Safe Routing in Energy-aware IP networks

Youcef Magnouche, Jérémie Leguay, Feng Zeng
Huawei Technologies Ltd., France Research Center
Email: firstname.name@huawei.com

***Abstract*—The reduction of greenhouse gas emissions represents one of the major concerns worldwide. In telecommunications, the research community and the industry are developing various solutions to decrease the energy consumption of infrastructures. In this paper, we focus on the energy efficiency of IP networks and, in particular, on energy-aware routing as a good complement to power management mechanisms inside routers. We present a safe energy-aware routing solution where the network controller leverages knowledge about device behaviors for power management to further increase energy savings. As energy saving mechanisms can impact network reliability, decisions are taken to avoid any disruption. First, we propose a mathematical model for the associated optimization problem, and then, we develop a hybrid Benders decomposition and Column generation algorithm to solve it. Finally, we present a performance evaluation on a real backbone network with two possible implementations of the solution.**

***Index Terms*—Energy-aware routing, Reliability, Integer linear programming, Benders decomposition, Column generation.**

I. INTRODUCTION

Global warming has become a major concern, and it is urgent to widely integrate energy-saving mechanisms wherever it is possible. Indeed, the Information and Communications Technology (ICT) sector accounts for between 5% and 9% of the total electricity consumption [1], and it generates between 1.2 and 2.2 Gigatons of CO2-eq per year. In 2016, the "Paris Agreement", an international treaty, has been signed with the promise to decrease greenhouse gas emissions by 45% until 2030 compared to 2010 levels. While energy efficiency has been a concern for a long time in telecommunications, researchers are ramping up their efforts to find new solutions to achieve this challenging target.

A broad variety of solutions have been proposed to improve the energy consumption in networks [20]. Their ultimate goal is to reach the smallest energy consumption as possible and, in particular, a consumption that is *proportional* to the utilization of devices. In fact, in a seminal benchmarking study from 2009, researchers have shown that almost every tested device demonstrates a non-proportional behavior. To mitigate this issue, in recent years, several mechanisms have been developed in protocols and devices. The most popular one is Energy-Efficient Ethernet (EEE), a collection of Ethernet standards including 802.3az [12; 23], which reduces power consumption during periods of low activity. It allows each port on a switch to power down if no device is connected, and to adjust the link rate based on traffic conditions. It can also detect the cable length and adjust the power used for transmission accordingly. Inside the devices themselves, power management has become a key feature, and it is rapidly evolving thanks to new hardware technologies [21]. Similarly to Dynamic Voltage and Frequency Scaling (DVFS) in processors, power management of hardware components (e.g., ports, serdes, chipsets and boards) in network devices can be accomplished by tuning the clock frequency and/or the number of active circuits to follow traffic variations. Several functional points, simply called *frequencies* in the rest of this paper, are now becoming widely available on every sub-component of routers and switches.

One of the main difficulty with power management is that it can impact reliability and performance. Typically, power management follows an *idle logic* to reduce power consumption by rapidly down scaling sub-components when traffic decreases, and by re-waking them up when traffic increases. However, the down scaling of sub-components generally reduces the available capacity and impacts Quality of Service (QoS). Studies [24] have shown that it increases traffic burstiness and congestion, potentially inducing high delays and packet losses. Turning off completely a sub-component saves more energy than powering it at its smallest frequency, but it may disrupt networking protocols. In the case of Interior Gateway Protocols (IGP), shutting down a physical port modifies the logical topology and affects routing stability.

Energy-aware routing and traffic engineering solutions have emerged as a natural complement to power management mechanisms. Their goal is to steer traffic so that energy gains can be maximized. Indeed, extensions of IGP protocols [9; 10] and centralized Software-Defined Networking (SDN) solutions [13; 20] have been proposed to offload traffic from low efficiency components and to shut them down whenever it is possible. Nevertheless, while distributed routing solutions can ease adoption by network operators, most of them lead to instability issues. For safety reasons, centralized methods are preferred for now. On the other side, centralized solutions mainly rely on metrics related to power consumption or energy-efficiency. They do not optimize routing considering the knowledge of the behavior of devices. Indeed, a finer knowledge about the layout of devices (e.g., list of ports attached to a given chipset/board) and about power management mechanisms (e.g., state machines) of sub-components can help to make better routing decisions in terms of energy-efficiency. In the context of energy-aware traffic engineering, a key feature is to absolutely avoid any disruption in the network on the control plane (e.g., IGP protocol) and the data plane (e.g., packet losses).

In this paper, we present a safe energy-aware routing algorithm for SDN controllers that leverages knowledge about

978-1-6654-7598-3/23/$31.00 ©2023 IEEE

device behaviors for power management to further increase energy savings. It considers in particular the internal layout of routers with their sub-components (boards, chipsets and ports) and the associated power management mechanisms. Its main objective is to engineer traffic, using Segment Routing (SR) [2] policies for instance, while protecting the underlying IGP so that routing tables remain stable. Safety is realized by enforcing a minimum residual capacity for signalling between adjacent routers, i.e. so that logical links at IP layer are not affected. To efficiently optimize routing at large-scale, we propose a hybrid algorithm based on Benders decomposition [7] and Column generation [19]. As a use case for the evaluation, we consider a real high capacity backbone, from a regional service provider in China, where many parallel links between routers are aggregated as Link Aggregated Groups (LAG). We show that the global optimization of routing policies based on the knowledge about device behaviors under safety constraints for the IGP leads to significant energy gains. We compare two candidate solutions where the controller can decide 1) to tune both SR policies and frequencies of sub-components, or 2) to control routing only and let devices self-adapt their frequency. In the later case, a state-of-the-art LAG mechanism locally adjusts frequencies to save energy [17]. We show that a global coordination of frequencies can increase energy savings by 6%, at the cost of a higher complexity.

The rest of this paper is structured as follows. Sec. II reviews the related work. The system architecture and the problem formulation are introduced in Sec. III and IV. Sec. V provides a Benders decomposition to efficiently solve the problem exactly. Sec. VI presents our numerical results and Sec. VII concludes this paper.

II. RELATED WORKS

Several works in the literature have tackled the energy-aware routing optimization problem. They considered different settings in terms of which sub-components to manage, centralized and distributed solutions, heuristic and exact methods.

Centralized solutions. In [8] an optimization problem (ILP) is used to decide which links and nodes to turn-off such that the traffic demand is satisfied and the total energy consumption is minimized. Based on traffic predictions, authors in [3] formulate a robust optimization problem and develop a heuristic to optimize energy consumption by putting in sleep mode unnecessary line cards and chassis. In [25], a traffic engineering problem for MPLS networks is solved to turn-off ports and line cards. A line card can be turned-off only if all its ports are switched-off. The authors propose a mathematical model that ensure that the traffic matrix is satisfied and develop a heuristic (ILP-based) where the candidate paths are precomputed. In [14], a method is proposed for hybrid IP/SDN networks. The authors proposed an ILP model and developed a heuristic to turn off some links to minimize the energy consumption but also the maximum link utilization. For legacy IP networks with the shortest path routing, authors in [11] proposed a method to compute OSPF [18] weights to switch-off some network elements (nodes or links). Similarly, authors in [4] tried to optimize two functions, the energy consumption and the network congestion. They proposed a mathematical model that decides which element (router or line cards) to switch-off, together with the OSPF weights. They also develop a heuristic to solve the problem. While centralized routing solutions have been proposed to integrate the layout of devices, none of the existing methods consider a detailed model of device behaviors, i.e., the layout of devices (e.g., list of ports attached to a given chipset/board) and power management capabilities with several frequencies for each sub-component.

Distributed solutions. Several related works have been proposed on distributed energy-saving protocols that aim at turning on/off links rather than controlling routing. In [9] periodic Link State Advertisements (LSA) are broadcasted in the network describing the state of the links (on/off, power consumption). LSAs are also used to broadcast critical states, e.g., presence of unreachable destinations or performance degradation. All nodes run the same algorithm to identify which link to turn on/off. If LSAs confirm the satisfaction of "routing" and "performance" constraints, one link is selected to be possibly switched off (the least loaded link or the most power hungry). Before putting a link into sleep mode, all nodes check if the network would still be connected after its removal. If LSA reports a constraint violation, nodes react by bringing back to operational state some links. In [10] authors presented a distributed mechanism where every node takes local decisions based only on the knowledge of the current load, power consumption of incident links, and the current network topology. LSA messages distribute information about the current network topology, augmented by information about eventual congestions in the network. Q-Learning algorithm allows the node to iteratively select the best configuration (the incident links to turn on/off) and penalize the configurations that create disconnection and congestion. If a bad decision is taken by a node (due to congestion or disconnection) this latter is regretted, the node comes back to a all-on configuration or the previous one. Compared to our solution, these solutions only consider links (not sub-components) and they are based on "trial and error". Most of these solutions lead to instability issues and are difficult to deploy in practice for safety reasons.

Energy-efficient LAGs. In the presence of parallel links, [17] proposes a mechanism to monitor bandwidth utilization of a Link Aggregation Group (LAG) to adjust the set of active parallel links accordingly. However, only the consumption of associated ports is considered when making a decision to turned-off a physical link. Our solution optimizes LAG configurations with a broader view over multiple LAGs attached to the same sub-components inside devices (e.g., same chipset / board). Therefore, it can decide to shut down a set of physical links belonging to multiple LAGs in order to scale down the frequency of the associated chipset (which typically consumes more than the total set of ports it manages). When selecting links, our solution also ensures safety for the routing protocol by leaving a residual capacity for signalling.

In summary, our work goes beyond state-of-the-art as no

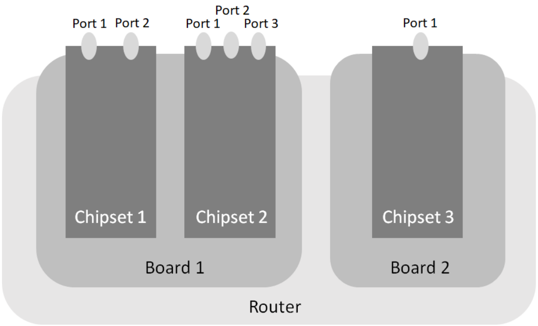

Fig. 1. Layout of a router: hierarchical relationship between its sub-components (e.g., boards, chipsets, ports).

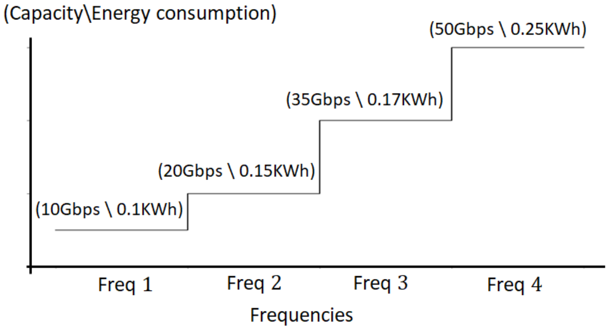

Fig. 2. Capacity and energy consumption for a Serdes at different frequencies.

solution allows to 1) integrate device behaviors (i.e., power management capabilities of sub-components along with the relationship between them) when making centralized traffic engineering decisions (i.e. deciding SR policies) and to 2) consider control plane safety as a constraint (i.e. keep routing tables in the underlying IGP stable).

III. SYSTEM ARCHITECTURE

This section presents the system architecture around the energy-aware routing solution we propose. At routers' level, it will detail the device model based on the relationship between sub-components and the way power management operates in each of them. It will also explain the way parallel links are managed inside LAGs. At the controller level, it will describe how energy-aware routing decisions are taken according to two candidate solutions, under a safety constraint to avoid any disruption in the IGP routing protocol.

Device model. Let consider a network composed of a set of routers (or switches) N. Every node $n \in N$ is equipped with a set of independent boards B_n (i.e., line cards). Every board $b \in B_n$ contains a set of independent chipsets C_n^b. Finally, every chipset $c \in C_n^b$ controls a set of ports P_n^{bc}. As an example, Fig. 1 illustrates the internal and hierarchical layout of a router containing 2 boards. The first board has 2 chipsets and the second one has 1 chipset. The 3 chipsets have a different number of associated ports.

The union of boards, chipsets and ports for all routers in the network represents the set D of sub-devices, i.e., *sub-components*. Every sub-component $d \in D$ has a set of *frequencies* F_d, i.e., a set of operational / functional points, where each frequency allows tuning, i.e., increase or decrease, its capacity and the associated power consumption. At frequency $f \in F_d$, sub-component $d \in D$ consumes an amount of energy $e_d^f \in \mathbb{R}^+$, which does not account for the energy consumption of the sub-components it controls, and it has a capacity $C_d^f \in \mathbb{R}^+$. For each physical link (u, v), we suppose that $F_u = F_v$ and the link capacity $C_{(u,v)}^f = C_u^f = C_v^f$ for $f \in F_u$. This is typically enforced by the Ethernet protocol.

Fig. 2 illustrates, for a Serdes (i.e, port), how the capacity and the energy consumption evolve for different frequencies. Other components like boards and chipsets follow the same principle. However, remarks that, at full speed, the capacity and energy consumption of boards is much higher than that of chipsets, which is, in turn, much higher than that of ports. Furthermore, for all sub-components, the energy consumption at a given frequency is considered as constant. Indeed, while in practice the consumption evolves linearly with the traffic load, the slope is negligible, as observed in several studies [22].

Parallel links. Let $G = (V, E)$ be a network, where V represents the set of ports and E denotes the set of logical links (i.e., IP link). Each logical link between two routers can be composed of multiple parallel physical links that are aggregated at layer 2 as a LAG, also called *trunk* or *bundle*. Each physical link aggregate L is typically managed by the LAG Protocol (LACP) and presented as one logical link at IP layer, which is then used for routing. Fig. 3 presents a toy example with 3 routers (A, B and C). Router A is connected to router B using 2 physical links, both aggregated as a LAG. The configuration is similar for the logical link between routers B and C. Router A and C are connected by a single physical link. The figure highlights the internal layout of router B and shows that LAGs can aggregate physical links that are actually attached to ports belonging to different chipsets / boards. Indeed, ports connecting to A belong to two different chipsets / boards. It also shows which frequencies have been selected (in bold) for each sub-component and their associated capacity. In this example, traffic of 14Mbps needs to be routed from A to C and the best configuration in terms of energy efficiency is displayed in the figure. The traffic is routed over path A-B-C and load balanced inside 2 consecutive LAGs from A to C. The overall consumption is 49Watts.

Energy-aware controller. As one can observe on Fig. 3, if the traffic from router A to C is routed over the direct link, frequency 2 has to be activated on the ports associated with this physical link, while all the other sub-components in the network remain at the same frequency. In this case, the overall consumption is 60Watts, which represents a 22% increase of the overall power consumption. This simply illustrates the benefit of considering the device layout and the knowledge about power management mechanisms of each sub-component when making routing decisions. Therefore, in the rest of the paper, we propose an energy-aware routing architecture where the controller can steer traffic to maximize energy efficiency

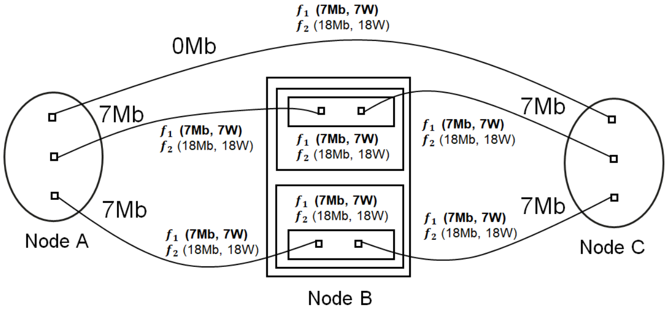

Fig. 3. Example with 3 routers: activated frequencies (with the associated capacity and power consumption) are depicted in bold, along with traffic on each link, to sustain a 14Mbps traffic demand from router A to router C.

based on such knowledge.

The network controller can use Segment Routing [2] or any other traffic engineering mechanism. It decides routing policies for each flow (i.e., traffic aggregate or service) and frequencies of each sub-component. However, two candidate implementations are possible. In the first one, the controller is able to tune the frequencies of each sub-component in routers. In a simpler and more practical implementation, the controller only controls routing policies, and it lets devices locally adjust their frequencies based on traffic.

To preserve stability in the underlying IGP routing protocol, the routing controller enforces that signalling traffic can still pass between two IP routers. This practically means that at least one physical link should remain active at its minimum non-zero rate. If only two frequencies, say ON and OFF, are available on ports associated with all links inside a LAG, at least 1 physical link needs to remain active.

Additional safety mechanisms may be needed to handle unexpected traffic variations and failures. In addition, performance criterions could be embedded into the optimization of routing policies to minimize the congestion or meet end-to-end QoS requirements. However, for the sake of simplicity, we will not consider protection mechanisms as well as additional objectives and constraints in the rest of the paper. Remark that the model and the algorithm can be easily extended to support additional requirements.

IV. PROBLEM FORMULATION

In the following, we present the path mathematical formulation for our energy-aware routing optimization problem.

Let K be a set of commodities such that every $k \in K$ is defined by a source $s_k \in V$, a destination $t_k \in V$ and a traffic demand $b(k) \in \mathbb{R}^+$. For every commodity $k \in K$, let P^k be the set of paths between s_k and t_k. Every path $p \in P^k$ is formed by a sequence of logical IP links, that are each one composed of a LAG, denoted as $L \in E$. The model computes the split ratios for every path of every commodity. Also, it selects the frequency of the sub-components among D. Consider the following variables. Let $x_p^k \in [0,1]$ be the split ratio of commodity $k \in K$ over path $p \in P^k$. Let $t_d^f \in \{0,1\}$ be a binary variable that equals 1 if sub-component $d \in D$ is powered with frequency $f \in F_d$, 0 otherwise. The objective is to minimize the total energy consumption over all the network. The energy optimization problem is equivalent to the following integer linear program.

$$\min \sum_{d \in D} \sum_{f \in F_d} e_d^f t_d^f \tag{1}$$

$$\sum_{p \in P^k} x_p^k = 1 \quad \forall k \in K, \tag{2}$$

$$\sum_{f \in F_d} t_d^f = 1 \quad \forall d \in D, \tag{3}$$

$$\sum_{k \in K} b(k) \sum_{p \in P^k | L \in p} x_p^k \leq \sum_{(u,v) \in L} \sum_{f \in F_d} C_{(u,v)}^f t_u^f \quad \forall L \in E, \tag{4}$$

$$\sum_{c \in C_n^b} \sum_{f \in F_c} C_c^f t_c^f \leq \sum_{f \in F_b} C_b^f t_b^f \quad \forall n \in N, b \in B_n, \tag{5}$$

$$\sum_{p \in P_n^{bc}} \sum_{f \in F_p} C_p^f t_p^f \leq \sum_{f \in F_c} C_c^f t_c^f \quad \forall n \in N, b \in B_n, c \in C_n^b, \tag{6}$$

$$t_u^f = t_v^f \quad \forall L \in E, (u,v) \in L, f \in F_u, \tag{7}$$

$$t_d^f \in \{0,1\} \quad \forall d \in D, f \in F_d,$$

$$x_p^k \in [0,1] \quad \forall k \in K, \forall p \in P^k.$$

The objective function (1) consists in minimizing the energy consumption of the network. Inequalities (2) ensure that the whole traffic demand is sent over the associated paths. Inequalities (3) guarantee that one frequency is selected for every sub-component. Inequalities (4) represent capacity constraints. They also allow linking the t and x variables, in order to select the frequency of every link in the LAG. Constraints (5)-(6) link the frequencies of two embedded sub-components based on their capacities. Finally, Constraints (7) ensure that the two ports of one link have the same frequency.

If a physical link (u,v) admits a frequency f with $C_{(u,v)}^f = 0$ (full shutdown), then the following constraints are required to guarantee network connectivity at the IP layer and avoid IGP disruptions:

$$\sum_{(u,v) \in L} \sum_{f \in F_d | C_{(u,v)}^f \neq 0} t_u^f \geq 1 \quad \forall L \in E. \tag{8}$$

In other words, that at least one link is active in every LAG.

A. Complexity

Theorem 1. *The energy-aware routing optimization problem is NP-Hard.*

Proof. We propose a polynomial reduction to the minimum edge-cost flow problem, known to be NP-hard [15]. Consider a graph $G = (V,E)$ where V is a set of nodes and E a set of links. Let $s, t \in V$ be a source and destination nodes, $c : E \to \mathbb{Z}^+$ be a capacity function and $w : E \to \mathbb{Z}^+$ be a cost function. Fig. 4(a) illustrates G with the link weights. The minimum edge-cost flow problem consists in computing a flow $f \in \mathbb{R}^E$ between s and t such that the capacity constraints are satisfied and the total cost of the active links ($f_e > 0$) is minimum. We construct an instance of the energy-aware routing optimization problem, such that for every link $l = (u,v) \in E$, add a port p_u^l,

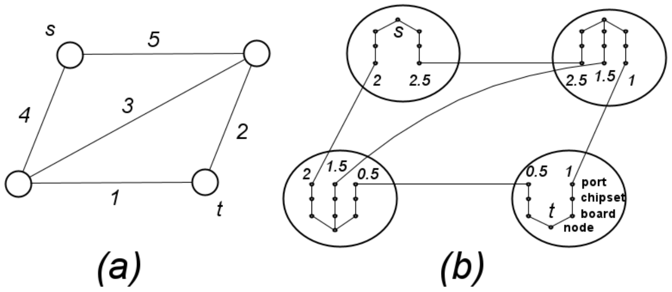

Fig. 4. Transformation of the graph.

a chipset c_u^l, a board b_u^l and a router r_u, that are respectively nested. We do the same for node v. Note that, for both r_u and r_v, the sub-components represent trees where leafs are ports. We consider two frequencies for all sub-components : f_0 with 0 energy consumption and 0 capacity, and f_1 defined as follows:

- energy consumption e_d of every chipset and port d is 0,
- for every link $l = (u, v) \in E$, $e_{b_u^l} = e_{b_v^l} = \frac{w_l}{2}$,
- for every link $l = (u, v) \in E$ add a link $l' = (p_u^l, p_v^l)$,
- l', p_u^l and p_v^l have the same capacity as l,
- the capacity of the board equals the capacity of the included chipset (unique), that in turn equals the capacity of the included port (unique).

Fig. 4(b) illustrates the graph transformation. Let $H = (V', E')$ be the graph obtained from the new instance, where V' is the set of all ports and E' the set of LAGs, such that every LAG contains one link between two ports.

Clearly, solving the minimum edge-cost flow problem in G is equivalent to solving the energy optimization problem in H. $\square$

V. BENDERS DECOMPOSITION

Benders decomposition is a well-known method for solving large-scale combinatorial optimization problems [6]. The algorithm consists in solving the master problem and the sub-problem iteratively. Precisely, the master problem takes the first-stage decisions and the sub-problem checks whether they represent a feasible optimal solution for the whole problem. Otherwise, new constraints, called *Benders cuts*, are added to the master problem, in order to change the first-stage decisions. The Benders decomposition [7] is, generally, used when the model requires two types of variables, main and secondary variables. In this case, instead of solving a unique model with a big number of variables and constraints, Benders decomposition allows solving multiple times small sub-problems. In the energy optimization problem, t represents the main variables and x the secondary variables. Clearly, x is used only to check whether the demand can be satisfied, or not.

A. Sub-problem (MCF problem)

The sub-problem represents a Multi-Commodity Flow (MCF) problem that maximizes the throughput. Indeed, after the decisions of the master problem (frequency selection), the sub-problem checks if a feasible split of traffic exists satisfying all the demands in respect to the capacity constraints.

Consider a continuous solution of the master problem $\bar{t}$. Let $S \in \mathbb{R}^{|K|}$ be the slack variable vector. The sub-problem is equivalent to the following linear program,

$$\min \sum_{k\in K} S_k \tag{9}$$

$$\sum_{p\in P^k} x_p^k + S_k \geq 1 \quad \forall k \in K, \tag{10}$$

$$-\sum_{k\in K} b(k) \sum_{p\in P^k | T\in p} x_p^k \geq \sum_{(u,v)\in T} \sum_{f\in F_d} -C^f_{(u,v)} \bar{t}^f_u \quad \forall L \in E, \tag{11}$$

$$x_p^k \geq 0 \quad \forall k \in K, \forall p \in P^k.$$

The objective function (9) consists in minimizing the total demand constraints violation. Inequalities (10) and (11) represent the demand and capacity constraints, respectively.

Let $\alpha \in \mathbb{R}^{|K|}$ and $\beta \in \mathbb{R}^{|E|}$ be the dual variables associated with constraints (10) and (11).

B. Master problem (frequency selection)

The master problem consists in selecting a frequency for every sub-component $d \in D$. The problem is equivalent to the following integer linear program

$$\min \sum_{d\in D} \sum_{f\in F_d} e_d^f t_d^f \tag{12}$$

$$\sum_{f\in F_d} t_d^f = 1 \quad \forall d \in D, \tag{13}$$

$$\sum_{c\in C_n^b} \sum_{f\in F_c} C_c^f t_c^f \leq \sum_{f\in F_b} C_b^f t_b^f \quad \forall n \in N, b \in B_n, \tag{14}$$

$$\sum_{p\in P_n^{bc}} \sum_{f\in F_p} C_p^f t_p^f \leq \sum_{f\in F_c} C_c^f t_c^f \quad \forall n \in N, b \in B_n, c \in C_n^b, \tag{15}$$

$$t_u^f = t_v^f \quad \forall L \in E, (u, v) \in L, f \in F_u, \tag{16}$$

$$\sum_{L\in E} \beta^L \sum_{(u,v)\in L} \sum_{f\in F_d} C^f_{(u,v)} t_u^f \geq \sum_{k\in K} \alpha_k \quad \forall(\alpha, \beta), \tag{17}$$

$$t_d^f \in \{0, 1\} \quad \forall d \in D, f \in F_d.$$

where (12)-(16) are equivalent to (1)-(7). And, Constraints (17) represent the Benders cuts. Constraints (8) can be added in case where links admit frequencies with empty capacities.

C. Hybrid Benders decomposition and Column-generation

In this section, we describe the hybrid Benders decomposition and Column-generation algorithm (HyBenCg) we developed. The main idea consists in computing a continuous t vector that guarantees the demand constraints. Then, a rounding phase is performed in order to obtain an integer t solution. The algorithm is composed of the following steps:

1) Solve the linear relaxation of the master problem: at this step, a continuous solution is found for the frequency selection problem. The initial model does not contain benders-cuts, leading to a solution where all sub-components are set to the lowest frequency. Often, this results in a violation of the demand constraints.

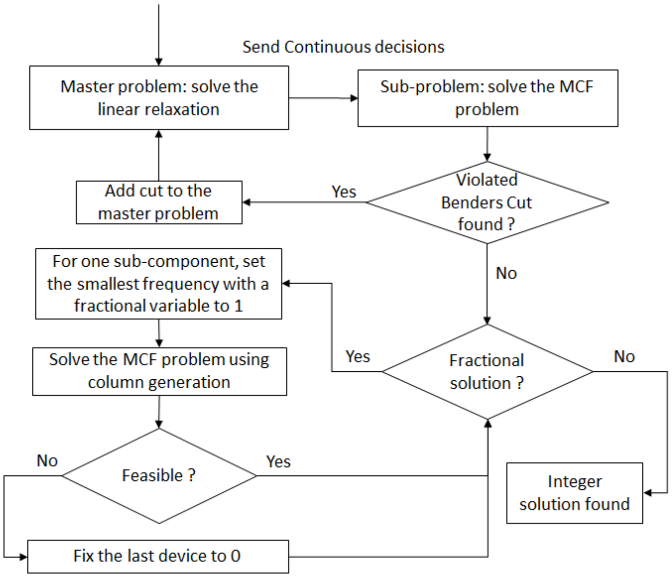

Fig. 5. Benders decomposition and Column Generation: overall algorithm.

2) Solve the sub-problem: this step allows checking if the decisions (continuous) token by the master problem allows satisfying the demand constraints with respect to the capacity constraints. The sub-problem contains an exponential number of x variables. Fortunately, the column generation algorithm [19] allows solving this type of problems, efficiently. It consists in generating some x variables, dynamically, by solving another problem called, the pricing problem. This latter computes, for every tunnel $k \in K$, a path between s_k and t_k minimizing $b(k)\beta^T a_L$ where $a_L \in \{0,1\}$ equals 1 if LAG L belongs to the path, 0 otherwise.
3) Find an integer feasible solution: once a continuous feasible solution is found, it is easy to construct an integer one. Indeed, iteratively, we select the smallest frequency for one sub-component having a fractional value. Then, set this frequency to 1 and all other frequencies to 0. Then, we solve the sub-problem to check if a feasible solution exists. If not, backtrack to the previous solution and set the selected frequency to 0. Note, at this stage, solving the sub-problem is fast, thanks to the already generated good columns.

Fig. 5 displays the whole algorithm diagram. Additional QoS constraints can easily be considered in the proposed algorithm. Indeed, in the associated pricing problem of the column generation (step 2), it suffices to solve a constrained shortest path problem to generate columns satisfying end-to-end delay constraints, for instance. Furthermore, protection can be guaranteed by avoiding to turn-off some sub-components. One can also adapt the sub-problem (9)-(11) to consider a pair of disjoint primary and backup paths and ”reserve” bandwidth for backup paths.

For unsplittable routing, an additional rounding phase can be performed to generate an integer routing solution. Or, the algorithm can be embedded, easily, within a branching scheme. For instance, in [5], the authors propose a branching rule for the unsplittable multicommodity flow problem that does not destroy the structure of the pricing and the master problems in the column generation algorithm.

Algorithm 1: LAG Adjuster

```
Input: $G = (V, E)$, Capacities $C$, LAG loads $LL$,
       Link energy consumption
Output: Set of sub-components to turn-off $\bar{D}$
for router $n_i$ do
    for router $n_j$ neighbor of $n_i$ with $i < j$ do
        $L_j^i$ is the LAG set between $n_i$ and $n_j$
        Sorts links in $L_j^i$ in an ascending w.r.t. $\frac{\text{Link energy consumption}}{\text{Link capacity}}$
        $\bar{c} = 0$
        for $(u, v) \in L_j^i$ do
            if $\bar{c} = 0$ or $\bar{c} < LL(n_i, n_j)$ then
                $\bar{c} = \bar{c} + C_{(u,v)}$
            else
                $\bar{D} = \bar{D} \cup \{u, v\}$
for chipset $c \in D$ do
    if all ports of $c$ belongs to $\bar{D}$ then
        add $c$ to $\bar{D}$
for board $b \in D$ do
    if all chipsets of $b$ belongs to $\bar{D}$ then
        add $b$ to $\bar{D}$
```

VI. PERFORMANCE EVALUATION

This section presents numerical results for the evaluation of our energy-aware routing solution. Without loss of generality, we will consider a simpler setting where all devices (i.e., ports, chipsets, boards) have only two frequencies: ON and OFF. It corresponds to a network with legacy devices that do not support multiple frequencies. We also consider that traffic can be split arbitrarily for load balancing purposes.

A. *Benchmark and algorithms implementation*

For the evaluation, we compare two deployment scenarios where the controller can either decide 1) to tune both routing policies and frequencies of sub-components, or 2) to control routing only and let devices self-adapt their frequency. In the later case, a state-of-the-art LAG mechanism [17] locally adjusts the set of active links based on traffic. The greedy algorithm that is used to select active links is described in Alg. 1, denoted as *LAG adjuster*. Note that it takes as input the traffic load on every logical link, when routing has been decided by HyBenCg. It first activates the physical links in increasing order of energy efficiency for each logical link and, then, when the list of active ports is known for each node, it shuts down all possible chipsets and boards. Also, it keeps at least one active physical link in every LAG. In the following, we refer to HyBenCg with (resp. without) LAG adjuster as *HyBenCg-with* (resp. *HyBenCg-without*).

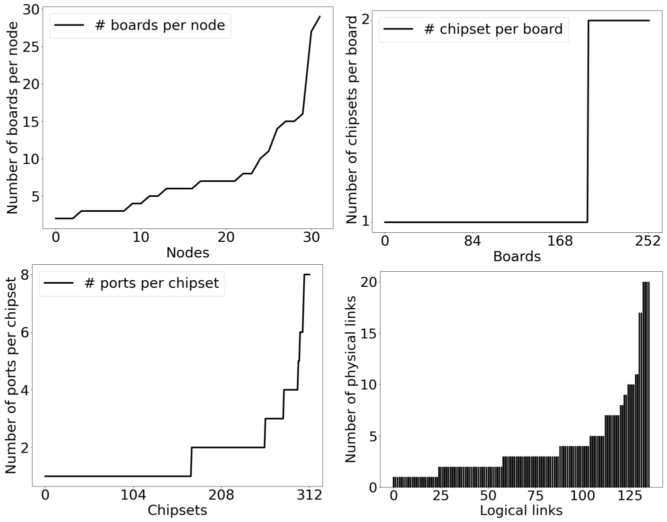

Fig. 6. The number of sub-components (ports per chipset, chipsets per board, boards per node) and the distribution of physical links in every LAG.

The algorithms have been implemented in C++ using Cplex 12.6 [16] as MILP-solver on a machine with Intel(R) Xeon(R) CPU E5-4627 v2 of 3.30GHz with 504GB RAM, running under Linux 64 bits. A maximum of 1 thread has been used.

B. Network instance

We consider a real backbone network from a regional operator in China. This network is composed of 32 core routers interconnected by 554 links. In total, routers are composed of 250 boards, 309 chipsets and 616 ports. Fig. 6 presents the distribution of physical links per logical IP link, i.e. per adjacent router pair. As we can observe, this backbone network has a lot of parallel links: 4 in average and up to 20. The energy consumption of ports is as follows: 3.5W for 100GE ports, 0.8W for 10GE ports and 0.4W for 1GE ports. Fig. 6 also shows the distribution of sub-components: ports per chipset, chipsets per board and boards per router. Note that multiple sub-components in the network cannot be controlled (they always need to be ON), as for the access ports at edge devices. Therefore, in the considered instance, the maximum energy saving that can be reached is 34.09% (when all controllable sub-components are turned-off and there is no traffic).

C. Traffic generation

In order to evaluate the performance of our solution, we generate a traffic matrix at random that consumes all the network capacity, i.e., a maximal traffic matrix that the network can handle. This implies using the highest frequency for every sub-component, i.e., 0% of energy saving. This way, we can easily evaluate the energy saving by scaling down all demands: 10%, ..., 70% of the maximal traffic. Alg. 2 presents the procedure we used to find the maximal traffic matrix.

D. Numerical results

Algorithm 2: Maximum traffic matrix generation

```
Input: $G = (V, E)$, LAG capacities $C$
Output: Maximum traffic matrix $d$
$K \leftarrow$ all combinations of node-pairs $(n_1, n_2)$ in $V$
Randomly shuffle the set $K$
for $(n_1, n_2) \in K$ do
    Compute a path $P$ in $G$ between $n_1$ and $n_2$
    $L \leftarrow$ the LAG in $P$ with a minimum capacity
    for $\bar{L} \in P$ do
        $C_{\bar{L}} = C_{\bar{L}} - C_L$
    Remove $L$ from $G$
    $d(n_1, n_2) = C_L$
```

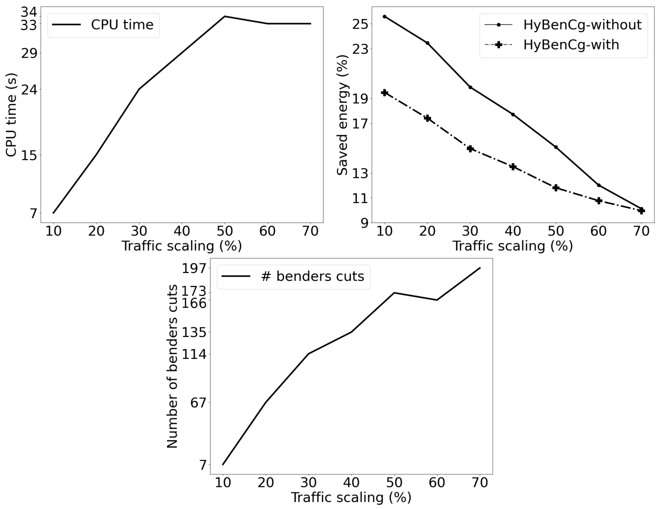

Fig. 7. CPU time, saved energy and number of generated cuts at different traffic scaling factors. Saved energy is plotted with and without LAG adjuster.

Fig. 7 shows three plots: the CPU time, the saved energy and the number of generated Benders cuts. Clearly, both CPU time is and the number of cuts increase respected to the traffic load. The higher the traffic load is, the higher the CPU time and the number of generated benders cuts are. The energy plot displays the energy saving gap, i.e., $\frac{\text{Energy saved}}{\text{Total energy consumption}} \times 100$, for HyBenCg-without and HyBenCg-with. We notice that the energy savings decrease when traffic grows. Indeed, when the traffic is higher, we can turn-off less sub-components. Conversely, HyBenCg-without gives much better energy savings, especially when traffic is small. Indeed, when traffic load is at 10% of the maximum traffic, HyBenCg-without gives 6% more saved energy. Fig. 8 compares the number of turned-off sub-components using HyBenCg-without and HyBenCg-with, when the traffic load is at 40% of the maximum traffic. It is easy to see that HyBenCg-with fails at selecting the right links in the LAGs in order to turn-off the boards. Indeed, HyBenCg-without turned-off 84.7% more boards.

VII. CONCLUSION

In this paper, we proposed an energy-aware routing solution that leverages knowledge about power management inside routers. It ensures reliability in the control plane by preventing any disruption to the IGP protocol. To make efficient routing

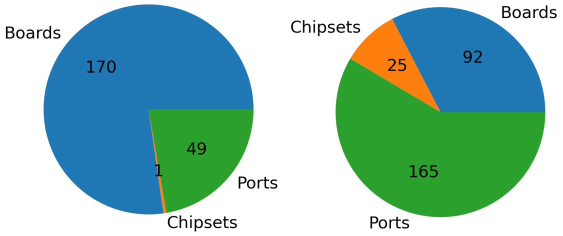

Fig. 8. Pie charts for the number of switched-off sub-components using HyBenCg-without (left) and HyBenCg-with (right). Note that ports and chipsets in turned-off boards are not counted. Same for ports in chipsets.

decisions at the controller, we proposed a path formulation of the associated optimization problem and showed that it is NP-Hard. We proposed a hybrid Benders decomposition and Column generation algorithm which decouples continuous and integer variables to efficiently to solve the problem on real instances. We evaluated two candidate deployment scenarios for the solutions: one where the controller decides both routing policies and frequencies of sub-components, and one where it only controls routing and lets devices adapt. In the later case, devices self-adapt their frequency. We showed on a real backbone, from a regional provider in China, that the joint optimization of power management and routing can further save energy by 6% at the cost of a higher complexity.

To keep complexity low and maintain loose interaction between routers and the controller, our work showed that there is room for interesting research on the development of distributed mechanisms to coordinate power management mechanisms among neighbouring routers. Indeed, LAG adjusters could make more informed decisions so that chipsets / boards of routers could also switch to better energy saving modes.

References

[1] Between 10-20% of electricity consumption from ICT sector in 2030? Enerdata report, 2018.

[2] Abdullah, Z.N., Ahmad, I., Hussain, I.: Segment routing in software defined networks: A survey. IEEE Communications Surveys & Tutorials **21**(1), 464–486 (2018)

[3] Addis, B., Capone, A., Carello, G., Gianoli, L.G., Sanso, B.: A robust optimization approach for energy-aware routing in mpls networks. In: Proc. IEEE ICNC (2013)

[4] Amaldi, E., Capone, A., Gianoli, L.G.: Energy-aware ip traffic engineering with shortest path routing. Computer Networks **57**(6), 1503–1517 (2013)

[5] Barnhart, C., Hane, C.A., Vance, P.H.: Using branch-and-price-and-cut to solve origin-destination integer multicommodity flow problems. Operations Research **48**(2), 318–326 (2000)

[6] Belieres, S., Hewitt, M., Jozefowiez, N., Semet, F., Van Woensel, T.: A benders decomposition-based approach for logistics service network design. European Journal of Operational Research **286**(2), 523–537 (2020)

[7] Benders, J.F.: Partitioning procedures for solving mixed-variables programming problems. Numerische mathematik **4**(1), 238–252 (1962)

[8] Bianzino, A.P., Chaudet, C., Larroca, F., Rossi, D., Rougier, J.L.: Energy-aware routing: A reality check. In: Proc. IEEE Globecom Workshops (2010)

[9] Bianzino, A.P., Chiaraviglio, L., Mellia, M.: Distributed algorithms for green IP networks. In: Proc. IEEE INFOCOM Workshops (2012)

[10] Bianzino, A.P., Chiaraviglio, L., Mellia, M., Rougier, J.L.: Grida: Green distributed algorithm for energy-efficient ip backbone networks. Computer Networks **56**(14), 3219–3232 (2012)

[11] Capone, A., Cascone, C., Gianoli, L.G., Sanso, B.: OSPF optimization via dynamic network management for green IP networks. IEEE (2013)

[12] Christensen, K., Reviriego, P., Nordman, B., Bennett, M., Mostowfi, M., Maestro, J.A.: IEEE 802.3 az: the road to energy efficient ethernet. IEEE Communications Magazine **48**(11), 50–56 (2010)

[13] Fernandez-Fernandez, A., Cervello-Pastor, C., Ochoa-Aday, L.: Achieving energy efficiency: An energy-aware approach in sdn. In: Proc. IEEE GLOBECOM (2016)

[14] Galán-Jiménez, J., Polverini, M., Lavacca, F.G., Herrera, J.L., Berrocal, J.: Joint energy efficiency and load balancing optimization in hybrid IP/SDN networks. Annals of Telecommunications pp. 1–19 (2022)

[15] Garey, M.R., Johnson, D.S.: Computers and intractability. A Guide to the to the theory of NP completeness (1979)

[16] IBM: ILOG CPLEX Solver, https://www.ibm.com/analytics/cplex-optimizer

[17] Imaizumi, H., Nagata, T., Kunito, G., Yamazaki, K., Morikawa, H.: Power saving mechanism based on simple moving average for 802.3 ad link aggregation. In: Proc. IEEE Globecom Workshops (2009)

[18] Ishiguro, K., Manral, V., Davey, A., Lindem, A.: Traffic engineering extensions to OSPFv3. Tech. rep. (2008)

[19] Lübbecke, M.E.: Column generation. Wiley encyclopedia of operations research and management science. Wiley, New York pp. 1–14 (2010)

[20] Maaloul, R., Chaari, L., Cousin, B.: Energy saving in carrier-grade networks: A survey. Computer Standards & Interfaces **55**, 8–26 (2018)

[21] Noguera, J., Kennedy, I.O.: Power reduction in network equipment through adaptive partial reconfiguration. In: International Conference on Field Programmable Logic and Applications. IEEE (2007)

[22] Reviriego, P., Sivaraman, V., Zhao, Z., Maestro, J.A., Vishwanath, A., Sánchez-Macian, A., Russell, C.: An energy consumption model for energy efficient ethernet switches. In: Proc. HPCS (2012)

[23] Reviriego, P., Hernández, J.A., Larrabeiti, D., Maestro, J.A.: Performance evaluation of energy efficient ethernet. IEEE Communications Letters **13**(9), 697–699 (2009)

[24] Wierman, A., Andrew, L.L., Tang, A.: Power-aware speed scaling in processor sharing systems. In: Proc. IEEE INFOCOM (2009)

[25] Zhang, M., Yi, C., Liu, B., Zhang, B.: GreenTE: Power-aware traffic engineering. In: Proc. IEEE ICNP (2010)