Journal Code:	Article ID	Dispatch: 18.11.12	CE: Sandy Cheromyl Vilan
C P E	2 9 6 4	No. of Pages: 17	ME:

CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE
Concurrency Computat.: Pract. Exper. (2012)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/cpe.2964

SPECIAL ISSUE PAPER

Control system for reducing energy consumption in backbone computer network

Ewa Niewiadomska-Szynkiewicz[1,2,*,†], Andrzej Sikora[1,2], Piotr Arabas[1,2] and Joanna Kołodziej[3]

[1]*Institute of Control and Computation Engineering, Warsaw University of Technology, Nowowiejska 15/19, 06-665, Warsaw, Poland* Q1
[2]*Research and Academic Computer Network, Warsaw University of Technology, Wawozowa 18, 02-769, Warsaw, Poland*
[3]*Institute of Computer Science, Cracow University of Technology, Warszawska 24, 31-155 Cracow, Poland*

SUMMARY

Network optimization concerned with operational traffic management in existing data networks is typically oriented towards either maximizing throughput in congested networks while providing for adequate transmission quality, or towards balancing the traffic so as to maintain possibly large free capacity for carrying additional (new) traffic. Nowadays, the reduction of power consumption is a new key aspect in the development of modern wired networks. Power management capabilities allow modulating the energy consumption of devices that form a network by putting them into standby state, or by decreasing their performance in case of low incoming traffic volume. This paper presents a framework for backbone network management, which leads to the minimization of the energy used by this network. The policy for dynamic power management of the whole network through energy-aware routing, traffic engineering, and network equipment activity control is introduced and discussed. The concept of the system is to achieve the desired trade-off between total power consumption and the network performance according to the current load, incoming traffic, and user requirements. The effectiveness of our framework is illustrated by means of a numerical study. Copyright © 2012 John Wiley & Sons, Ltd.

Received 15 June 2012; Revised 18 September 2012; Accepted 31 October 2012

KEY WORDS: green computer network; energy-aware network; traffic engineering; optimization

1. INTRODUCTION

The optimization of energy consumption in computer networks has been recently an important research issue in green networking and computing. According to different studies and reports, the information and communication technology sector belongs to the group of big power consumers. We can indicate two main motivations that drive the quest for green networking.

- Environmental, related to the reduction of impact on CO_2 emissions and wastes.
- Economic, stemming from the need of telecoms and service providers to reduce their operational costs.

On the other hand, to support new generation network infrastructures and services, network operators need a larger number of sophisticated devices able to perform more complex operations in a scalable way. Therefore, it is recognized that energy awareness is an important part of the modern

*Correspondence to: Ewa Niewiadomska-Szynkiewicz, Warsaw University of Technology, Nowowiejska 15/19, 06-665 Warsaw, Poland.
†E-mail: ens@ia.pw.edu.pl

network design and management. The main challenge is to design, develop, and test novel technologies and architectures able to reduce energy requirements, control mechanisms and strategies for network equipment enabling energy saving, by adapting network capacities and resources to the actual traffic load and demands, while ensuring end-to-end quality of service (QoS).

Nowadays, a broad spectrum of activities in green networking have been undertaken both in the industry and research domain. In general, the idea is to reduce the gap between the capacity provided by a network and the requirements of users, especially during low traffic periods. In some cases, the energy utilized by the network may be minimized by inactivation (switching off) of idle devices (routers, line cards, and communication ports) and by reduction of the speed of link transfers. It is clear that instead of balancing traffic, during energy saving periods, it will be required to concentrate data transfers along as few routes as possible and to switch off as many energy consuming components as feasible. Because typical backbone networks are overprovisioned, selectively shutting down links in periods of low demand seems to be a good solution for reduction of the energy consumption.

The ECONET consortium has developed a general framework for backbone network management that leads to the minimization of the energy utilized by a network, [1]. Our implementation of this framework provides local and central strategies and algorithms together with simple interfaces for exchanging data among network devices that execute the decisions of control units. In this work, we mainly focus on algorithms for the centralized network-wide control. The energy optimization is formulated as a mathematical programming problem with various constraints and control parameters. We start from the complete problem stated in terms of binary variables, which we try to solve using the branch-and-bound method. Next, we provide simplifications that finally lead to a heuristic method. In our formulations, all possible energy saving decisions are directly specified, together with decisions concerning traffic assignment to particular links. In general, the idea is to concentrate network traffic on a minimal subset of network components.

The remainder of this paper is organized as follows. The energy consumption sources in a network, models for power management, and common approaches to decrease energy demands are analyzed in Section 2. In Section 3, we present our framework for power control. In Section 4, we investigate solutions for energy-aware networks described in the literature. Two formulations of network-wide optimization of energy saving and its relaxation based on a heuristic approach are presented in Section 5. The results of simulations and discussion of the performance of proposed control strategies are described in Section 6. We conclude the paper in Section 7.

2. ENERGY CONSUMPTION IN COMPUTER NETWORKS

Before the discussion of energy-aware network design, we analyze the energy consumption characteristics of the network equipment. Each network node is a multi-chassis device which is composed of many entities, that is, processor, chassis, line cards, communication ports, power supply, fans, and so on. In our approach, we provide a hierarchical view of the internal organization of the device. It is represented through several layers, namely, a device itself (network node), chassis, cards, ports, and other subcomponents. Each component is energy powered.

In general, power and energy management methodologies in networks and distributed computing environments can be classified into two main categories, namely (i) *static energy management* and (ii) *dynamic energy management* technologies [2, 3]. The methodologies from both classes can operate on the hardware and software levels. In the static mode at the hardware level, the energy can be optimized by using low-power devices or nano-processors. In the dynamic case, the power supply of the network devices can be modulated (*dynamic voltage and frequency scaling* method [4, 5]) or the idle devices can be deactivated. The detailed taxonomy of the energy and power management in highly parametrized distributed environments can be found in [6–9] and [10].

In the network built of standard devices, the energy usage is relatively constant regardless of network traffic. This is because of the fact that most of the currently available network equipment does not implement any energy saving mechanisms. However, recently various efforts have been undertaken to develop prototypes of energy-aware (‘green’) network devices (see [1]) that can

DOI: 10.1002/cpe

operate in different modes, which differ in the power consumption, and implement dynamic power management techniques. Two methods are commonly used to reduce energy requirements of network devices.

- *Smart standby*-the capability of automatically switching off the whole device or its component when there is no data to transmit.
- *Dynamic power scaling*-the capability of decreasing the energy demands of a given network device by changing its performance.

Adaptive rate and low power idle (LPI) are two common techniques of power scaling approaches. The adaptive rate method allows scaling the processing capacity of a given device or the transmission or reception speed of the network interface. The LPI method puts a given device into low power state during short inactivity periods. Most personal computers implement both AR and LPI techniques. The Advanced Configuration and Power Interface (ACPI) specification described in [11] defines a number of EAS (EAS) attained via voltage and clock frequency scaling and idle states in which the processor is in the standby mode. The 802.3az standard [12] defines the implementation of LPI for Ethernet interfaces. It must be noted that apart from 802.3az and some Linux-based [13] or FPGA [14] routers, real green network devices are not commonly used, and all results concerning power consumption are purely theoretic, usually based on analogous performance of personal computers with ACPI. However, numerous formal models that describe the correlation between an amount of transmitted data and energy consumption are provided in the literature (see [15]). The function describing this characteristic is typically considered nonlinear [16]. To simplify a model, various approximations are proposed, that is, linear [16], polynomial [13], or stepwise function [17].

3. FRAMEWORK FOR LOW ENERGY CONSUMPTION NETWORKS

It is viable that new network equipment will be more energy effective because of modern hardware technology including moving more tasks to optical devices. In addition, even greater energy savings may be obtained by employing energy-aware traffic management and modulating the energy consumption of network devices. Various control strategies can be applied to reduce power utilization. The energy efficient configuration of each device can be calculated locally due to the incoming traffic load measured by its interfaces. Another approach is to calculate decisions concerning operation of network equipment based on the global data about the network topology, load measurements, and in some cases expected demands. The network-wide control strategies (i.e., routing and traffic engineering) give the possibility of moving traffic load among network nodes in such a way that some devices can be put in standby mode (see Figure 1). L. Chiaraviglio *et al.* suggest in [18] that the greatest energy savings may be obtained by selective shutdown of network nodes (routers).

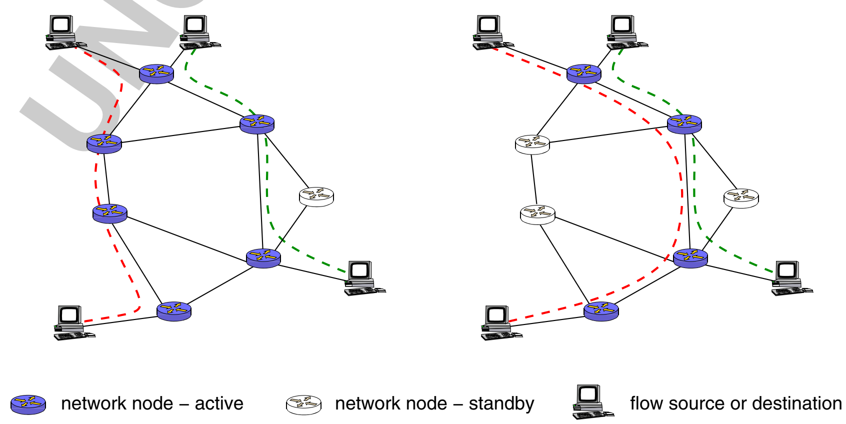


Figure 1. Operation of the local and network-wide control strategies.

Color Online, B&W in Print

DOI: 10.1002/cpe

Color Online, B&W in Print

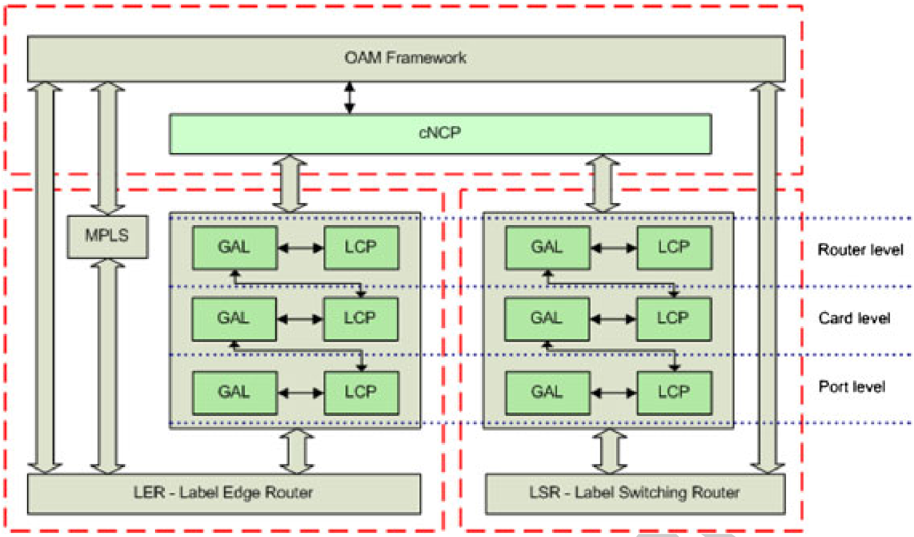


Figure 2. The architecture of the control framework.

Unfortunately, it is often impractical for real applications—in most well-designed networks, it is assumed that all routers are constantly used. However, we can shutdown selected links, or control their current throughput, and at the same time their consumed energy.

A general control framework for a high performance energy-aware backbone network has been developed in the ECONET project, [1]. The smart standby and dynamic power scaling techniques are employed in this system. Our implementation of this framework provides a centralized variant of network-wide control. The concept of the architecture of our framework is presented in Figure 2. It consists of four main components.

F2

- **OAM**. Monitoring and operation administration and management that provides monitoring of the network and plays the role of middleware between all components of the network. OAM should support Open Shortest Path First and Multiprotocol Label Switching Traffic Engineering (MPLS TE) protocols.
- **cNCP**. Central network-wide control policy. The decision process, whose goal is to optimize the network performance to reduce power usage. The optimization problem is formulated and solved for a given network, taking into account its topology and expected requirements of users. The outcome of cNCP depends on the implementation. Two variants are considered: variant 1—the routing tables for the MPLS protocol are provided; variant 2—the routing tables and suggested power status of network devices are sent to adequate Local control policies (LCPs).
- **LCP**. Two options of the LCP operation can be employed, owing to the implemented variant of cNCP. In the case of variant 1, the objective of the LCP is to optimize the configuration of each component of a given device in order to achieve the desired trade-off between energy consumption and performance according to the incoming traffic load measured by the OAM framework. In the case of variant 2, the operation of each LCP module is reduced to realize the decisions calculated by cNCP, by taking into account constraints related to the current local load and incoming traffic. The outcome of the LCP component is calculated based on the capability of energy-aware elements, incoming traffic, and the knowledge about the internal architecture of a given device.
- **GAL**. Green abstraction layer. The GAL is the standard interface between monitoring layer, control plane layer, and hardware for exchanging data regarding the power status of the device and its components. The objective is to hide the implementation details of energy saving techniques, as well as to provide standard interfaces between control and monitoring frameworks, and energy-aware technologies. Because of the internal architectural complexity of network equipment, the GAL is based on hierarchical decomposition. Each layer provides an abstraction

DOI: 10.1002/cpe

of internal components of the device. The GAL transforms the outcome of LCPs into the power-management configuration of a given component of the device (e.g., switching off a given router or card or switching a given port into an energy-aware state). The detailed description of the GAL component is provided in [19].

In this paper, we focus on network-wide control policies for optimal management of energy resources in computer networks. Various energy saving optimization problems can be implemented in the cNCP layer of the control framework depicted in Figure 2. A brief survey of techniques described in the literature is presented in the next section. Then, we provide our formulations of the optimization problem and efficient algorithms to solve them.

4. NETWORK-WIDE CONTROL—RELATED WORK

Recent studies obtained with real data from service providers suggest that network-wide control strategies could significantly decrease energy consumption of a network [20, 21]. Typical telecommunication networks, specifically in the core part have redundant links, hence, it is obvious that, at least in the periods of low traffic, some links could be switched off or operate at lower rate. An additional source of redundancy is constituted by bundled links, that is, links composed of a number of fibers to multiply their bandwidth. The common approach to dynamic power management is to formulate an optimization problem similar to the network design problem [22] or QoS provisioning task [23, 24] but with a cost function defined as the sum of energy consumed by all components of the network. Such formulations are proposed in [18, 25, 26] and [27]. L. Chiaraviglio *et al.* [18] provide an integer linear programming formulation to find network nodes and links that can be switched off. J. Chabarek *et al.* in [25] propose reducing the power consumption by finding links and line cards that can be switched off for a particular traffic matrix by solving a large mixed-integer linear problem. A similar approach can be found in [28], where scaling of link rates is attained by selectively switching on and off fibers composing them. The optimization of a two level structure with an IP network built over optical equipment is covered in [29]. Recently, Kołodziej *et al.* [3, 30] have provided the comparative empirical analysis of the effectiveness of both network management strategies with switching off the idle devices and modulation of the power supply in computational grids.

It must be noted that the optimization of an energy-aware network is much more difficult to solve than the typical shortest path calculation. The selection of paths is not independent and, on the contrary, should be aggregated allowing to move traffic out of some links and switch them off. Moreover, the energy consumption functions are often non-convex because of the presence of integer variables. As a result, the fully formulated problem is *NP*-complete, relaxing some constraints introduces suboptimality or, for example, instability [17] of the system.

To speed up calculations, various heuristics are typically constructed. They incorporate solving simpler (usually relaxed) mathematic programming tasks for preselected conditions—usually for a set of active links or paths—see [28, 31] and [32, 33], respectively. Apart from using heuristics, it is sometimes possible to aggregate nodes and demands, which allows solving the defined task in reasonable time; however, additional effort is needed to de-aggregate results [18].

All the aforementioned approaches may be employed in the centralized control scheme; however, for reliability and scalability, telecommunication networks tend to use distributed logic. Distributed energy-aware mechanisms are typically built as extensions of existing routing protocols—for example, Open Shortest Path First [34–36], BGP, and MPLS. Extension of the signaling infrastructure with green technologies allows to partially overcome the absence of the traffic matrix [31]—the past state of the network can be used to compute flows [35]. It is important to note that most of the cited distributed mechanisms use relatively simple heuristics. Decomposition of the previously described mathematical programming tasks is difficult because of complexity of interconnections and properties of performance indices (e.g., nonconvexity). Q2

Many authors propose to use multi-path routing, which is typically avoided in real networks. They also assume a relatively simple model of network equipment and assume only two states—active

DOI: 10.1002/cpe

and switched off, [18]. The objective of works conducted in recent projects on green networking is to develop devices, which can switch among several EAS, and model their internal structure including dependencies among subcomponents [1, 37]. Incorporating these features into a mathematical programming task involves the introduction of a large number of integer variables and the difficulty to separate constraints. Examples of two-level systems are presented in [25, 29]. In [28], the authors use the partial deactivation of bundled links to scale their capacities.

In this work, we propose a centralized network-wide control policy that calculates the optimal network performance based on known network topology and expected demands. In our formulation, each router, line card, and port can operate in different modes, which differ in the power consumption. The goal is to reduce the energy usage by putting selected network equipment in low energy states.

5. NETWORK-WIDE CONTROL PROBLEM FORMULATION

We consider a network formed by the routers labeled with $r = 1, 2, \ldots, R$; S_R is the set consisting of all routers. They are classified into two groups—edge and transit routers. We provide the following hierarchical representation of a router. Each router is equipped with cards labeled with $c = 1, 2, \ldots, C$, and the card can contain a number of communication ports. S_C denotes the set of cards in the whole network. Direct links labeled with $e = 1, 2, \ldots, E$ link all pairs of ports from different routers and cards. S_E denotes the set of all links in the network. All listed elements, that is, routers, cards, and ports can operate in different EAS that are related to the application of power scaling and standby techniques. We distinguish active, sleeping, and switched-off states of a given device; see Figure 3. The EAS labeled with $k = 1, 2, \ldots, K$ are defined as power settings. We assume that at a given time, instant two ports connected by the eth link are in the same state k. The power consumption ξ_{ek} associated with state k depends on the current throughput. In our approach, the power profile model is defined by a stepwise function describing power consumption because of a given throughput; see Figure 3. The throughput of link e in state k is defined as M_{ek}. A set S_K contains all provided EAS. Moreover, we assume fixed power levels W_c and T_r associated to cards and routers, respectively.

F3

All demands imposed on the network are labeled with $d = 1, 2, \ldots, D$. They are transmitted by means of flows allocated to given IP/MPLS paths under QoS requirements. Two nodes, the port of the source node (s_d) and the port of the destination node (t_d) are associated with each demand. The volume of demand d is equal to h_d. The set of paths $P(d)$ for each demand d is defined as the set of all paths between nodes s_d and t_d. In our formulation, we assume that demands between edge routers are known.

Given the presented notation, we provide various formulations of the energy saving optimization problem. We start from the complete problem formulation with discrete variables that can be solved only for small-size networks. Next, we simplify this formulation providing predefined paths. Finally, we present the efficient heuristic for relaxation and transformation of the basic discrete formulation into a continuous one.

Color Online, B&W in Print

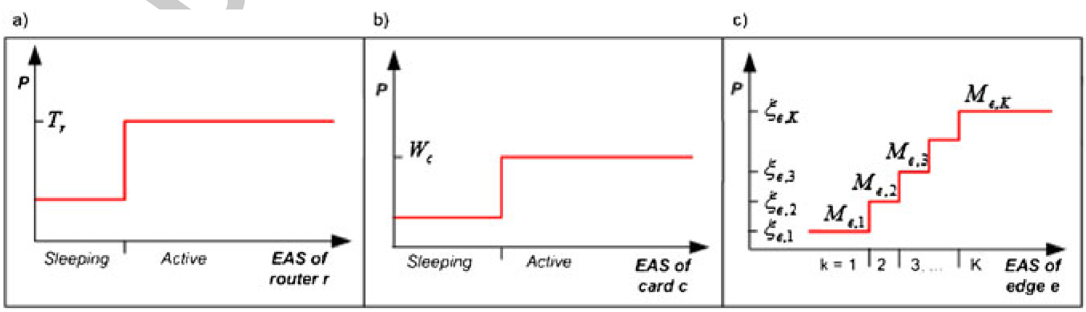


Figure 3. The energy-aware states (EAS)—router, card, and link.

DOI: 10.1002/cpe

5.1. Basic formulation: link node problem (LNP)

The energy-aware network management problem can be formulated as an integer linear optimization problem. We begin with *link-node* formulation assuming full routing calculation and energy state assignment to all links in a network. In our formulation, the total power utilized in the system for finalizing all network operations is assumed to be minimized.

$$\text{minimize}_{x_c, y_{ek}, z_r, u_{ed}} \left\{ F_{LN} = \sum_{e=1}^{E} \sum_{k=1}^{K} \xi_{ek} y_{ek} + \sum_{c=1}^{C} W_c x_c + \sum_{r=1}^{R} T_r z_r \right\}, \tag{1}$$

subject to the constraints

$$\forall_{\substack{d=1,\dots,D,\\ c=1,\dots,C}} \quad \sum_{p=1}^{P} l_{cp} \sum_{e=1}^{E} a_{ep} u_{ed} \leqslant x_c, \tag{2}$$

$$\forall_{\substack{d=1,\dots,D,\\ c=1,\dots,C}} \quad \sum_{p=1}^{P} l_{cp} \sum_{e=1}^{E} b_{ep} u_{ed} \leqslant x_c, \tag{3}$$

$$\forall_{\substack{r=1,\dots,R,\\ c=1,\dots,C}} \quad g_{rc} x_c \leqslant z_r, \tag{4}$$

$$\forall_{e=1,\dots,E} \quad \sum_{k=1}^{K} y_{ek} \leqslant 1, \tag{5}$$

$$\forall_{\substack{d=1,\dots,D,\\ p=s_d}} \quad \sum_{e=1}^{E} a_{ep} u_{ed} - \sum_{e=1}^{E} b_{ep} u_{ed} = 1, \tag{6}$$

$$\forall_{\substack{d=1,\dots,D,\\ p\neq t_d, p\neq s_d}} \quad \sum_{e=1}^{E} a_{ep} u_{ed} - \sum_{e=1}^{E} b_{ep} u_{ed} = 0, \tag{7}$$

$$\forall_{\substack{d=1,\dots,D,\\ p=t_d}} \quad \sum_{e=1}^{E} a_{ep} u_{ed} - \sum_{e=1}^{E} b_{ep} u_{ed} = -1, \tag{8}$$

$$\forall_{e=1,\dots,E} \quad \sum_{d=1}^{D} V_d u_{ed} \leqslant \sum_{k=1}^{K} M_{ek} y_{ek}. \tag{9}$$

In the aforementioned formulation, indices denote the following: $r = 1, \dots, R$ routers in a network, $c = 1, \dots, C$ cards in a network, $p = 1, \dots, P$ ports in a network, $e = 1, \dots, E$ links between two ports, $k = 1, \dots, K$ EAS of links, and $d = 1, \dots, D$ demands.

Constants denote the following:
$g_{rc} = 1$ if the card c belongs to the router r (0 otherwise), $l_{cp} = 1$ if the port p belongs to the card c (0 otherwise), $a_{ep} = 1$ if the link e is outgoing from the port p (0 otherwise), $b_{ep} = 1$ if the link e is incoming to the port p (0 otherwise), V_d volume of the demand d, s_d source edge node for the demand d, t_d destination edge node for the demand d, M_{ek} capacity of the link e in the state k, ξ_{ek} power consumed by the link e in the state k – the sum of power consumed by two ports connected by the link e), W_c fixed power consumed by the card c, and T_r fixed power consumed by of the router r.

Variables denote the following:
$z_r = 1$ if the router r is used for data transmission (0 otherwise), $x_c = 1$ if the card c is used for data transmission (0 otherwise), $u_{ed} = 1$ if the path d belongs to the link e (0 otherwise), and $y_{ek} = 1$ if the link e is in the state k (0 otherwise).

DOI: 10.1002/cpe

The constraints (2)–(4) determine the number of routers and cards that are used for data transmission. The conditions specified in Equation (5) assure that each link can be in one energy-aware state. The constrains defined in Equations (6)–(9) express the conservation of flow. The Equations (6)–(8) are formulated according to Kirchhoff's law applied for source, transit, and destination nodes. Finally, constraint (9) assures that the flow will not exceed the capacity of a given link.

5.2. Formulation with predefined paths: link-path problem (LPP)

The problems (1)–(9) are an NP-complete challenging optimization task and can appear to be far too difficult to be solved in a reasonable time due to numerous binary variables. It is necessary to reduce its complexity and dimension. This can be performed by introducing different modifications. The first one is to predefine possible sets of paths for all traffic relations. Hence, the *link-path* formulation in which the lists of candidate paths are predefined is proposed.

$$\text{minimize}_{x_c, y_{ek}, z_r, u_{hd}} \left\{ F_{LP} = \sum_{e=1}^{E} \sum_{k=1}^{K} \xi_{ek} y_{ek} + \sum_{c=1}^{C} W_c x_c + \sum_{r=1}^{R} T_r z_r \right\}, \tag{10}$$

subject to the following constraints:

$$\forall_{\substack{d=1,\dots,D,\\ c=1,\dots,C}} \quad \sum_{h \in P(d)} l_{ch} u_{hd} \leqslant x_c, \tag{11}$$

$$\forall_{\substack{r=1,\dots,R,\\ c=1,\dots,C}} \quad g_{rc} x_c \leqslant z_r, \tag{12}$$

$$\forall_{e=1,\dots,E} \quad \sum_{k=1}^{K} y_{ek} \leqslant 1, \tag{13}$$

$$\forall_{d=1,\dots,D} \quad \sum_{h \in P(d)} u_{hd} = 1, \tag{14}$$

$$\forall_{e=1,\dots,E} \quad \sum_{d=1}^{D} \sum_{h \in P(d)} \delta_{edh} V_d u_{hd} \leqslant \sum_{k=1}^{K} M_{ek} y_{ek}, \tag{15}$$

where the index $h \in P(d)$ denotes a path predefined for the demand d, $u_{hd} = 1$ if the demand d belongs to the path h, (0 otherwise), $l_{ch} = 1$ if the path h passes through the card c (0 otherwise), and $\delta_{edh} = 1$ if the link e belongs to the path h transmitting the demand d, (0 otherwise).

In this formulation, the constraints defined in Equations (11) and (12) determine the number of cards and routers that are used for data transmission, and the constraints (13) assure that each link can be in one energy-aware state. Finally, the conditions specified in Equation (14) assure that all requests for traffic will be enforced, and Equation (15) assures that the flow will not exceed the capacity of a given link. Concluding, as the solution of the aforementioned problems (1)–(9) and (10)–(15), we obtain the optimal network configuration minimizing the power consumption (i.e., set of active routers and cards and suggested states of all ports) and the routing table for the MPLS protocol that allows to switch off some network equipment and fulfills all constraints, mainly concerned with expected quality of service. The implementation of the described control strategy in the control framework presented in Figure 2 is depicted in Figure 4.

The complexity of LNP and LPP formulations may be compared by estimating the number of variables and the number of constraints for some example networks. To do this, precisely, it is necessary to have a complete description of some network, preferably a real one, together with the detailed demand scenario. However, the dimensionality of these tasks may be roughly estimated by simpler calculations for some generic structure of the network. To allow comparison, three relatively sparse networks composed of 20, 50, and 100 nodes connected, respectively, by 40, 100, and 200 links constructed of 12 fibers each were considered. It was assumed that fibers could be individually switched-off and on, each node was constructed of a single router with four cards and appropriate

DOI: 10.1002/cpe

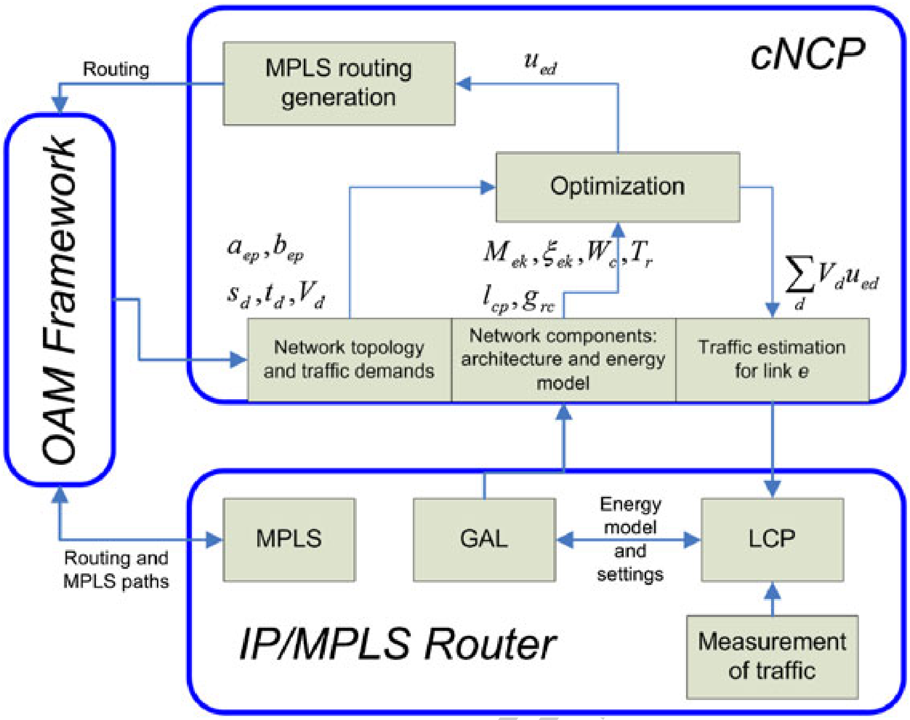


Figure 4. The control plane—architecture and data flow.

Color Online, B&W in Print

Table I. Estimation of LNP and LPP computational complexity for various sizes of the network.

Number of nodes	Number of	LPP	LNP
20	Variables	7300	515,300
	Constraints	67,680	146,912
50	Variables	18,250	3,208,250
	Constraints	415,200	913,232
100	Variables	36,500	12,816,500
	Constraints	1,650,400	3,646,432

LPP, link-path problem; LNP, link-node problem.

number of ports, all links had the same speed and could work in one of five EAS. The uniform distribution of sources and destinations among all nodes was assumed. Moreover, for the LPP, a set of five paths for each traffic relation was assumed. The results of all estimations are presented in Table I. T1

The results show the superiority of the formulation with predefined paths both in number of variables and number of constraints; however, the complexity grows with the number of nodes, making the solution difficult for larger networks. In particular, the number of constraints grows faster than the number of variables; despite more complex Kirchhoff's law, constraints (6), (7), and (8) are eliminated in the variant with predefined paths. It must be also noted that the example networks analyzed are sparse and their number of links grows linearly with the number of nodes, which is a relatively mild case when compared with real networks. On the other hand, many core networks, constructed of, for example, several intersecting rings exhibit similar sparsity. So, the presented estimation may be considered a lower bound for realistic applications.

Note that the formulations (10)–(15) are easier to solve because of a smaller number of constraints, but it is still *NP*-complete; therefore, one cannot expect to find time efficient algorithms for exactly solving both presented problems for medium-size and large-size networks. Techniques

DOI: 10.1002/cpe

based on the branch-and-bound approach are proposed for solving smaller size problems. Thus, for more realistic network sizes, the formulations (1)–(9) have to be relaxed and heuristics have to be employed to solve it.

5.3. Problem relaxation—heuristic approach, LNHP

A heuristic has been developed and applied to the complete link-node formulation (1)–(9). Similarly to the formulations LNP and LPP presented in the previous subsections, we assume that all pairs of ports from different routers and cards are linked by direct links labeled with $e = 1, 2, \ldots, E$, and each element can operate in various EAS labeled with $k = 1, 2, \ldots, K$ with power profile model presented in Figure 3. We assume that at a given time, two ports connected by the link e are in the same energy-aware state k. The only difference is the meaning of decision variables: z_r, x_c, u_{ed}, and y_{ek}. They are continuous and are defined as follows:

- $u_{ed} \in [0, 1]$ part of the demand d directed through the link e;
- $y_{ek} \in [0, 1]$ utilized throughput of the link e in the energy-aware state k;
- $x_c \in [0, 1]$ utilized resources of the card c in data transmission; and
- $z_r \in [0, 1]$ utilized resources of the router r in data transmission.

The power consumption and throughput utilization of the link e in the state k are described in the form of incremental model. The current values of ξ_{ek} and M_{ek} are calculated as follows:

- $\xi_{ek} = pow_e(k) - pow_e(k-1)$, where $pow_e(k)$ denotes the power used by the link e in the state k.
- $M_{ek} = load_e(k) - load_e(k-1)$, where $load_e(k)$ denotes the load of the link e in the state k.

Because of the presented relaxation, we can formulate a linear optimization problem that is stated in terms of continuous variables, with linear cost function to be minimized

$$\text{minimize}_{x_c, y_{ek}, z_r, u_{ed}} \left\{ F_{LNH} = \sum_{e=1}^{E} \sum_{k=1}^{K} \xi_{ek} y_{ek} + \sum_{c=1}^{C} W_c x_c + \sum_{r=1}^{R} T_r z_r \right\}, \tag{16}$$

subject to the following constraints:

$$\forall_{e=1,\ldots,E} \quad y_{e1} \geqslant y_{ei} \geqslant \ldots \geqslant y_{eK}, \tag{17}$$

$$\forall_{\substack{e=1,\ldots,E,\\ k=1,\ldots,K, c=1,\ldots,C}} \quad \sum_{p=1}^{P} l_{cp} a_{ep} y_{ek} \leqslant x_c, \tag{18}$$

$$\forall_{\substack{e=1,\ldots,E,\\ k=1,\ldots,K, c=1,\ldots,C}} \quad \sum_{p=1}^{P} l_{cp} b_{ep} y_{ek} \leqslant x_c, \tag{19}$$

$$\forall_{\substack{r=1,\ldots,R,\\ c=1,\ldots,C}} \quad g_{rc} x_c \leqslant z_r, \tag{20}$$

$$\forall_{\substack{d=1,\ldots,D,\\ p=s_d}} \quad \sum_{e=1}^{E} a_{ep} u_{ed} - \sum_{e=1}^{E} b_{ep} u_{ed} = 1, \tag{21}$$

$$\forall_{\substack{d=1,\ldots,D,\\ p=t_d, p\neq s_d}} \quad \sum_{e=1}^{E} a_{ep} u_{ed} - \sum_{e=1}^{E} b_{ep} u_{ed} = 0, \tag{22}$$

$$\forall_{\substack{d=1,\ldots,D,\\ p=t_d}} \quad \sum_{e=1}^{E} a_{ep} u_{ed} - \sum_{e=1}^{E} b_{ep} u_{ed} = -1, \tag{23}$$

$$\forall_{e=1,\ldots,E} \quad \sum_{d=1}^{D} V_d u_{ed} \leqslant \sum_{k=1}^{K} M_{ek} y_{ek}. \tag{24}$$

DOI: 10.1002/cpe

In this formulation, we allow the situation in which a given link is in more than one energy state but we add the constraint (17) for utilized throughput in various states; utilized throughput in different states are sorted. The constraints (18), (19), and (20) force binary values of variables z_r, x_c in case when y_{ek} takes a binary value. The meaning of the remaining constraints (21)–(24) is the same like in the LNP problem.

We have developed an efficient algorithm to solve the relaxed problem (16)–(24). The algorithm is composed of two phases. In the first phase, the preliminary solution is calculated by means of a commonly used linear solver. In the optimal solution, all decision variables should take binary values. We stop the algorithm when the calculated optimal decision variables take binary values (0 or 1), otherwise, we execute the second phase—we modify the original problem and repeat the calculations. In general, the algorithm operates as follows.

Algorithm LNH_solve.

step 0 : Start. Let OP denote the formulation of the optimization problem (16)–(24).

step 1 : Solve the optimization problem OP. Calculate the optimal values of the decision variables: $\hat{z}_r$, $\hat{x}_c$, $\hat{u}_{ed}$ and $\hat{y}_{ek}$.

step 2 : Check the results of the optimization process calculated in step 1. If all calculated variables $\hat{z}_r$, $\hat{x}_c$, $\hat{u}_{ed}$, and $\hat{y}_{ek}$ take binary values—STOP. Otherwise,
- if the links for which $\hat{y}_{ek} \in (0,1)$ have been detected, then create a subset $S_E^* \subset S_E$ consisting all these links and execute step 3. Otherwise,
- if the links for which $\hat{u}_{ed} \in (0,1)$ have been detected, then create a subset $S_E^{**} \subset S_E$ consisting all these links and execute step 4.

step 3 : Create a subset $S_{E_{min}} \subset S_E^*$ consisting of links that operate in the lowest energy-aware state k^*. Select from the set $S_{E_{min}}$ a link e^* for which $\hat{y}_{ek}$ takes a minimal value. Remove $\hat{y}_{e^*k^*}$ from the set S_E^*. Formulate the extended optimization problem OPe—add to the optimization problem OP a new constraint $y_{e^*k^*} = 1$. Assign OP = OPe and back to step 1.

step 4 : Select from the set S_E^{**} a link e^{**} for which $\hat{u}_{ed}$ takes a maximal value. Remove $\hat{u}_{e^{**}d}$ from the set S_E^{**}. Formulate the extended optimization problem OPe—add to the optimization problem OP a new constraint $u_{e^{**}d} = 1$. Assign OP = OPe and back to step 1.

Summarizing the aforementioned algorithm, the optimization problem (16)–(24) is repetitively modified and solved until all decision variables take binary values (0 or 1).

6. PERFORMANCE EVALUATION

We validated our strategies and algorithms for energy saving in a backbone computer network through numerical evaluation. Two formulations of a network energy saving optimization problem described in Section 5 were applied to dynamic control of five synthetic network topologies composed by routers and SDH links and inspired by the real network structures. We compared the performance and efficiency of the following approaches to centralized power management in computer networks. Q4

- Complete LNP (1)–(9) and branch-and-bound solver.
- Relaxed problem LNHP (16)–(24) and heuristic algorithm LNH_solve combined with linear solver.

To model the power consumption of routers and links, we considered power requirements of network devices provided in [37]. The backbone routers in our tests were equipped with one or more line cards with three or four ports, whereas the edge routers were equipped with only one card. We assumed that each router and card could operate in two modes—active or sleeping. Each port, and at the same time link, could operate in five EAS (from the active to the sleeping one), which differed in used power and corresponding throughput. The network equipment enabling energy

Table II. Power consumption in different states—routers and cards.

Device	Active state	Sleep state
Router	2000 W	100 W
Line card	100 W	10 W

Table III. Power consumption and corresponding throughput—links.

	State 1	State 2	State 3	State 4	State 5
Power consumption	16 W	32 W	48 W	64 W	80 W
Throughput	200 Mb/s	400 Mb/s	600 Mb/s	800 Mb/s	1 Gb/s

saving is under development now, hence, we had to assume amounts of power used in EAS and corresponding throughput. Tables II and III report the power used by network devices operating in different states. T2 T3

Two series of numerical evaluations were performed. The goal of all of them was to check and compare the performance of LNP and LNHP approaches for different networks' dimensions. The branch-and-bound implementation was incorporated from the open source solver Lp_solve (http://lpsolve.sourceforge.net/5.5/). All tests were performed on Intel Core2 Duo CPU, 2.2 GHz, 2 GB RAM. Q5

6.1. Performance comparison of link-node problem and LNHP—small-size networks

The objective of the experiments was to compare the performance and efficiency of LNP and LNHP approaches to energy saving in a small-size network. The network composed of six routers, eight line cards, and 10 unidirectional links was considered (see Figure 5a). We performed five experiments for various numbers of demands E1: $D=3$, E2: $D=5$, E3: $D=7$, E4: $D=10$, E5: $D=13$. The source and destination edge nodes for demands taken into account in all experiments (E1–E5) are specified in Table IV. The dimensions and complexity of each considered optimization problem and the results of calculations, that is, values of performance functions describing power consumption for all tests are given in Table V. Moreover, the table presents the execution times for all tests. The values collected in the adequate columns denote *Var.*— the number of decision variables, *Constr.*—the number of constraints, *Subtasks*—the number of calculated subtasks generated by branch-and-bound (B&B), and LNH_solve solvers, *Solution*—the value of the performance measure for the optimal vector of decision variables (total power in Watt), and *Time*—time of calculation in seconds. T4 T5

Table IV. Source and destination edge nodes for demands in experiments E1–E5.

Demand	Experiments	Source node (s_d)	Destination node (t_d)
$d=1$	E1, E2, E3, E4, E5	A1	A6
$d=2$	E1, E2, E3, E4, E5	A1	A2
$d=3$	E1, E2, E3, E4, E5	A1	A6
$d=4$	E2, E3, E4, E5	A1	A5
$d=5$	E2, E3, E4, E5	A2	A6
$d=6$	E3, E4, E5	A2	A6
$d=7$	E3, E4, E5	A2	A5
$d=8$	E4, E5	A6	A2
$d=9$	E4, E5	A3	A6
$d=10$	E4, E5	A2	A4
$d=11$	E5	A1	A6
$d=12$	E5	A1	A2
$d=13$	E5	A1	A6

DOI: 10.1002/cpe

Table V. Problem complexity and results of calculations (small-size network).

Experiment	Method	*Var.*	*Constr.*	*Subtasks*	*Solution* [W]	*Time* [s]
E1	B&B	176	174	402	8146	0.078
$D=3$	LNH_solve	176	189	6	8146	0.094
E2	B&B	216	224	1298	10,232	0.281
$D=5$	LNH_solve	216	222	9	10,232	0.187
E3	B&B	256	276	3152	10,264	0.826
$D=7$	LNH_solve	256	255	10	10,264	0.140
E4	B&B	316	350	74,104	12,286	7.285
$D=10$	LNH_solve	316	302	13	12,318	0.218
E5	B&B	376	428	210,610	12,350	103.303
$D=13$	LNH_solve	376	354	17	12,382	0.405

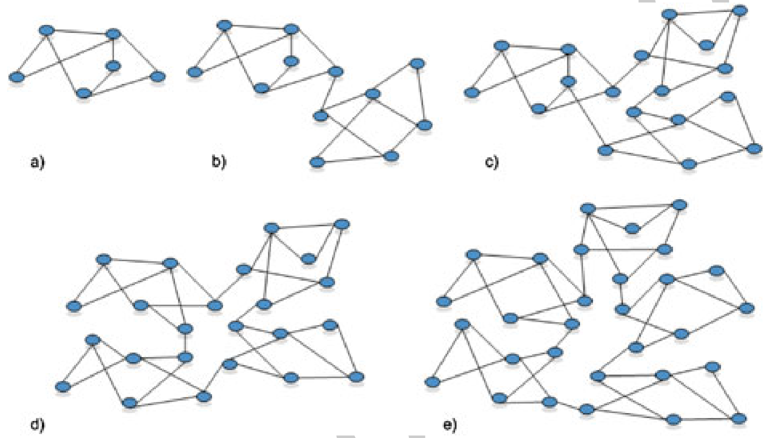


Figure 5. Synthetic networks, (a) *Network 1*, (b) *Network 2*, (c) *Network 3*, (d) *Network 4*, (e) *Network 5*.

The optimal values of decision variables (states of routers, cards, and ports) calculated, for example, E1 are presented in Figure 6. We can observe that because of the results of optimization, one router and two cards belonging to two other routers were put to a sleep mode and all active ports were in different EAS. The results indicate that the calculation time strongly depends on the complexity of the optimization problem. It is obvious that the complexity of the problem grows with the size of the network and the number of demands to be considered. It was observed that D (number of demands) is a key parameter that seriously increases the calculation time. From the depicted results, we can see that both LNP and LNHP give similar results (similar cost function), but the time of calculations is dramatically reduced after using the heuristic method.

6.2. Performance evaluation of LNHP—medium-size networks

The objective of the second series of tests was to check the performance of our heuristic algorithm when applied to power management in medium-size networks. Five network configurations were considered, that is,

- *Network 1* : $R = 6$, $C = 8$, $E = 10$, $D = 7$, $K = 5$,
- *Network 2* : $R = 12$, $C = 16$, $E = 21$, $D = 21$, $K = 5$,
- *Network 3* : $R = 18$, $C = 24$, $E = 33$, $D = 35$, $K = 5$,
- *Network 4* : $R = 24$, $C = 30$, $E = 44$, $D = 42$, $K = 5$,
- *Network 5* : $R = 30$, $C = 50$, $E = 55$, $D = 49$, $K = 5$,

where R, C, E, D, and K denote, respectively, numbers of routers, cards, links, demands, and number of EAS for all links. We assumed that each backbone router was equipped with one or two line cards, whereas the edge routers were equipped with only one card. The detailed specification

DOI: 10.1002/cpe

Color Online, B&W in Print

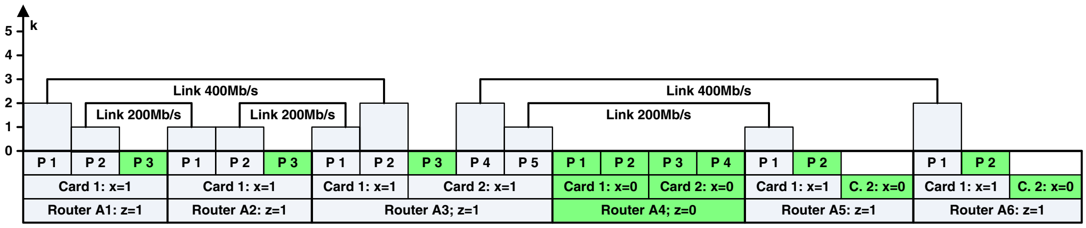


Figure 6. Calculated states of routers, cards, and ports (example E1); k denotes a given energy-aware state.

for *Newtork 1* is presented in Figure 6. The topologies of *Network 2–Network 5* were formed by the combination of 2, 3, 4, and 5 *Newtork 1* topologies, respectively. Contrarily to the tests reported in Table V, we considered bidirectional links, which resulted in increasing complexity of the problems. The topologies of the listed networks are presented in Figure 5. Experiments confirmed that defining the complete network problem (LNP) and using the B&B technique to solve a medium-size network problem needs very high computation overhead and is impractical for medium and large scale networks. Therefore, we formulated the relaxed optimization problems (16)–(24) for all listed networks and applied our heuristic algorithm LNH_solve to solve them. The dimensions and complexity of each considered optimization problem and the results of calculations, that is, values of performance functions describing power consumption for all tests and calculation times are given in Table VI.

F6 T6

Values placed in the columns correspond to the variables from Table VI. *Full power* denotes total power when the simple shortest path routing is calculated without any energy saving mechanisms (all network equipment operate in the active state with the typical power usage). The results confirm that the relaxation of the problem formulation and branch-and-bound implementation supported by efficient heuristics give good results in reasonable time.

It is obvious that the calculation time nonlinearly increases with the complexity of the problem. Parallel calculation is a common approach to decrease the computation time. The only problem is network decomposition. It is impractical to apply parallel computation to strongly connected networks (the decomposition can be ineffective). Fortunately, in case of many practical applications,

Table VI. Problem complexity and results of calculations (medium-size networks).

Example	*Var.*	*Constr.*	*Subtasks*	*Solution* [W]	*Full power* [W]	*Time* [s]
Network 1	256	246	10	10,264	13,800	0.161
Network 2	1124	953	22	20,772	27,680	0.755
Network 3	2688	2121	27	31,126	41,640	4.245
Network 4	4200	3276	38	41,634	55,520	13.770
Network 5	6020	4655	57	51,840	69,400	29.499

Color Online, B&W in Print

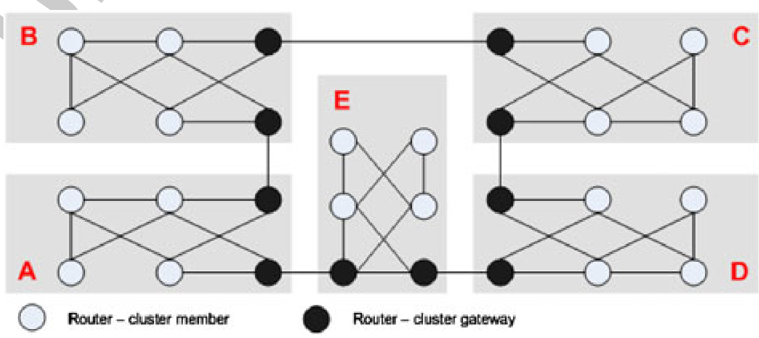


Figure 7. Decomposition of *Network 5* into five subnetworks.

DOI: 10.1002/cpe

Table VII. Subproblems complexity and results of calculations.

Subnetwork	*Var.*	*Const.*	*Subtasks*	*Solution* [W]	*Time* [s]
A: $R=6$, $E=10$, $D=19$	496	438	11	10,264	0.164
A: $R=6$, $E=10$, $D=21$	536	470	13	10,360	0.201
C: $R=6$, $E=10$, $D=19$	496	438	15	10,328	0.266
D: $R=6$, $E=10$, $D=23$	576	502	27	10,392	0.433
E: $R=6$, $E=10$, $D=23$	576	502	13	10,360	0.222

especially when considering core and backbone networks, it is natural to divide the whole network into subnetworks, formulate several parallel optimization problems, and solve them in a multiprocessor machine or in a cluster of workstations. Figure 7 shows the decomposition of *Network 5* into five similar size subnetworks as follows: A, B, C, D, and E. The subtasks complexity and results of calculations are collected in Table VII. It can be observed that the total computation time for the whole network dramatically decreases after problem decomposition, and the solution (value of the cost function) is similar.

F7

T7

7. SUMMARY AND CONCLUSIONS

This paper presents the control framework for energy-aware networks composed of two decision layers, namely, central and local control layers. The central decision unit is responsible for the power control in the whole network, and the local decision unit is responsible for the control of the configuration of components of individual network devices. We formulated an energy saving optimization problem for IP/MPLS networks (the complete link-node formulation) under the assumption of applying power scaling and standby techniques in order to reduce the power consumption in a given network. This problem is indicated as an *NP*-complete task and is too difficult for many conventional optimization methods, such as branch-and-bound algorithms. To reduce the problem complexity, we provided the modified version of the problem with predefined paths (the link-path formulation). However, such formulation is still *NP*-complete. Finally, we developed a heuristic and employed it to the complete link-node formulation. The idea was based on transforming an integer programming problem into a linear continuous one. Multiple numerical experiments were performed to test the efficiency of the proposed solutions. The results of tests show that application of power saving strategies allows to achieve the desired trade-off between power consumption and network performance according to the traffic load. As a final observation, we can say that the simplified formulation of the network-wide optimization problem and more efficient branch-and-bound implementation supported by heuristics must be considered, especially for medium and large networks. The second approach to speed up calculations is a parallel implementation of the optimization task. In the nearest future, we plan to provide a comprehensive empirical evaluation of the presented control framework for energy-aware networks in the OmNetC++ simulator (http://www.omnetpp.org/) and on a testbed network.

ACKNOWLEDGEMENTS

This research was supported by ECONET 7/ Framework Program UE grant (258454).

REFERENCES

1. ECONET. project homepage. (Available from: http://www.econet-project.eu). Q8
2. Wang L, Khan S. Review of performance metrics for green data centers: a taxonomy study. *The Journal of Supercomputing* 2011;1–18. DOI: 10.1007/s11227-011-0704-3. Q9
3. Kołodziej J, Khan S, Wang L, Zomaya A. Energy efficient genetic-based schedulers in computational grids. *Concurrency and Computation: Practice and Experience* 2012; DOI: 10.1002/cpe.2839. Q10
4. Lorch J, Smith A. Improving dynamic voltage scaling algorithms with pace. *Proceedings of the ACM SIGMETRICS 2001 International Conference on Measurement and Modeling of Computer Systems*, 2001; 50–61. Q11

5. Min R, Furrer T, Chandrakasan A. Dynamic voltage scaling techniques for distributed microsensor networks. *Proceedings of IEEE Workshop on VLSI*, 2000; 43–46.
6. Bianzino AP, Chaudet C, Rossi D, Rougier JL. A survey of green networking research. *to appear in IEEE Communication Surveys and Tutorials* 2012. http://arxiv.org/pdf/1010.3880.pdf.
7. Q12 Bolla R, Bruschi R. Energy-aware load balancing for parallel packet processing engines. In *Online Conference on Green Communications (GreenCom)*. IEEE, 2011; 105–112.
8. Bolla R, Bruschi R, Davoli F, Cucchietti F. Energy efficiency in the future internet: a survey of existing approaches and trends in energy-aware fixed network infrastructures. *IEEE Communications Surveys & Tutorials* 2011; **13**:223–244.
9. Q13 Kołodziej J, Khan S, Zomaya A. A taxonomy of evolutionary-inspired solutions for energy optimization: problems and intelligent resolution techniques. In *Advances in Intelligent Modelling and Simulation: Artificial Intelligence-based Models and Techniques in Scalable Computing, Chapter 10, Studies in Computational Intelligence*, Vol. 442. Springer-Verlag: Berlin-Heidelberg, 2012.
10. Q14 Valentini GL, Lassonde W, Khan SU, Min-Allah N, Madani SA, Li J, Zhang L, Wang L, Ghani N, Kołodziej J, *et al.*. An overview of energy efficiency techniques in cluster computing systems. *Cluster Computing* 2011; DOI: 10.1007/s10586-011-0171-x.
11. Q15 Hewlett-Packard Corp., Intel Corp., Microsoft Corp., Phoenix Technologies Ltd., Toshiba Corp. *Advanced Configuration and Power Interface Specification, Revision 5.0*, 2011.
12. IEEE. Institute of Electrical and Electronics Engineers, IEEE 802.3az Energy Efficient Ethernet Task Force, 2012. (Available from: http://grouper.ieee.org/groups/802/3/az/public/index.html).
13. Bolla R, Bruschi R, Ranieri A. Green support for pc-based software router: performance evaluation and modeling. *Proceedings of IEEE International Conference on Communications (ICC'09)*, 2009; 1–6.
14. Sivaraman V, Vishwanath A, Zhao Z, Russell C. Profiling per-packet and per byte energy consumption in the NetFPGA gigabit router. In *Proceedings of IEEE Workshop on Green Communications and Networking (INFOCOM 2011)*. IEEE, 2011; 331–336.
15. Bolla R, Bruschi R, Carrega A, Davoli F, Suino D, Vassilakis C, Zafeiropoulos A. Cutting the energy bills of Internet Service Providers and telecoms through power management: an impact analysis. *Computer Networks* 2012; **56**:2320–2342.
16. Qureshi A, Weber R, Balakrishnan H. Cutting the electric bill for internet-scale systems. In *Proceedings of SIGCOMM'09*. ACM, 2009; 123–134.
17. Vasić N, Kostić D. Energy-aware traffic engineering. *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking (E-ENERGY 2010)*, 2010.
18. Chiaraviglio L, Mellia M, Neri F. Minimizing ISP network energy cost: formulation and solutions. *IEEE/ACM Transactions on Networking* 2011; **20**:463–476.
19. Reforgiato D, Lombardo A, Davoli F, Fialho L, Collier M, Donadio P, Bolla R, Bruschi R. Exporting data-plane energy-aware capabilities from network devices toward the control plane:. *Proceedings of the 17th European Conference on Network and Optical Communication (NOC 2012)*, 2012; 1–5.
20. Bianco F, Cucchietti G, Griffa G. Energy consumption trends in the next generation access network – a telco perspective. *Proceedings of the 29th International Telecommunication Energy Conference (INTELEC 2007)*, 2007; 737–742.
21. Roy SN. Energy logic: a road map to reducing energy consumption in telecommunications networks. *Proceedings of the 30th International Telecommunication Energy Conference (INTELEC 2008)*, 2008.
22. Pióro M, Mysłek M, Juttner A, Harmatos J, Szentesi A. Topological design of MPLS networks. *Proceedings of GLOBECOM'2001*, 2001.
23. Jaskóła P, Malinowski K. Two methods of optimal bandwidth allocation in TCP/IP networks with QoS differentiation. *Proceedings of Summer Simulation Multiconference (SPECTS'04)*, 2004; 373–378.
24. Malinowski K, Niewiadomska-Szynkiewicz E, Jaskóła P. Price method and network congestion control. *Journal of Telecommunications and Information Technology* 2010; **2**:73–77.
25. Chabarek J, Sommers J, Barford P, Estan C, Tsiang D, Wright S. Power awerness in network design and routing. *Proceedings of the 27th Conference on Computer Communications (INFOCOM 2008)*, 2008; 457–465.
26. Niewiadomska-Szynkiewicz E, Sikora A, Arabas P, Kołodziej J. Control framework for high performance energy aware backbone network. *Proceedings of the European Conference on Modelling and Simulation ECMS'2012*, 2012; 490–496.
27. Restrepo J, Gruber C, Machuca C. Energy profile aware routing. *Proceedings of the 1st International Workshop on Green Communications, IEEE International Conference on Communications (ICC'09)*, 2009; 1–5.
28. Fisher W, Suchara M, Rexford J. Greening backbone networks: reducing energy consumption by shutting off cables in bundled links. In *Proceedings of the 1st ACM SIGCOMM Workshop on Green Networking (Green Networking'10)*. ACM, 2010; 29–34.
29. Idzikowski F, Orlowski S, Raack C, Rasner H, Wolisz A. Saving energy in IP-over-WDM networks by switching off line cards in low-demand scenarios. In *Proceedings of the 14th Conference on Optical Network Design and Modeling (ONDM'10)*. IEEE, 2010.
30. Kołodziej J, Khan S, Xhafa F. Genetic algorithms for energy-aware scheduling in computational grids. *Proceedings of the 6th IEEE International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC2011), 26–28.10.2011*, Barcelona, Spain, 2011; 17–24.

31. Chiaraviglio L, Mellia M, Neri F. Energy-aware backbone networks: a case study. In *Proceedings of the 1st International Workshop on Green Communications, IEEE International Conference on Communications (ICC'09)*. IEEE, 2009; 1–5.
32. Zhang M, Yi C, Liu B, Zhang B. GreenTE: power-aware traffic engineering. In *Proceedings of IEEE International Conference on Network Protocols (ICNP'2010)*. IEEE, 2010.
33. Shen G, Tucker RS. Energy-minimized design for IP over WDM networks. *Journal of Optical Communications and Networking* 2009; **1**:176–186.
34. Cianfrani A, Eramo V, Listani M, Marazza M, Vittorini E. An energy saving routing algorithm for a green OSPF protocol. In *Proceedings of IEEE INFOCOM Conference on Computer Communications*. IEEE, 2010; 1–5.
35. Bianzino AP, Chiaraviglio L, Mellia M. GRiDA: a green distributed algorithm for backbone networks. In *Online Conference on Green Communications (GreenCom 2011)*. IEEE, 2011; 113–119.
36. Cuomo F, Abbagnale A, Cianfrani A, Polverini M. Keeping the connectivity and saving the energy in the internet. In *Proceedings of IEEE INFOCOM 2011 Workshop on Green Communications and Networking*. IEEE, 2011; 319–324.
37. Bolla R *et al.* Econet deliverable d2.1 end-user requirements, technology, specifications and benchmarking methodology, 2011. (Available from: https://www.econet-project.eu/Repository/DownloadFile/291).

DOI: 10.1002/cpe

Author Query Form

Journal: Concurrency and Computation: Practice and Experience

Article: cpe_2964

Dear Author,

During the copyediting of your paper, the following queries arose. Please respond to these by annotating your proofs with the necessary changes/additions.

- If you intend to annotate your proof electronically, please refer to the E-annotation guidelines.
- If you intend to annotate your proof by means of hard-copy mark-up, please refer to the proof mark-up symbols guidelines. If manually writing corrections on your proof and returning it by fax, do not write too close to the edge of the paper. Please remember that illegible mark-ups may delay publication.

Whether you opt for hard-copy or electronic annotation of your proofs, we recommend that you provide additional clarification of answers to queries by entering your answers on the query sheet, in addition to the text mark-up.

Query No.	Query	Remark
Q1	AUTHOR: Changes have been made in the affiliations. Please check if appropriate.	
Q2	AUTHOR: Please define BGP.	
Q3	AUTHOR: Please define LNHP.	
Q4	AUTHOR: Please define SDH.	
Q5	AUTHOR: Please provide manufacturer information for Intel Core2 Duo CPU.	
Q6	AUTHOR: "(c) *Network 4*, c) *Network 5*" has been changed to "(d) *Network 4*, (e) *Network 5*", respectively. Please check if appropriate.	
Q7	AUTHOR: Figures have been renumbered according to citation order. Please check.	
Q8	AUTHOR: Please provide accessed date for References 1, 12 and 37.	
Q9	AUTHOR: Please provide volume for Reference 2.	
Q10	AUTHOR: If this reference has now been published online, please add relevant volume/issue/page range information for References 3, 6 and 10.	
Q11	AUTHOR: Please provide the city location where the proceedings or conference was held for References 4, 5, 13, 17, 19, 20, 21–23 and 25–27.	
Q12	AUTHOR: Please provide the city location of publisher for References 7, 14, 16, 28, 29, 31, 32 and 34–36.	

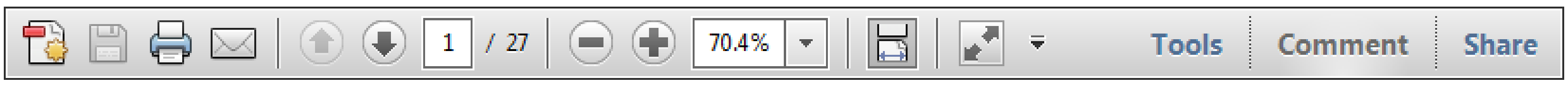
Q13	AUTHOR: Please provide the page range for References 9, 17, 21, 22, 29 and 32.	
Q14	AUTHOR: Please provide list of author's name for References 10 and 37.	
Q15	AUTHOR: Please provide the name of the publisher and city lcaotion for Reference 11.	

WILEY-BLACKWELL

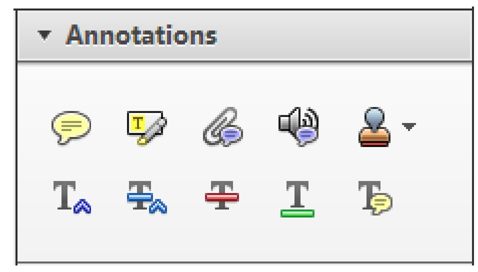
USING e-ANNOTATION TOOLS FOR ELECTRONIC PROOF CORRECTION

Required software to e-Annotate PDFs: Adobe Acrobat Professional or Adobe Reader (version 7.0 or above). (Note that this document uses screenshots from Adobe Reader X)
The latest version of Acrobat Reader can be downloaded for free at: http://get.adobe.com/uk/reader/

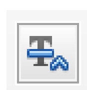
Once you have Acrobat Reader open on your computer, click on the Comment tab at the right of the toolbar:



This will open up a panel down the right side of the document. The majority of tools you will use for annotating your proof will be in the Annotations section, pictured opposite. We've picked out some of these tools below:



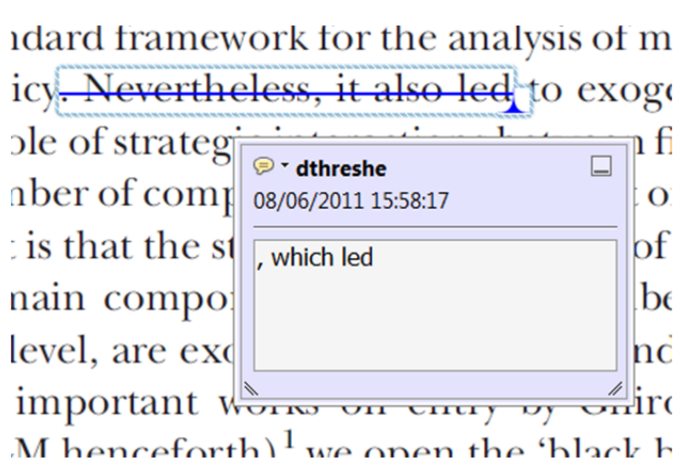
1. Replace (Ins) Tool – for replacing text.



Strikes a line through text and opens up a text box where replacement text can be entered.

How to use it

- Highlight a word or sentence.
- Click on the Replace (Ins) icon in the Annotations section.
- Type the replacement text into the blue box that appears.

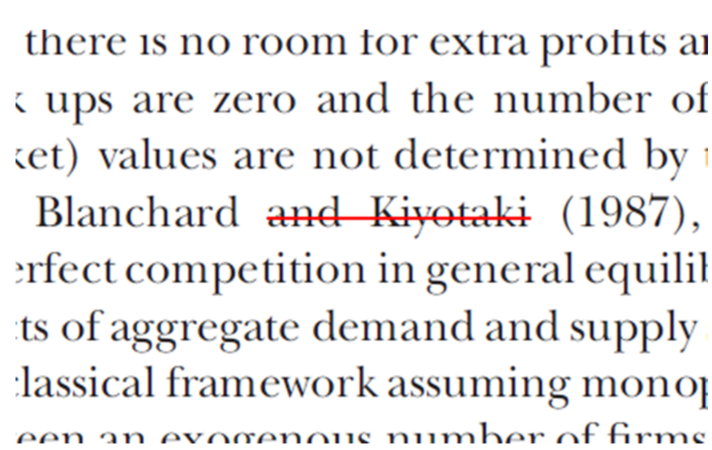


2. Strikethrough (Del) Tool – for deleting text.

Strikes a red line through text that is to be deleted.

How to use it

- Highlight a word or sentence.
- Click on the Strikethrough (Del) icon in the Annotations section.

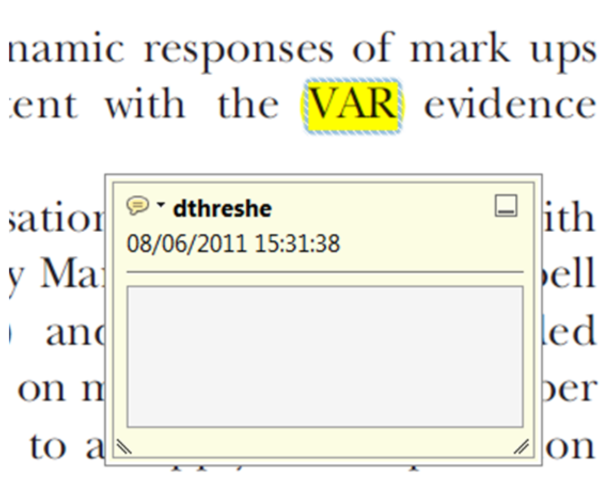


3. Add note to text Tool – for highlighting a section to be changed to bold or italic.

Highlights text in yellow and opens up a text box where comments can be entered.

How to use it

- Highlight the relevant section of text.
- Click on the Add note to text icon in the Annotations section.
- Type instruction on what should be changed regarding the text into the yellow box that appears.

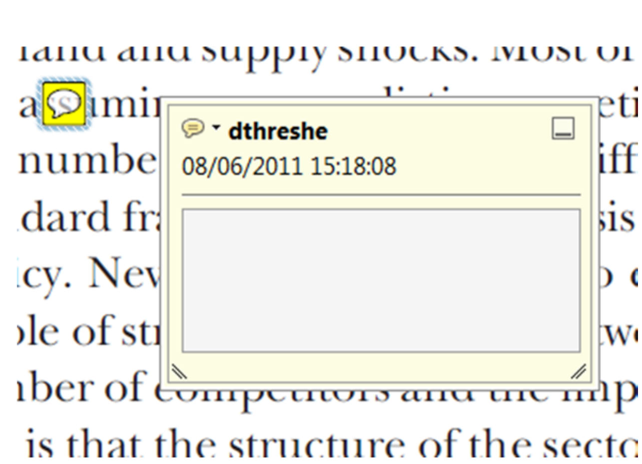


4. Add sticky note Tool – for making notes at specific points in the text.

Marks a point in the proof where a comment needs to be highlighted.

How to use it

- Click on the Add sticky note icon in the Annotations section.
- Click at the point in the proof where the comment should be inserted.
- Type the comment into the yellow box that appears.



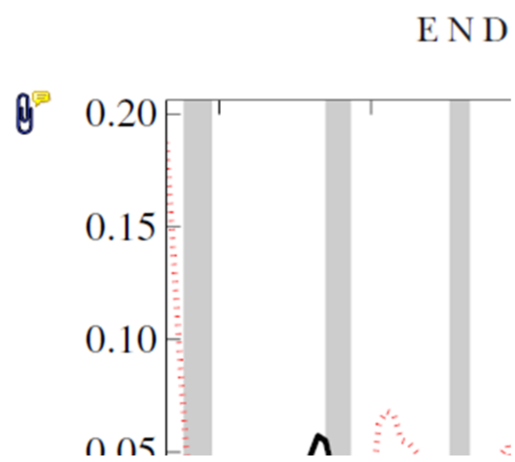
USING e-ANNOTATION TOOLS FOR ELECTRONIC PROOF CORRECTION

5. Attach File Tool – for inserting large amounts of text or replacement figures.

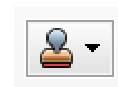
Inserts an icon linking to the attached file in the appropriate pace in the text.

How to use it

- Click on the Attach File icon in the Annotations section.
- Click on the proof to where you'd like the attached file to be linked.
- Select the file to be attached from your computer or network.
- Select the colour and type of icon that will appear in the proof. Click OK.



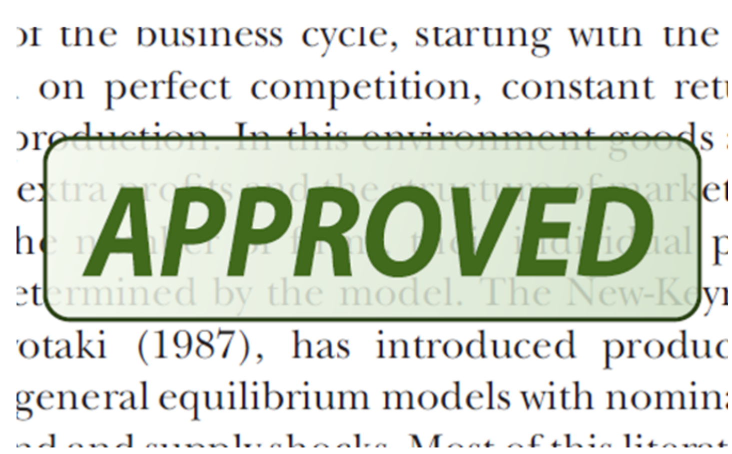
6. Add stamp Tool – for approving a proof if no corrections are required.

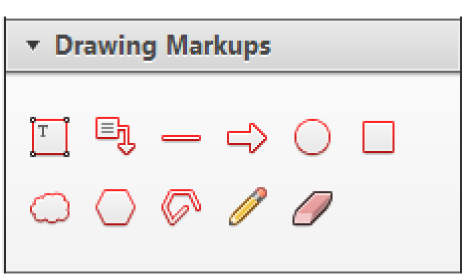


Inserts a selected stamp onto an appropriate place in the proof.

How to use it

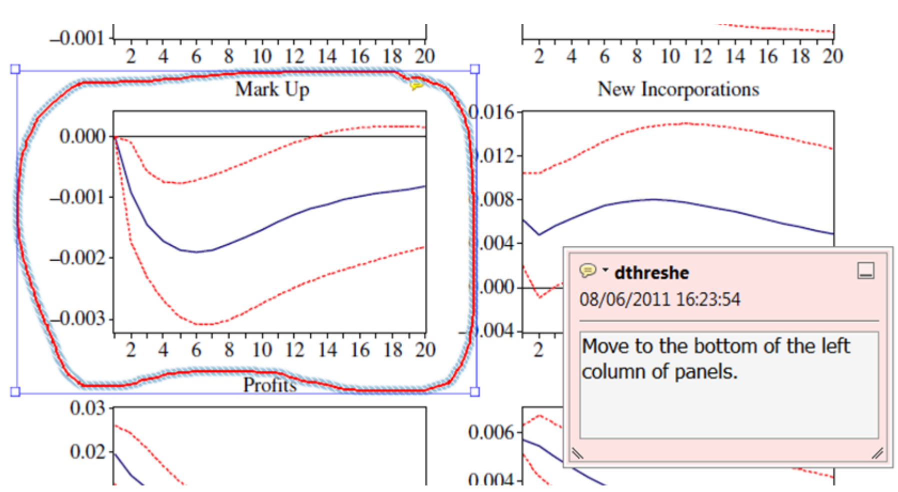
- Click on the Add stamp icon in the Annotations section.
- Select the stamp you want to use. (The Approved stamp is usually available directly in the menu that appears).
- Click on the proof where you'd like the stamp to appear. (Where a proof is to be approved as it is, this would normally be on the first page).





7. Drawing Markups Tools – for drawing shapes, lines and freeform annotations on proofs and commenting on these marks.

Allows shapes, lines and freeform annotations to be drawn on proofs and for comment to be made on these marks..



How to use it

- Click on one of the shapes in the Drawing Markups section.
- Click on the proof at the relevant point and draw the selected shape with the cursor.
- To add a comment to the drawn shape, move the cursor over the shape until an arrowhead appears.
- Double click on the shape and type any text in the red box that appears.

For further information on how to annotate proofs, click on the Help menu to reveal a list of further options: