Energy saving heuristics in Backbone Networks

Rosario G. Garroppo, Stefano Giordano, Gianfranco Nencioni, Michele Pagano
Dip. di Ingegneria dell'Informazione
Università di Pisa
firstname.lastname@iet.unipi.it

Maria Grazia Scutellà
Dip. di Informatica
Università di Pisa
scut@di.unipi.it

***Abstract*—The paper presents an approach aimed to reduce the overall power consumption of a backbone network by exploiting the power behavior of green network devices. This approach is based on the solution of an optimization problem that has a Mixed Integer NonLinear Programming (MINLP) formulation. Given that the problem is NP-hard, exact methods for finding optimal solutions can be used only for scenarios of limited size. To cope with the case of complex networks, the paper proposes two variations of the Fast Greedy Heuristic (FGH), denoted as Time Limited PAR Heuristic (TLPH) and PAR Meta Heuristic (PMH). The simulation study highlights the capability of the proposed heuristics to obtain solutions near the optimum and to outperform the other approaches in terms of power savings and CPU times needed to find a solution in complex network scenarios.**

I. Introduction

The Internet is rapidly becoming a major consumer of power, with significant economic and environmental impacts. Furthermore, energy consumption is one of the key issues for the future life. Considering these aspects, it is not surprising that communication operators and equipment vendors are trying to reduce the power consumption of networks, and a part of the research community is focused on the design of Green Networks.

Recent works on Green Networks have proposed the energy aware problems and (in some cases) solutions on two relevant aspects: the *Network Device Design* and the *Network Power Management*. Several approaches for these two issues can be jointly used.

The Network Device Design consists in energy efficient mechanisms implemented in network equipments. A plethora of mechanisms exists to reduce the energy consumption of electrical circuits, such as circuit simplifications or the use of clock gating approaches, i.e. the addition of circuits that disable portions of the circuitry when these are not used [1]. Other approaches refer to the utilization of Dynamic Voltage or Dynamic Frequency Scaling mechanisms (DVS, DFS) that are widely known from modern computer processors. The authors of [2] show that when DVS and DFS are jointly applied, the power consumption decreases cubically with the workload. Other methods for improving the power efficiency of Network Devices can be applied at system level. As an example, the current components can be exchanged by more energy efficient ones, such as the replacement of electrical components with their optical counterpart.

The Network Power Management refers to strategies of network design and planning and aims to achieve further energy savings by means of appropriate strategies that exploit the power behavior of green network devices [3] [4]. During network planning, a drastic decrease in energy consumption can already be achieved by preferring manufacturers with energy-efficient components or by minimizing energy-hungry components (e.g. large IP routers) and transporting traffic at the lowest (more energy efficient) layer if possible. An example of this approach is shown by the authors of [5], which illustrated the impact of choosing different equipment (chassis/line card configurations) on the overall network energy consumption. Recently, one of the most common practices for a power aware network dimensioning consists in resource consolidation. This solution is based on the observation that the traffic level in a given network approximately follows a well known daily and weekly behavior. By exploiting the knowledge of this traffic pattern, the operator has the chance to aggregate traffic flows over a subset of the network devices and links, allowing other devices to be temporarily powered off. Obviously, this strategy aims to use an amount of resources that is dimensioned over the actual traffic demand, rather than for the peak demand, while the connectivity and the required Quality of Service (QoS) are guaranteed.

The paper focuses on Network Power Management by describing a general approach for reducing the power consumption of current and future backbone networks. This approach leads to a Mixed Integer NonLinear Programming (MINLP) formulation, which is NP-hard in the general case. Hence, exact solution methods may be suitable only for small networks. However, the development of simple heuristics could lead to find suboptimal solutions in an acceptable computational time also for medium-large networks. In this framework, the paper firstly presents a suitable formulation for the described problem, then it proposes two heuristic approaches. The simulation analysis is aimed at comparing the power saving obtainable with the proposed approaches and at evaluating this ability to provide a solution near to the optimum.

The paper is structured as follows. Section II presents

978-3-901882-46-3

the background. Section III and Section IV introduce the problem statement and the related works, respectively. Then, Section V depicts the proposed heuristics. Section VI reports the network scenarios considered in the simulations, whereas Section VII presents the simulation settings and discusses the obtained results. The conclusions are drawn in Section VIII.

II. Background

One of the most relevant components for designing effective Network Power Management strategies is the energy characterization of the network devices. In the paper, we have taken into account a general power consumption model of a router, which consists of three main components [6]:

- chassis;
- Physical Interface Cards (PICs);
- route processor.

In general, the chassis can be powered off (i.e. it works at a low power mode); the corresponding power consumption can be assumed constant if the chassis is powered on, and zero otherwise. However, we assumed that in the *backbone network* the chassis can not be powered off since it can not quickly be activated in case of failures.

The energy consumed for transferring a bit from a node u to a node v is due to diverse components: the power consumed by the PIC in node u to transmit the bit and that used by the PIC of node v to receive it. In the rest of the paper, when considering traffic sent from a certain node u to a certain node v, we shall associate the corresponding power consumption and the related capacity value with the PIC at node u, without distinguishing the power contributions given by the transmitter and the receiver PICs. Obviously, in the reverse direction, we shall associate them with the PIC at node v. Furthermore, since in most actual scenarios the network operators try to use similar devices in their core network, we shall assume that the power consumption and the capacity of the PIC used to transmit along (u,v) are equal to those used for transmitting along (v,u). In general, however, in the case of a link connecting two nodes with diverse hardware features, the power consumption and the capacity of the PICs used to transfer the traffic in the two directions could be different. Each PIC can be powered off. In particular, there is a constant, non zero, power consumption when the PIC is powered on, and a zero power consumption when the PIC is powered off.

As far as the power consumption of the route processor is concerned, it generally depends on the traffic load of the router in a nonlinear way. In [3] the authors present several possible behaviors, such as linear, on-off, and cubic.

Finally, in modern core networks pairs of routers are typically connected, for each traffic direction, by multiple PICs that form one logical bundled link [7]; this technique is called link aggregation and is standardized by IEEE 802.1AX [8]. Link aggregation technique is widely diffused because it allows one to easily upgrade the link capacity by adding new PICs and so reaching link capacities bigger than that available by using the current fastest technology. For example, a 40 Gb/s bundled link may comprise four OC-192 PICs with capacity $10 Gb/s$ each. By taking into account this scenario, in the paper we shall consider also the case of links composed of multiple PICs, where each PIC of the bundled links can be powered off.

III. Problem statement

Let us introduce the parameters and the notation which will be used in the following. The starting point of the analysis is a network modeled as a directed graph $G=(V,E)$, where V denotes the set of the nodes and E is the set of the arcs, modelling the network links.

The following parameters are assumed to be given in order to characterize the power consumption of the network elements:

- P_{uv}^{PIC} is the power consumption of a PIC that transmits traffic from node u to node v (i.e., a PIC of link (u,v));
- P_v^C is the power consumption concerning the chassis of the node v;
- $P_{v,T(v)}^{RP}$ is the power consumption concerning the route processing of the node v at the traffic throughput $T(v)$; as justified before, hereafter we shall assume that $P_{v,T(v)}^{RP}$ is a nonlinear function.

Since each logical link is generally composed of a set of PICs, the overall power consumption for the traffic transmission on a link (u,v) is equal to the number of powered on PICs, in node u connected to v, multiplied by P_{uv}^{PIC}. Hence, the maximum power consumption associated with a directional link (u,v) is given by $N_{uv} \cdot P_{uv}^{PIC}$, where N_{uv} is the maximum number of PICs forming the bundled link that connects node u to node v.

Concerning the traffic demand and the capacity of nodes and links, respectively, we define the following parameters:

- D is the set of the origin-destination pairs of the traffic matrix;
- d_{sd} is the traffic demand between the source node s and the destination d;
- C_v^N is the capacity of node v;
- C_{uv}^{PIC} is the capacity of each PIC which composes the link (u,v).

Two sets of variables are defined:

- f_{uv}^{sd} is the amount of d_{sd} flowing through the link (u,v);
- n_{uv} is the number of powered on PICs which compose the link (u,v).

The traffic throughput of node v can then be defined as the total traffic entering v plus the flow originated from v, according to the following formula:

$$T(v) = \sum_{(u,v)\in E} \sum_{(s,d)\in D} f_{uv}^{sd} + \sum_{(v,d)\in D} d_{vd}. \quad (1)$$

To reduce the power consumption of the backbone network with bundled links, we defined the *PARND-BNBL* problem. This minimization problem considers both the possibility of routing the traffic demands by taking into account the relation between the power consumption of the route processor and the traffic load, and of powering off the links, as well as even single PICs of bundled links. In the problem, since we are considering a backbone scenario, for reliability reasons we excluded the possibility to power off the network nodes.

The above introduced PARND-BNBL problem leads to a Mixed Integer NonLinear Programming (MINLP) design and routing model, which can be formulated as follows:

$$\text{minimize} \sum_{v\in V} P_v^C + \sum_{v\in V} P_{v,T(v)}^{RP} + \sum_{(u,v)\in E} P_{uv}^{PIC} \cdot n_{uv} \tag{2}$$

subject to the constraints

$$\sum_{(v,u)\in E} f_{vu}^{sd} - \sum_{(u,v)\in E} f_{uv}^{sd} = \begin{cases} d_{sd} & \text{if } v = s \\ -d_{sd} & \text{if } v = d \qquad \forall (s,d) \in D \\ 0 & \text{otherwise} \qquad \forall v \in V \end{cases} \tag{3}$$

$$f_{uv}^{sd} \in \mathbb{R}^+ \quad \forall (u,v) \in E \quad \forall (s,d) \in D \tag{4}$$

$$T(v) \leq C_v^N \quad \forall v \in V \tag{5}$$

$$f_{uv} = \sum_{(s,d)\in D} f_{uv}^{sd} \leq C_{uv}^{PIC} \cdot n_{uv} \quad \forall (u,v) \in E \tag{6}$$

$$n_{uv} \in \mathbb{N}_0 \quad n_{uv} \leq N_{uv} \quad \forall (u,v) \in E. \tag{7}$$

Equations (3) are the classical flow conservation constraints, instead Equations (4) provide the definition of the flow variables. Equations (5) and (6) are the node and the link capacity constraints, respectively. Finally, the constraints (7) guarantee that even single PICs of bundled links can be powered off.

Since there are no commodity-dependent costs or capacities in the proposed model, in order to reduce the computational effort we indeed considered and implemented aggregated versions of the flow variables, where all the commodities having the same origin node are considered to be "the same kind of flow".

IV. Related Works

In past few years, a lot of works have addressed the Network Power Management problem from different points of view. All these works focus on minimizing the overall power/energy consumption of the network, but everyone considers a different power model of the network elements.

In [3] the authors have proposed the problem of routing the traffic demands in such a way as to minimize the overall power consumption of the network by considering different functions to represent the power consumption of the routers and solved it by using a linear programming solver (e.g CPLEX [9]), by considering coarse linear approximations of power consumption of the nodes, which is generally a nonlinear function. In [10] the authors have proposed a heuristic solution based on Dijkstra's algorithm, and have deeply investigated the impact on the performance of the traffic load and the network topology.

In [4] the authors have considered the possibility to power off nodes and links of the network in order to reduce the overall power consumption and compared diverse heuristics to solve the corresponding problem.

Successively, other papers [11] [12] [13] have tried to combine the design and the routing issues by considering at the same time the power consumption of route processing and the opportunity of powering off nodes and links, but not individual cables in the bundle. In [11] the authors have assessed this issue by assuming that nodes and even links have a linear energy behavior. They have solved their problem by using CPLEX. In [12] the authors have considered only the link energy consumption, which is linearly proportional to the traffic that flows through. They have solved this problem by means of a heuristic. Differently from [11] and [12], in [13] the authors have proposed an energy model of a router based on [6] and a heuristic solution of this model. In particular, the considered energy model is given by the sum of two constant addends associated with the power consumption of the chassis and the line cards, and one variable (nonlinear) element associated with the route processor.

In [14] the authors have focused on a problem similar to PARND-BNBL, but they have only considered the opportunity of powering off individual cables in the bundle. They have proposed and compared different heuristics and the results highlighted that the performances differ by at most a couple of percentage points among the considered heuristics. Hence, we will consider the simplest and fastest of them, denoted as Fast Greedy Heuristic (FGH), as benchmark in this paper.

To the best of our knowledge, none has proposed a Network Power Management model as PARND-BNBL that permits to minimize the overall power consumption of a backbone network by choosing a traffic routing strategy that jointly exploits the ability of powering off all (or even single) cables of a bundled link and the power behavior of route processors.

V. Heuristics to solve PARND-BNBL

For solving PARND-BNBL we propose a new heuristic (see Algorithm 1), which is based on the above-mentioned FGH approach.

The FGH heuristic is based on the Maximum Spare Capacity (MSC) problem, which is used as a building block.

Taking into account the notations introduced in the previous section, the MSC problem can be formulated as follows:

$$\text{minimize} \sum_{(u,v)\in E} f_{uv} \tag{8}$$

subject to the Equations (3), (4), (5), and

$$f_{uv} = \sum_{(s,d)\in D} f_{uv}^{sd} \leq C_{uv}^{L} \quad \forall (u,v) \in E, \tag{9}$$

where C_{uv}^{L} is the capacity of the link (u,v) (in the case of bundled link $C_{uv}^{L} = C_{uv}^{PIC} \cdot n_{uv}$, where n_{uv} is the number of active PICs). Obviously, Equations (9) ensure that no link carries more traffic flow than its capacity.

The FGH heuristic consists in solving the MSC problem to obtain the flow f_{uv} assigned to each edge. Then, taking into account that all the flows are still satisfying the capacity constraints, see Eq. (9), the maximal number of cables are removed. After these cable removals, the edge with the greatest spare capacity is identified, i.e., we find the (u, v) for which:

$$\arg\max_{(u,v)} (C_{uv}^{PIC} \cdot n_{uv} - f_{uv}) \tag{10}$$

where n_{uv} denotes the number of remaining cables after the cable removals. In order to reduce the excess traffic to be rerouted, only one cable of (u,v) is considered for the removal. This cable is temporarily removed, and then the MSC is solved with the new link capacities. If the problem has a feasible solution, the cable considered for removal is permanently removed, otherwise it is not removed and the corresponding edge is marked as final (FE), in the sense that no additional cables are removed from final edges. In fact, in the next iterations, the identification of the edge with the greatest spare capacity is performed ignoring all final edges. The iterative procedure concludes when all edges become final.

The choice of basing our heuristic on FGH is due to its attractiveness given by its simplicity. The main differences with respect to FGH are that at each iteration a cable is removed if the overall power consumption is actually reduced and the Power Aware Routing (PAR) problem is used as a building block in place of MSC. These differences are particularly relevant when the power consumption due to route processing is dependent on the node throughput. In this case, the power consumption due to route processing has to be explicitly addressed in the problem formulation, and just to power off links/PICs does not necessarily determine a reduction of the overall network consumption.

The PAR problem consists in determining the traffic routing strategy that permits to reduce the overall power consumption of the network by taking into account only the power consumption of the nodes concerning the route processor, i.e. $P_{v,T(v)}^{RP}$. The PAR problem can be formulated according to the following nonlinear multicommodity flow model:

$$\text{minimize} \sum_{v\in V} P_{v,T(v)}^{RP} \tag{11}$$

subject to the Equations (3), (4), (5) and (9).

The general structure of the heuristic framework is depicted by Algorithm 1. In the following, we present two different versions of this framework, each one based on a different solution of the PAR problem.

Algorithm 1 PARND-BNBL Heuristic

1: Solve PAR $\Rightarrow (status, Power\_Consumption)$;
2: Remove cables to match flows and set $FE = \varnothing$;
3: Initialize $Prev\_cons = Power\_Consumption$;
4: **repeat**
5:   Sort edges (greatest spare capacity);
6:   **repeat**
7:     Select the first edge $\notin FE \rightarrow (u,v)$;
8:     Disable one cable from selected edge (u,v);
9:     Solve PAR $\Rightarrow (status, Power\_Consumption)$;
10:     **if** ($status == Solve\_Succeeded$ AND $Power\_Consumption < Prev\_cons$) **then**
11:       Remove cable from (u,v);
12:       Update $Prev\_cons = Power\_Consumption$;
13:     **else**
14:       Enable cable and $FE = FE \cup (u,v)$;
15:     **end if**
16:   **until** ($status == Solve\_Succeeded$ AND $FE == E$)
17: **until** ($FE == E$)

A. Time Limited PAR Heuristic (TLPH)

In the first version of the heuristic, the PAR has been solved by means of IpOpt [15], a software package for large-scale nonlinear optimization that implements an interior-point line-search filter method, considering the actual power consumption of nodes and finding the optimal solution.

It has been observed that this approach is time expensive because, when there is not a solution to the PAR problem, the IpOpt requires a lot of time to establish it. Therefore, to speed up the proposed heuristic, the running time of IpOpt for solving PAR has been bounded as following:

- At step 1 of the Algorithm 1, calculate the running time for solving PAR $\rightarrow T$;
- At step 9, bound the running time for solving PAR to $(T+1s)*\gamma$, where γ is a multiplicative factor.

The drawback of this method is that the time can expire even if PAR can be solved ("false negative" event). Hence, a suitable value of the multiplicative factor γ should be chosen to limit both time and false negative events.

B. PAR Meta Heuristic (PMH)

In order to make the heuristic approach faster and faster, we propose a new simple strategy to solve the PAR problem, called HPAR and summarized in Algorithm 2. The algorithm is very simple and consists in considering one origin-destination $(s,d) \in D$ at a time as follows. After the initialization of the node loads $T^{I}(v) = 0 \ \forall v \in N$, the flows $f_{uv}^{I} = 0 \ \forall (u,v) \in E$, and the traffic demands $d_{sd}^{I} = d_{sd} \ \forall (s,d) \in D$, the origin-destination pair having the maximum demand, i.e. $(s^M, d^M) | d_{s^M d^M}^{I} \geq d_{sd}^{I} \ \forall (s,d) \in D$, is calculated. Then, a suitable cost $w(v)\ \forall v \in N$ is set and a path from s^M to d^M, P, is computed by using a Minimum-cost Capacity-constrained Routing (MCCR), which finds (if it exists) the minimal cost ($w(v)$)-path between two nodes that satisfies the link and node constraints, see Eq. (5) and (6). If the MCCR finds a feasible solution, the values of the node loads $T^I(v) += d^I_{s^M d^M}\ \forall v \in P$, the flows $f^I_{uv} += d_{s^M d^M}\ \forall (u,v) \in P$, the traffic demand $d^I_{s^M d^M} = 0$ are updated, otherwise the algorithm ends. Successively, the new maximum traffic demand is computed, if it is equals to 0 the algorithm ends, otherwise the algorithm begins a new iteration.

Algorithm 2 HPAR

```
Initialize T^I(v) ∀v ∈ N, f^I_uv ∀(u,v) ∈ E, and d^I_sd;
Calculate maximum demand ⇒ (s^M, d^M);
repeat
    Set cost w(v) = P^RP_{v,T^I(v)+d^I_{s^M d^M}} − P^RP_{v,T^I(v)} ∀v ∈ V;
    Solve MCCR(w, T^I(v), f^I_uv) from s^M to d^M ⇒ (P, status);
    if (status != Solve_Succeeded) then
        break;
    end if
    Update T^I(v) ∀v ∈ P, f^I_uv ∀(u,v) ∈ P and d^I_{s^M d^M};
    Calculate maximum demand ⇒ (s^M, d^M);
until (d^I_{s^M d^M} = 0)
```

Therefore, PAR Meta Heuristic (PMH) consists in the Algorithm 1 where PAR problem is solved by using HPAR.

VI. Network scenarios

The performance analysis of the presented problems has been carried out referring to a set of core network scenarios. The first considered scenario is a European core network topology obtained from the Simple Network Description Library [16]; in particular, we used the file *nobel-eu*. Other network scenarios are backbone topologies obtained from the set of data collected during the Rocketfuel study [17]. In particular, we considered the following topologies: Exodus (US), Ebone (EU), Abovenet (Australia), and Sprintlink (US), which correspond respectively to the dataset AS 3967, AS 1755, AS 6461, and AS 1239 of the Rocketfuel study. The last considered network scenario is a large Austrian core topology (ta2) taken from [16] and given by the Telekom Austria.

The nodes of all the tested networks are assumed to have the same energy profile, and the links are supposed to be symmetric, i.e. $C^{PIC}_{uv} = C^{PIC}_{vu}$, $P^{PIC}_{uv} = P^{PIC}_{vu}$, and $n_{uv} = n_{vu}\quad \forall (u,v) \in E$. This last assumption implies that all the tested networks are indeed composed of undirected links, which are modelled in terms of two directed links, one for each direction.

Table I
Statistics of the Network Scenarios

	nobel-eu	Exodus	Ebone	Abovenet	Sprintlink	ta2
# Nodes	28	22	23	22	43	65
# Links	41	37	38	42	83	108
# S.L.N.	0	1	4	5	13	1

Table I shows the statistics of the considered networks, i.e. the number of the nodes (# *Nodes*), the number of the undirected links (# *Links*), and the number of the nodes which are connected to the rest of the network with a single undirected link (# *S.L.N.*).

In all network scenarios, each node represents a core router. We assumed the use of the Juniper T1600 core router, having a total throughput capacity of 1600Gb/s and a maximum power consumption of 8352W [18]. Thus, all the nodes of the networks have the same energy behavior. For each link we assumed to use multiple SONET/SDH OC768c/STM256 PICs. Each one has a payload bandwidth of $38.486 Gb/s$ and a power consumption of 65.7W [19].

The results reported in [5] show that the power consumption of a chassis is equal to about 200W for all classes of routers; thus, for each node v we set $P^C_v = 200W$.

In summary, we have considered the following setting for the model parameters:

- $P^{MAX}_v = 8352W$ and $C^N_v = 1600 Gb/s\ \forall v \in V$,
- $P^{PIC}_{uv} = 65.7W$ and $C^{PIC}_{uv} = 38.486 Gb/s\ \forall (u,v) \in E$,
- $P^C_v = 200W\ \forall v \in V$,

where P^{MAX}_v is the maximum power consumption of node v.

Concerning the power consumption component of the route processing for the energy characterization of the devices, we focused our attention on a *cubic curve* (see Figure 1), since it represents the state-of-the-art of circuit-level energy-efficient mechanisms [3]. In particular, the cubic curve gives the energy behavior of network equipments that use energy savings techniques such as Dynamic Voltage and Dynamic Frequency Scaling (DVS-DFS), which permit energy consumption to scale with resource requirements.

Current routers do not implement such techniques and are very energy inefficient, but we assume that they could be implemented in the next generation routers. Based on the previously cited parameters of the actual routers, we have thus defined the power consumption concerning the route processing as follows:

$$P^{RP}_{v,T(v)} = \frac{P^{MAX}_v - P^C_v}{(C^N_v)^3} \cdot T(v)^3 \qquad \forall v \in V. \tag{12}$$

Concerning the origin-destination demands, for the *nobel-eu* topology the traffic matrix has been obtained from the data file "Nobel-2 directed graph" downloaded from [16]; the file contains the undirected traffic demand between each couple of nodes of the considered network scenario. To obtain the directed traffic demands, we have randomly split the demand between the two directions. The total amount

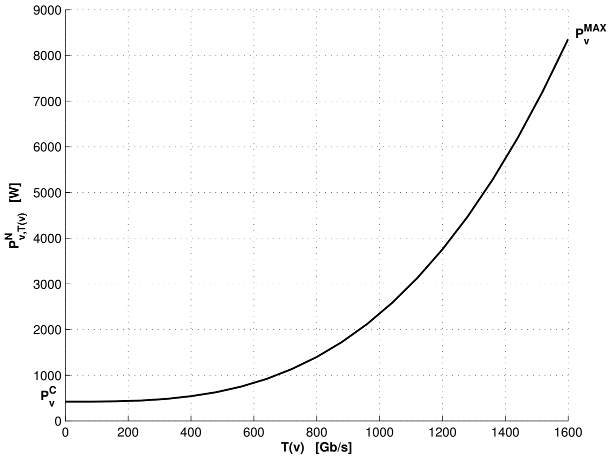

Figure 1. Power consumption of node v, $P^N_{v,T(v)} = P^C_v + P^{RP}_{v,T(v)}$

of traffic demand is $1898 Gb/s$, distributed among the 756 origin-destination pairs (i.e. the pairs $(s,d) \in D$). The mean traffic demand of an active node toward all the other nodes is about $\overline{d^s} = 67.8 Gb/s$.

To obtain the traffic matrix for the other topologies acquired from the Rocketfuel study and Telekom Austria, we set the mean value of traffic demand from s to d, i.e. $\overline{d_{sd}}$, to $\frac{\overline{d^s}}{|V|-1}$. Then, each element of the matrix d_{sd} has been extracted from a uniform distribution: $d_{sd} = U\left[0.5 \cdot \overline{d_{sd}},\ 1.5 \cdot \overline{d_{sd}}\right] \quad \forall (s,d) \in D$. The choice of a traffic matrix where each node of the network is both source and destination of traffic implies that only links (or PICs) can be powered off.

Finally, as far as the link capacity is concerned, the maximum number of PICs per link has been computed as follows:

$$N_{uv} = \lceil \frac{\max(f_{uv}, f_{vu})/\beta}{C^{PIC}_{uv}} \rceil \quad \forall (u,v) \in E, \qquad (13)$$

where $f_{ij} = \sum_{(s,d)\in D} f^{sd}_{ij}$ denotes the total flow on link (i,j) when the Shortest Path Routing (SPR) is applied, while $\beta = 0.5$ is the overprovisioning factor (see [4]). In more details, to calculate the f^{sd}_{ij} we assumed that each traffic demand d_{sd} uses a single shortest path from s to d. The weight of each link is assumed equal to 1. Therefore, a minimal cardinality path has been computed for each origin-destination pair (s,d). For the tested, symmetric, networks we set $N_{uv} = N_{vu}$ for each undirected link (u,v).

Table II shows the statistics about the number of PICs per link for the different topologies, i.e. the maximum (*Max*) and the average (*Avg*) number of PICs per link, and the number of links composed of a single PIC (# *S.P.L.*).

VII. Performance analysis

In the following a performance analysis of the different approaches is presented. The power aware methods have been compared to SPR, which represents in fact the widely used approach in core networks when no specific administrative or cost constraints are present, and to FGH, on which our proposed heuristics are based.

Table II
Number of Line Cards per Link

	nobel-eu	Exodus	Ebone	Abovenet	Sprintlink	ta2
Max	16	7	11	7	23	23
Avg	4.66	3.46	3.58	2.5	3.46	4.7
# S.P.L.	7	4	8	12	15	20

The *PARND-BNBL* problem has been solved by the mixed integer nonlinear programming solver BONMIN 1.5.0 (only for small instances) or by means of a linear approximation using the IBM ILOG CPLEX Optimization Studio V12.2 [9], whereas the solution of the PAR problem has been calculated by using the IpOpt version 3.6stable [15]. The linear approximation (used for addressing the large PARND-BNBL instances) consists in approximating the cubic function representing the power consumption of the route processing, i.e. $P^{RP}_{v,T(v)}$ (see Figure 1), via a piece-wise linear function composed of 20 segments, as in [10]. When possible, we also found the optimal solution of the nonlinear PARND-BNBL models by using BONMIN 1.5.0 [20]. The results have shown that the power consumption difference between the solutions are order of 0.01%. However, the computational time needed to BONMIN for determining the optimal solution is often two order of magnitude higher than the time spent by CPLEX to calculate the solution of the linear approximation. Furthermore, in some cases, BONMIN produced no results after 160 hours of CPU time. As a consequence, in the remaining of the study we shall consider only the linear approximation of PARND-BNBL.

In order to provide Quality of Service (QoS) solutions, in all the tested models we have limited the link utilization by multiplying C^{PIC}_{uv} by a factor $\rho \in (0,1)$ in the link capacity constraints (9) and (6). The value of ρ, which represents the link utilization, has to be appropriately determined in order to guarantee the QoS (i.e. a limited delay). To this aim, we have modeled the transmission of the traffic on the link as a M/M/1 queue, where the service rate μ is the link capacity. By considering that the minimum link capacity of the network is C^{PIC}_{uv} (for those links which are composed of a single PIC), the mean delay is equal to $\frac{1}{C^{PIC}_{uv} \cdot (1-\rho)}$. Therefore, if we consider $\rho = 0.95$, the mean delay is equal to $51.97 ns$, which is widely sufficient to guarantee a low end-to-end delay.

By setting $\rho = 0.95$, we then analyzed the performance and the behavior of the different approaches, on the various topologies, by studying the following indicators:

- overall power consumption of the network;
- number of powered on links/PICs;
- CPU time.

It is relevant to note that in the analysis we consider the pure FGH, i.e. calculated setting $\rho = 1$. The results obtained with $\rho = 0.95$ will be indicated by FGH-QoS.

A. Evaluation of TLPH

A first study was aimed to evaluate the impact of the multiplicative factor γ on the performance of the TLPH. Table III shows the power consumption and the CPU time for calculating the solution of TLPH. The results refer to all considered topologies and to some values of γ (the symbol ∞ denotes the case where no time bound is set).

First of all, we have to point out that the time scale for finding the solution to apply in the actual network should be day/night, i.e. peak/off-peak traffic periods, because there are significant changes in the traffic matrix only between the peak hours and the off-peak hours. Furthermore, we can assume that the traffic data of the last peak period (or off-peak period) is used to estimate the traffic matrix in the next peak period (or off-peak period). Therefore, if we consider that the duration of the peak/off-peak period is order of day/night, the maximum acceptable CPU time for computing the solution is in order of 6/12 hours.

The results highlight that TLPH is very slow when the running time for the solution of the PAR problem is not limited, for example the CPU time of $108163s$ obtained for the case of the ta2 topology is unacceptable in an actual network scenario.

Further, the results show that bounding the time using $\gamma \in \{4,5\}$ leads to obtain the same power consumption obtained with $\gamma = \infty$, but in a time about 10 times smaller. Using smaller values of γ, the power consumption of the obtained solution is the same, but the CPU time reduces up to 50 times for small topologies (i.e. nobel-eu, Exodus, Ebone, and Abovenet). For the Sprintlink and the ta2 topologies, the differences in terms of the power consumption given by the obtained solution for diverse values of γ are due to the presence of the false negative events in the PAR computation. Furthermore, in the ta2 topology for $\gamma \in \{2,3\}$ the false negative events leads to a reduction of the power consumption. This phenomenon is due to the fact that, in the initial iterations of Algorithm 1, there was a false negative event, i.e. some PICs of a bundled link remain powered on. In the successive iterations, this event could allow to power off "new" PICs and, consequently, to save more power. In order to better understand this phenomenon, we should take into account that FGH approach used as the basis of the proposed heuristics presents the disadvantage that if the algorithm makes the "wrong" choice by removing a suboptimal PIC, it will never backtrack to correct the mistake.

In summary, we can observe that TLPH permits to achieve a good trade off between CPU time and power consumption by using $\gamma = 2$. This value will be used in the following experiments.

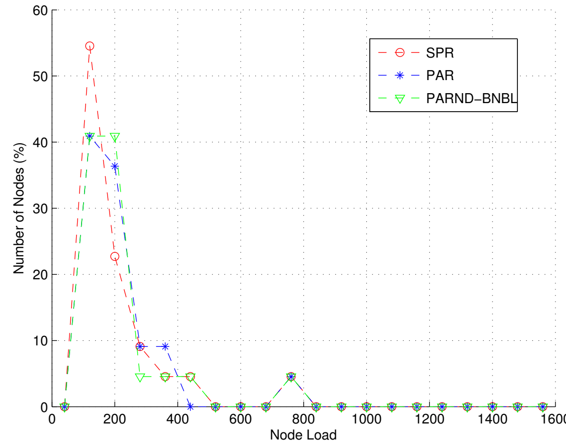

Figure 2. Histograms of the load of nodes with the SPR, PAR and PARND-BNBL approaches - Abovenet Topology

B. Overall power consumption of the network

The results of the network power consumption (in W) obtained with the different approaches are summarized in Table IV. In the case of the ta2 scenario, the utilization of the SPR strategy violates the constraints on the node capacity, since there are a lot of paths traversing some nodes in the middle of the topology. In general, the results show that the power savings produced by PAR with respect to SPR are almost negligible. Indeed, the maximum power savings is about 13.4% (for the ta2 topology), whereas the minimum is about 0.4% (for Abovenet and Exodus scenarios). The low power savings are mainly due to the low load of the networks, which implies that the nodes have on average a throughput value where the function $P^N_{v,T(v)}$ is almost linear and with a low slope.

As an example, Figure 2 shows the histograms of the load of nodes when the SPR, PAR and PARND-BNBL approaches are used in the Abovenet scenario. The figure shows that the majority of nodes (above the 95%) have a throughput under 500 Gb/s, whereas the remaining nodes work under the 800 Gb/s. By taking into account the function $P^N_{v,T(v)}$, shown in Figure 1, we can observe that these throughput values are in a quite linear region of the curve, with a very low slope. This observation can explain the low power savings produced by the PAR solution with respect to SPR. We observed a similar behavior in the other scenarios; these results are omitted for the sake of brevity.

Table IV clearly shows the power savings due to the PARND-BNBL approach and, in the case of large topologies (i.e. Sprintlink and ta2), the gap between the values of the solutions found by the proposed heuristics and the optimal solution value. Furthermore, we can observe that in the ta2 topology we can not find the optimal solution, but the software produced the error "out of memory" although the PC had a 8GB RAM. On the contrary, we determined

Table III
POWER CONSUMPTION AND CPU TIME OF TLPH VS γ

γ	nobel-eu	Exodus	Ebone	Abovenet	Sprintlink	ta2
∞	19816W in 4722s	12839W in 2556s	14580W in 4269s	12426W in 2803s	31135W in 55931s	61317W in 108163s
5	19816W in 627s	12839W in 375s	14580W in 382s	12426W in 387s	31135W in 4575s	61317W in 10694s
4	19816W in 507s	12839W in 303s	14580W in 308s	12426W in 313s	31135W in 4047s	61317W in 8781s
3	19816W in 388s	12839W in 231s	14580W in 235s	12426W in 240s	31135W in 3150s	61225W in 6841s
2	19816W in 267s	12839W in 87s	14580W in 160s	12426W in 165s	31252W in 2183s	61225W in 5161s
1	19816W in 147s	12839W in 87s	14580W in 86s	12426W in 91s	31956W in 1184s	62593W in 2650s

solutions using the considered heuristics, except in the case of nobel-eu scenario with PMH strategy. In this case, the heuristic used to solve the PAR problem was unable to allocate some traffic demands, since it over-utilizes the links given that it creates very long paths.

In general, we can observe that the proposed heuristics have similar performance of the FGH approaches when the topology is small. When the network scenario is more complex, such as in the ta2 topology, the proposed TLPH and PMH algorithms provide a power savings of about 14% with respect to the simple FGH approaches.

C. Number of powered on links/PICs

Table V shows the number of powered on links/PICs after the application of the solutions provided by the different approaches. The results highlight that the proposed TLPH and PMH heuristics produce the highest number of links (and PICs) powered on in all the considered topologies. However, even if the other solutions power off more links and PICs, they do not permit to produce more power savings than TLPH and PMH algorithms. This phenomenon confirms that, in the case of router hardware that implements power aware techniques, such as DVS and DFS, the powering off of links/PICs does not always determine a reduction of the overall network consumption.

D. CPU time

Table VI shows the CPU times required by the different approaches. The measurements have been carried out using a PC with 4 Intel Core i7 CPUs @ 3.07GHz (hyperthreading enabled), 8GB RAM, and an ASUS P6T DELUXE V2 Motherboard. The results highlight that the running time for the optimal solution of the PARND-BNBL problem can be unacceptable in some network scenarios, such as Sprintlink or ta2. On the contrary, we can observe the gain in term of CPU times given by the FGH approaches and the very short CPU time required by the PMH approach in finding a solution. The TLPH approaches produces interesting CPU time gains with respect to the FGH approach, although these results are worse then those of PMH. However, this worsening in terms of CPU time represents the cost to obtain a more energy efficient solution with respect to the TLPH, as shown by the results in table IV. As an example, in the case of the ta2 scenario, the power consumption obtained by the TLPH solution is about 7.5% lesser than the one produced by PMH, although it requires a CPU time almost 100 times higher.

VIII. CONCLUSIONS

In this paper, we have presented the PARND-BNBL approach, which is aimed to reduce the overall power consumption of a backbone network by considering the power consumption of the route processor and the possibility of powering off the links, as well as even single PICs of bundled links. Furthermore, we have presented two variations of the FGH heuristic, called as TLPH and PMH, developed in order to reduce the CPU time for finding a solution of the PARND-BNBL problem. The simulation analysis highlighted the improvements introduced by the TLPH and PMH algorithms both in terms of power savings and CPU times. These improvements are particularly relevant in the case of complex network scenario. Finally, the simulation results point out that in the case of a cubic energy profile of the route processor, the strategy of maximizing the number of powered off links/PICs may not provide the best results in term of power savings.

ACKNOWLEDGMENTS

This work has been partially supported by the EFFICIENT project funded by MIUR.

REFERENCES

[1] M. Mandviwalla and N. Tzeng, "Energy-efficient scheme for multiprocessor-based router linecards," in *Proceedings of the International Symposium on Applications and the Internet (SAINT 2006)*, 2006.

[2] B. Zhai, D. Blaauw, D. Sylvester, and K. Flautner, "Theoretical and practical limits of dynamic voltage scaling," in *Proceedings of the 41st Annual Design Automation Conference (DAC 2004)*, 2004.

[3] J. Restrepo, C. Gruber, and C. Machuca, "Energy Profile Aware Routing," in *Proceedings of the First International Workshop on Green Communications (GreenComm)*, June 2009.

[4] L. Chiaraviglio, M. Mellia, and F. Neri, "Minimizing ISP Network Energy Cost: Formulation and Solutions," *IEEE/ACM Transactions on Networking*, vol. PP, pp. 1–14, July 2011.

[5] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright, "Power awareness in network design and routing," in *Proceedings of the 27th IEEE Conference on Computer Communications (INFOCOM 2008)*, 2008.

Table IV
POWER CONSUMPTION (W) VS TOPOLOGY

	nobel-eu	Exodus	Ebone	Abovenet	Sprintlink	ta2
SPR	33704	22200	24565	19576	52966	108287
PAR	32856	22120	23706	19504	51166	93820
PARND-BNBL	19336	12841	14497	12307	30761	N/A
FGH	19245	12685	14790	12223	32246	70021
FGH-QoS	19813	12988	14935	12567	32717	71036
TLPH	19816	12839	14580	12426	31252	61225
PMH	N/A	13596	15130	12912	32323	66195

Table V
NUMBER OF POWERED ON LINK/PICS

	nobel-eu	Exodus	Ebone	Abovenet	Sprintlink	ta2
PARND-BNBL	41/87	36/57	34/65	34/49	75/130	N/A
FGH	40/83	34/55	36/64	33/47	74/127	100/227
FGH-QoS	39/87	34/57	35/65	33/49	72/129	99/239
TLPH	40/91	36/57	35/66	35/50	75/133	105/255
PMH	N/A	35/62	37/70	38/54	81/140	108/283

Table VI
CPU TIMES (S)

	nobel-eu	Exodus	Ebone	Abovenet	Sprintlink	ta2
SPR	0.01	0.01	0.01	0.01	0.09	0.41
PAR	2.20	0.79	0.99	0.57	11.59	17.77
PARND-BNBL	305.94	648.25	111.55	561.27	45437.32	N/A
FGH	4933.20	2240.68	4472.87	2175.40	49450.94	84827.68
FGH-QoS	3981.52	2194.84	3937.29	2412.73	46421.20	108556.36
TLPH	266.52	158.45	160.37	164.31	2014.31	4878.30
PMH	N/A	0.40	0.37	0.46	8.95	57.21

[6] F. Idzikowski, "Power consumption of network elements in IP over WDM networks," in *TKN Technical Report TKN-09-006*, July 2009.

[7] R. Doverspike, K. Ramakrishnan, and C. Chase, "Structural overview of ISP networks," *Guide to Reliable Internet Services and Applications*, pp. 19–93, 2010.

[8] IEEE Computer Society, "IEEE Standard 802.1AX: Link Aggregation," 2008.

[9] "IBM ILOG CPLEX Optimizer." Available at http://www.ibm.com/software/integration/optimization/cplex-optimizer/.

[10] R. G. Garroppo, S. Giordano, G. Nencioni, and M. Pagano, "Energy Aware Routing based on Energy Characterization of Devices: Solutions and Analysis," in *Proceedings of the 4th International Workshop on Green Communications (GreenComm4)*, June 2011.

[11] A. Bianzino, C. Chaudet, F. Larroca, D. Rossi, and J. Rougier, "Energy-aware routing: A reality check," in *Proceedings of the 3rd International Workshop on Green Communications (GreenComm3)*, 2010.

[12] S. Antonakopoulos, S. Fortune, and L. Zhang, "Power-aware routing with rate-adaptive network elements," in *Proceedings of the 3rd International Workshop on Green Communications (GreenComm3)*, pp. 1428–1432, December 2010.

[13] R. G. Garroppo, S. Giordano, G. Nencioni, and M. G. Scutellà, "Network Power Management: models and heuristic approaches," in *Proceedings of the IEEE Global Communications Conference (Globecom 2011)*, December 2011.

[14] W. Fisher, M. Suchara, and J. Rexford, "Greening Backbone Networks: Reducing Energy Consumption by Shutting Off Cables in Bundled Links," in *Proceedings of the first ACM SIGCOMM workshop on Green Networking*, August 2010.

[15] "Ipopt: Interior Point OPTimizer." Available at https://projects.coin-or.org/Ipopt.

[16] "Simple Network Description Library (SNDlib)." Available at http://sndlib.zib.de.

[17] "Rocketfuel: an ISP topology mapping engine." Available at http://www.cs.washington.edu/research/networking/rocketfuel/.

[18] "Datasheet of Juniper T Series Core Routers." Available at https://www.juniper.net/us/en/local/pdf/datasheets/1000051-en.pdf.

[19] "T1600 Core Router PIC Guide." Available at http://www.juniper.net/techpubs/en_US/release-independent/junos/information-products/topic-collections/hardware/t-series/t1600/pics/t1600-pic.pdf.

[20] "BONMIN: Basic Open-source Nonlinear Mixed INteger programming." Available at https://projects.coin-or.org/Bonmin.