, $\lambda(t)$. The crosses denote the number $\mathbf{n_{busy+idle}}(t)$ of servers that are busy or idle at time t, and the dots denote the number N(t) of jobs in the system at time t.

The illustration shows how, with the two dynamic policies, number $\mathbf{n_{busy+idle}}(t)$ and number N(t) almost completely follow the behavior of the demand pattern $\lambda(t)$ while in the *On/Off* case such numbers are somewhat dispersed, in other words, some servers remain in the *idle* state whereas they should be *busy* and vice versa, with the consequence of waste of power and worsened *response time*.

The two dynamic policies above simply try to optimize the E[P] by E[T] product.

In many practical situations, instead, the objective is to meet a given average response time, according to requirements dictated by specific Service Level Agreements (SLAs).

In this case, specific dynamic policies have been introduced, which try to respect the E[T] requirement while minimizing the average power consumed by the servers ($\mathbf{P}_{avg}$) and the average number of used servers ($\mathbf{N}_{avg}$).

Such policies are known as the *AutoScale* policy [6],the *AlwaysOn* policy [15], the *Reactive* policy [16] and the *Predictive MWA* policy [17, 18, 19, 20]. The latter will not be dealt with here, and we will only treat the *AutoScale* policy, which is an evolution of the remaining three.

The *AutoScale* policy generalizes the use of the $\mathbf{t_{wait}}$ time already seen for the *DelayedOff* policy. Differently from this latter, however, in the *AutoScale* case each server decides autonomously when to turn off, setting a timer of duration $\mathbf{t_{wait}}$ and sitting in the *idle* state for $\mathbf{t_{wait}}$ sec. As with the *DelayedOff*, however, if a request arrives at the server during these $\mathbf{t_{wait}}$ sec, then the server goes back to the *busy* state (with zero setup cost). Otherwise, the server is turned *off*.

The *AutoScale* and the three remaining policies have been evaluated in [6] according to a specific load pattern λ(t) varying over time between 0 and 800req/s (see Figure 6). Such a pattern, known as *Dual Phase* pattern is used to represent the diurnal nature of typical data center traffic, where the request rate is low at the night time and high at day time.

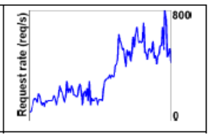
Dual phase	NLANR [nlanr 1995]	

FIGURE 6. Dual Phase pattern [6].

Figure 7 illustrates the performance of the *AutoScale* policy, when the *time requirement* to meet is a 95- percentile *response time* T goal of 400 to 500ms (denoted T_{95}). In the illustration, the red lines denote the number $k_{busy+idle+setup}(t)$ of *busy+idle+setup* servers and the blue lines the number $k_{busy+idle}(t)$ of *busy+idle* servers at time t. The k_{ideal} line represents the number of servers that should be *on* at any given time to fully satisfy the demand $\lambda(t)$.

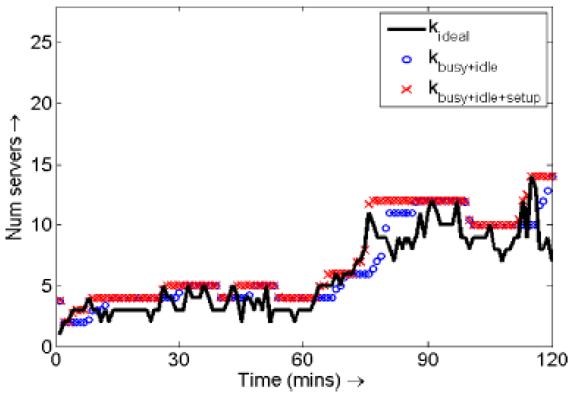
The illustration shows how, with the *AutoScale* policy, there is no dispersion in the available servers and the number $k_{busy+idle+setup}(t)$ and number $k_{busy+idle}(t)$ almost totally follow the demand pattern $\lambda(t)$ and a T_{95} = 491ms goal is achieved, with P_{avg}= 1,297W and N_{avg} = 7.2 servers.

In the mentioned similar policies (*AlwaysOn*, *Reactive* and *Predictive MWA*), instead, the T_{95} requirement can be seen met only at the expense of server dispersion and/or at the expense of P_{avg} and N_{avg} [6].

Indeed, in the *AlwaysOn* case the T_{95} requirement is met (T_{95}=291ms) but at the expense of large dispersion of servers and large power consumption (P_{avg}=2,322W and N_{avg}=14).

In the *Reactive* case, instead, a low dispersion of servers is achieved, with low power and low number of servers (P_{avg}=1,281W, N_{avg}=6.2) but the time requirement is absolutely out of range (T_{95} = 11,003ms).

A better time performance (T_{95} = 7,740ms) is found in the *Predictive MWA* with similarly low dispersion of servers, and similarly low power and number of servers (P_{avg}= 1,276W, N_{avg}= 6.3).



T_{95}=491ms, P_{avg}=1,297W, N_{avg}=7.2

FIGURE 7. Effects of dynamic *AutoScale* policy [6].

2.2 Energy Saving in *Multi-Farm* Data Centers

Energy saving in multi farms is based on so-called *self-organization* and *self-differentiation* algorithms, whose goal is to transfer the load from a server to a less loaded one, to maximize the power efficiency of the whole data center.

These algorithms are widely adopted in the Autonomic Computing field. The term Autonomic indicates systems able to self-manage, self-configure, self-protect, and self-repair, thus systems that have no need of external action to be managed [2].

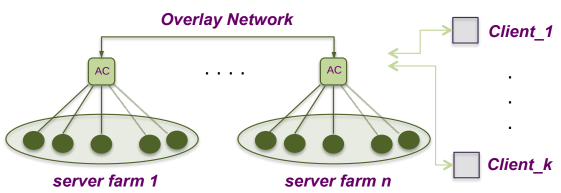


FIGURE 8. Example of a multi-farm data center.

Figure 8 illustrates the typical multi-farm architecture, that consists of a series of server farms (1 through n) each farm controlled by a so-called Autonomic Component (AC), with the ACs interacting through an Overlay Network (ON). Each farm serves a number of Clients (1 trough k).

The ON is a self-organized network, in other words a network which is created, maintained and optimized through *self-organization algorithms* which cluster the ACs according their properties or type [1].

The ACs, in turn, execute a particular kind of self-organization algorithm called *self-differentiation algorithm*, which take decentralized decisions on the state and configuration of the ACs.

The AC aims at putting in *idle* state the servers and transferring the load on the other servers to limit performance degradation. Three types of self-differentiation algorithms are known: *Stand-by*, *Load Distribution* and *Wake-up* algorithms whose details can be found in [1].

The algorithms were evaluated by means of simulations of an use-case in which server farms are in charge to serve requests issued by a set of clients. Each client performs several requests, before terminating the connection. The percentage of energy that can be saved in a day goes from about 7% to about 12%, with a debt to pay in terms of *response time* from about 9 units of time (when the power saving is 7%) to about 11 units of time (when the power saving is 12%).

3. ENERGY MANAGEMENT IN *TLC* SYSTEMS

Tlc systems may consist of wired or wireless access networks or of a combination thereof.

In addition to the basic *Dp* infrastructure, *Tlc* systems also include *Tlc-specific* subsystems: cell-phone, towers with associated base stations, subscriber stations, switching nodes, etc., for the wireless part, and communication processors, routers, gateways, switches, etc., for the wired part.

Power management in *Tlc* systems, thus, includes not only power optimization of their *Dp* infrastructure, but also power optimization of *Tlc-specific* subsystems.

In this Section we will only deal with *Tlc-specific* subsystems, since the power optimization of *Dp* infrastructure is dealt with as already seen in Sect. 2.

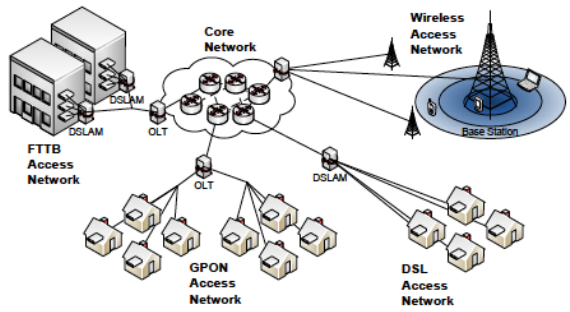


FIGURE 9. Typical *Tlc* network architecture [8].

Figure 9 describes a typical *Tlc* architecture, which combines wired and wireless communication networks.

In the *wired* part, three main types of connections are found: 1) the twisted pair copper cable connection based on the DSL (Digital Subscriber Line) technology; 2) the coax cable connection based on the DOCS (Data Over Cable Service) technology and 3) the optical fiber connection based on the GPON (Gigabit Passive Optical Network) technology, used when higher bit rates are required. The illustration also shows the DSLAM (DSL Access Multiplexer) nodes, the OLT (Optical Line Termination) nodes and the FTTB (Fiber to the Building) nodes.

In order to interconnect different user areas, a *core network* is used, that consists of a number of core nodes that are interconnected through wavelength-division multiplexed (WDM) optical fiber links, usually in a mesh or ring topology.

In the *wireless* part of the network we find *base stations* (BS) to which the user's devices are connected by means of radio signals. Each BS is further connected to the core network through a so-called *backhaul network*. Different technologies can be found, from WiMAX (Worldwide Interoperability for Microwave Access), to HSPA (High Speed Packet Access), and to the most recent LTE (Long Term Evolution).

In such a system, about 90% of *Tlc-specific* power consumption is concentrated in the routers (with 75% the line cards, 10% the power supply and fans, and 10% the switch fabric) [8]. Current routers consume between 0.01 and 0.1 W/Mbps. One can calculate that at ADSL access rates (8 Mbps) the power absorbed per subscriber is of about 0.24W/subs, while at 100Mbps becomes of about 3W/subs [8].

Currently, *Tlc* networks are designed to handle the *peak* loads. Designing *adaptable networks*, where one can switch *off* line cards when the demand is lower, can lead to lower power consuming networks.

In core networks this can be achieved by use of *dynamic topology optimization* algorithms: from all possible topologies that satisfy the required traffic demand, the topologies with lower overall power consumption are chosen. By such algorithms, reductions of power consumption for more than 50% during off-peak hours can be achieved [21].

Base stations (BS) with differentiated cell sizes are the key in wireless networks optimization if the so called *hybrid hierarchical BS deployment* is used.

A low layer access network is first created, providing a low bit rate (but large cell sizes) to the users. In the higher layers, BS with higher bit rates (but smaller cell sizes) is utilized to provide high bandwidth connections when required. The advantage is that the higher layers can be switched to the *idle*, and only switched *on* with high traffic demand.

Tlc power optimization, also tries to minimize the power consumption of the home gateways. These are individual devices that only need to be *on* when the user is active. At other times, they could be switched *off*. In reality this is rarely operated, but legislations concerning standby power consumption standards of 0.5 W are emerging [8].

4. CONCLUSIONS

The power management of ICT systems, i.e. data processing (*Dp*) and telecommunication (*Tlc*) systems, is a complex issue with implications in economical terms.

The paper has illustrated methods to optimize *Dp* power consumption by use of power management policies (static and dynamic policies) that yield electrical power-saving while maintaining the system QoS at acceptable levels.

The paper has also illustrated methods to optimize *Tlc* power consumption by use of power management policies to be adopted in wired and wireless *Tlc* systems. This achieves electrical power saving without compromising the service quality.

Acknowledgements

This work was partially supported by the *PhD research program* of the University of Roma TorVergata funded by Telecom Italia, and by the Guglielmo Marconi University of Studies in Roma.

References

1. S. Sahuquillo Martínez, J.S. Pareta, B. Otero, L. Ferrari, A. Manzalini, C. Moiso, "Self-Organized Server Farms for Energy Savings", ICAC-INDST'09, June 16, 2009, Barcelona, Spain.
2. D. Barbagallo, E. Di Nitto, D.J. Dubois, and R. Mirandola, "A Bio-Inspired Algorithm for Energy Optimization in a Self-organizing Data Center", SOAR'09 Proceedings of the First international conference on Self-organizing architectures, 127-151.
3. U.S. Environmental Protection Agency ENERGY STAR Program, "Report to Congress on Server and Data Center Energy Efficiency Public Law 109-431", August 2, 2007.
4. F. Renzi, "Business E^3: Energy, Efficiency, Economy", 2007 IBM Corporation.
5. A. Gandhi, V. Gupta, M. Harchol-Balter, M.A. Kozuch, "Optimality Analysis of Energy-Performance Trade-off for Server Farm Management", Performance Evaluation (2010), ScienceDirect, Elsevier, 1-23.
6. A. Gandhi, V. Gupta, M. Harchol-Balter, R. Raghunathan, M.A. Kozuch, "AutoScale: Dynamic, Robust Capacity Management for Multi-Tier Data Centers", ACM Transactions on Computer Systems, Nov. 2012, Vol. 30, No. 4, Article 14.
7. A. Gandhi, M. Harchol-Balter, I. Adan, "Server farms with setup costs", Performance Evaluation (2013), ScienceDirect, Elsevier, 1–22.
8. W. Vereecken, W.VanHeddeghem, et al., "Power Consumption in Telecommunication Networks: Overview and Reduction Strategies", IEEE Communication Magazine 2010, vol.4 issue 6.
9. M. Meo, M. Mellia, "Green Networking", 2010 IEEE INFOCOM Workshop on Communications and Control for Sustainable Energy Systems, Orlando Florida.

10. M.A. Marsan, “Energy-Efficient Networking: the views of leaders of international research projects”, CCW 2011, Hyannis, Oct. 2011.
11. A White Paper from the Experts in Business-Critical Continuity, “Energy Logic: Reducing Data Center Energy Consumption by Creating Savings that Cascade Across Systems”, 2012 Emerson Network Power.
12. I. Corporation, “Serial ATA Staggered Spin-Up”, White paper Sept. 2004.
13. M. W. Storer, K. M. Greenan, E. L. Miller, K. Voruganti, “Pergamum: replacing tape with energy efficient, reliable, disk-based archival storage”, FAST’08, CA, USA, 2008, pp. 1–16.
14. M. Harchol-Balter, “Open Problems in Power Management of Data Centers”, IFIP WG7.3 Workshop, University of Namur, Belgium Nov. 20, 2010.
15. G. Chen, W. He, J. Liu, et al., “Energy-aware server provisioning and load dispatching for connection-intensive internet services”, 2008, Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation (NSDI’08), 337–350
16. B. Urgaonkar and A. Chandra, “Dynamic provisioning of multi-tier internet applications”, 2005, Proceedings of the 2nd International Conference on Automatic Computing (ICAC’05), 217–228
17. P. Bodik, R. Griffith, et al., “Statistical machine learning makes automatic control practical for internet datacenters”, 2009, Proceedings of the 2009 Conference on Hot Topics in Cloud Computing (HotCloud’09).
18. D. Grunwald, C.B. Morrey III, et al., “Policies for dynamic clock scheduling”, Proceedings of the 4th Conference on Symposium of Operating System Design and Implementation (OSDI’00).
19. T. Pering, T. Burd and R. Brodersen, 1998, ”The simulation and evaluation of dynamic voltage scaling algorithms. In Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED’98) 76–81
20. A. Verma, G. Dasgupta, et al., “Server workload analysis for power minimization using consolidation”, Proceedings of the 2009 Conference on USENIX Annual Technical Conference (USENIX’09)
21. B. Puype, W. Vereecken, D. Colle, M. Pickavet, and P. Demeester, “Power reduction techniques in multilayer traffic engineering,” in Transparent Optical Networks, ICTON ’09. 11th International Conference On, July 2009, pp. 1–4.
22. J. Dean, “Designs, Lessons and Advice from Building Large Distributed Systems”, 2009 Google Fellow.
23. G. Iazeolla, A. Pieroni, G. Scorzini, “Simulation study of server farms power optimization”, RI.01.13 T.R., Software Engineering Lab., University of Roma TorVergata, Jan. 2013.
24. MorHarcol-Balter, “Performance Modeling and Design of Computer Systems”, Cambridge University Press, 2013.
25. R.W. Conway, W. L. Maxwell, L.W. Miller, “Theory of scheduling”, Addison-Wesley, 1967.