Energy-Efficient Service Function Chain Provisioning

Nicolas Huin, Andrea Tomassilli, Frédéric Giroire, and Brigitte Jaumard

RESEARCH REPORT

N° 8980

November 2016

Project-Team Coati

ISSN 0249-6399   ISRN INRIA/RR--8980--FR+ENG

Energy-Efficient Service Function Chain Provisioning

Nicolas Huin*, Andrea Tomassilli*, Frédéric Giroire*, and Brigitte Jaumard†

Project-Team Coati

Research Report n° 8980 — November 2016 — 15 pages

Abstract: Network Function Virtualization (NFV) is a promising network architecture concept to reduce operational costs. In legacy networks, network functions, such as firewall or TCP optimization, are performed by specific hardware. In networks enabling NFV coupled with the Software Defined Network (SDN) paradigm, network functions can be implemented dynamically on generic hardware. This is of primary interest to implement energy efficient solutions, which imply to adapt dynamically the resource usage to the demands. In this paper, we *study how to use NFV coupled with SDN to improve the energy efficiency of networks.* We consider a setting in which a flow has to go through a Service Function Chain, that is several network functions in a specific order. We propose a decomposition model that relies on lightpath configuration to solve the problem. We show that virtualization allows to obtain between 30% to 55 % of energy savings for networks of different sizes.

Key-words: Network Function Virtualization, Service Function Chains, Software Defined Networks, Energy Efficiency, Optimization, Column Generation.

[*] N. Huin, A. Tomassilli, and F. Giroire have been partially supported by ANR program ANR-11-LABX-0031-01 and Mitacs.

[†] B. Jaumard was supported by a Concordia University Research Chair (Tier I) and by an NSERC (Natural Sciences and Engineering Research Council of Canada) grant.

RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Affectation économe en énergie de chaînes de services

Résumé : La virtualisation des fonctions réseaux (NFV) est un concept d'architecture réseaux prometteur pour réduire les coûts opérationnels. Dans les réseaux traditionnels, les fonctions réseaux, comme un firewall ou un optimisateur TCP, sont effectués sur des équipements spécifiques. Dans les réseaux permettant la NFV ainsi que les réseaux logiciels (Software Defined Networks), les fonctions réseaux peuvent être implémentée dynamiquement sur des équipements génériques. Cela est très prometteur pour mettre en pratique des solutions efficaces en énergie qui implique une adaptation dynamique des ressources aux demandes. Dans ce paper, nous étudions comment utiliser les technologies NFV et SDN pour améliorer l'éfficacité énergétique des réseaux. Nous considérons un cadre dans lequel un flôt doit passer au travers d'une chaîne de fonctions de service, c'est-à-dire par une séquence de fonctions réseaux dans un ordre donné. Nous proposons un modèle de décomposition pour résoudre le problème. Nous montrons que la virtualisation permet d'obtenir entre 30 et 55% de gains énergétiques pour des réseaux de différentes tailles.

Mots-clés : Virtualisation des fonctions réseaux, Chaînes de fonctions de service, Réseaux logiciels, Efficacité énergétique, Optimisation, Génération de Colonnes

1 Introduction

Nowadays, network operators are trying to reduce their operational costs by smartly upgrading their networks. To this end, two recent paradigms are particularly promising: Software Defined Networks (SDNs) and Network Function Virtualization (NFV). SDN decouples the data plane from the control plane, and puts the intelligence into a centralized controller. This allows the deployment of advanced dynamic protocols and, thus, a finer network optimization based on metrology data, regularly collected by network nodes and sent to the controller.

For security and efficiency reasons, network flows have to go through a number of network functions. Examples of network functions are firewall, load balancers, Video Optimizer Controller,... Network functions are traditionnally implemented on specific harware. This leads to inefficiency as the hardware is provisionned for peak traffic. The principle of the NFV paradigm is to virtualize the functions to obtain Virtualized Network Functions (VNFs). This allows to implement these functions on generic hardware. Coupled with SDN, VNFs can be installed dynamically when needed, at the best position in the network, at the best time, and for only the adequate traffic. This allows a reduction of hardware cost (Capex) and of operational cost (Opex).

One of the important operational cost is the energy used by the nework. With the sharp increase of Internet traffic, networks have to be more energy efficient [1]. To this end, Energy-Aware Routing solutions have been proposed, see e.g. [2, 3]. The principle is to aggregate traffic on a number of network equipements as small as possible, in order to turn off the unused ones. SDN and NFV will allow to put the proposal into practice by enabling dynamic routing and network function provisionning.

In this paper, we *explore the potential of network virtualization to reduce network energy consumption.* We consider a setting in which a network flow has to go through a *Service Function Chain (SFC)*, that is an ordered sequence of VNFs. We study the problem of minimizing network energy consumption, while satisfying at the same time the link and node capacity constraints and the SFCs constraints. We call this problem the Capacitated Energy Efficient Service Function Chain Provisioning Problem (EE-SFCP),

Several optimization models were proposed in the litterature to solve the problem of SFC-provisioning. [4, 5, 6, 7, 8]. However, these works do not consider the minimization of the network energy consumption under a dynamic traffic. In [9], we introduce a new Integer Linear Program to model the problem of Energy-Efficient SFC provisioning and a heuristic algorithm to solve large instances. However, no garantees are provided on the heuristic algorithm efficiency. In this paper, we remedy this limitation by comparing it with a decomposition model. Our contributions are the following:

- We propose a column generation model to solve the problem of EE-SFC-provisioning on large instances.
- The energy-efficiency problem is a difficult optimization problem. As a matter of fact, it contains a sharp On-Off phenomena, as a network device consumes a large portion of its energy as soon as it is used, even if very lightly used. We thus discuss several variants of the model to reduce the integrality gap.
- We use the decomposition model to carry out extensive simulations on network of different sizes. We show that between 30% to 55% of energy can be saved while respecting the constraints of the service chains.
- We compare the results obtained from the decomposition model to the one of GreenChains. The decomposition model succeeds in finding better solutions than the heuristic

algorithm. However, the results are not very far. This validates the heuristic algorithm, which finds solutions not far from optimal for an important number of cases.

2 Statement of the Problem: SFC and VNF Placement

2.1 Notations

We assume the network to be represented by a directed graph $G = (V, L)$, where V is the set of nodes (indexed by v) and L is the set of links (indexed by ℓ). Each node $v \in V$ has a a set of compute, storage and network resources denoted by C_v to host network functions. Within the cope of this study, we will assume that the resources are described by a given number of cores.

Traffic is described by a set of demand requests D, in which each demand d is defined by a 3-tuple (v_s, v_d, c), where v_s is the source of the demand, v_d its destination, and c the requested service chain. Let D^c_{sd} be the bandwidth requirement of demand d. Indeed, each demand d is associated with a given application, which is required to pass through a given service function chain (SFC), i.e., a sequence of virtual functions. Let D^c_{sd} be the bandwidth requirement of demand d.

Let F be the overall set of virtual functions arising in the service chains, indexed by f, and C be the set of service chains, indexed by c. Each service chain c corresponds to a sequence of n_c functions $f^c_1, \dots, f^c_i, \dots, f^c_{n_c}$, where f^c_i denotes the ith function of chain c. Note that some functions may appear more than once in a given chain and that for various reasons (e.g., resiliency, privacy), two different functions f and f' may not be allowed to be installed in the same node. Each virtual function f has its one resource requirement and we denote by Δ_f the number (fraction) of cores required by function f per bandwidth unit.

The *Energy Efficient Service Function Chain Provisioning (*EE-SFCP*)* consists in jointly provisioning a set D of demand requests coupled with service function chains C and placing virtual functions arising in the chains, in order to minimize the network energy consumption, subject to link and node capacities.

2.2 Power Model

Campaigns of measures of power consumption (see, e.g., [10]) show that a network device consumes a large amount of its power as soon as it is switched on and that the energy consumption does not depend much on the load. Following this observation, on/off power models have been proposed and studied. Later, researchers and hardware constructors have proposed more energy proportional hardware [11]. To cover the different models for different hardware, we use a hybrid power model in which the power of an active link ℓ is expressed as follows:

$$P_\ell = P^{\text{ON}}_\ell + \frac{\text{BW}_\ell}{C_\ell} P^{\text{MAX}}_\ell. \tag{1}$$

where P^{ON}_ℓ represents the energy used when the link ℓ is switched on, BW_ℓ the bandwidth that is carried on ℓ, and P^{MAX}_ℓ the energy consumed by ℓ when it is fully capacitated, i.e., when the amount of carried bandwidth equals the transport capacity (C^{LINK}_ℓ) of link ℓ.
We assume that links can be put into sleep mode, by putting to sleep both endpoint interfaces. Two links in opposite direction ($\ell = (v, v')$ and $\ell' = (v', v)$) between a pair of nodes will be assumed to be in the same state (active or in sleep mode), as the send and receive elements of a unidirectional fiber are usually controlled by the same interface. Routers cannot be put into sleep mode, as there are the sources or destinations of network traffic. However, cores may be

put into sleep mode and the power used by nodes is given by

$$P_v = P_v^{\text{UNIT}} \times \#\text{cores}, \tag{2}$$

where P_v^{UNIT} represents the energy consumption of a single core.

2.3 Layered Graph

Following a similar idea as in [12, 13], we use a layered graph G^{L} that is defined as follows. The initial network graph G is transformed into a *layered graph* G^{L} by adding $\max\limits_{c \in C} n_c$ layers to the graph (counting G as the base layer) and each layer is an exact copy of the original graph. For every node $v \in V$, let v^i denote the corresponding node in the ith layer ($i = 0, 1, \ldots, n^c$). Every $(i-1, i)$ layer pair is connected vertically by links from v^{i-1} to v^i. We denote by $L(G^{\text{L}})$ and by $V(G^{\text{L}})$ the set of links and nodes of graph G^{L}, respectively.

Provisioning of a chain and node placement of its functions amounts to find a path from node v_s on the first layer (graph G) to node v_d on the n^cth layer. Placement of a function on a node is given by the endpoints of the link used to switch between layers.

We next set a column generation model for EE-SFCP, which makes use of the layered graph G^{L}.

3 Decomposition Models

We first present here a model using Column Generation, *CG-simple*. We then introduce two variants of the models, *CG-cuts*, and *CG-cut+*. Indeed, problems dealing with energy-efficiency frequently lead to large integrality gap and bad precision. This is due to the On-Off phenomena of power models, which translates into large steps of the objective function. We thus try to improve the precision of the model by introducing different sets of constraints. We discuss the precision of the models in Section 5.2.

3.1 Column Generation Formulation

We propose a column generation formulation that relies on the concept of chaining & function placement configurations: each configuration γ is associated with a 3-uplet (v_s, v_d, c) and defines: *(i)* a potential route for demand D_{sd}^c associated with node pair (v_s, v_d) for chain c and, *(ii)* node placement of the functions of chain c along the potential route. Route is described by parameters δ_ℓ^γ, equal to 1 if link ℓ belongs to the path, 0 otherwise. Node placement is given by $a_{vf_i}^\gamma$, equal to 1 if the ith function f_i of c is located at node v, 0 otherwise. We denote by Γ the overall set of configurations.

We now define the set of variables.

- $x_\ell \in \{0, 1\}$. $x_\ell = 1$ if link ℓ is on (active), 0 otherwise. Note that links are powered off by pair, i.e., $x_{\ell=(v,v')} = x_{\ell'=(v',v)}$
- $y_d^\gamma \in \{0, 1\}$. $y_d^\gamma = 1$ if demand d is routed using configuration γ, 0 otherwise.
- $\text{REQ}_v \in \mathbb{N}$. $\text{REQ}_v = \#$ required cores in node v.

The objective, i.e., the minimization of the energy, can be written

$$\min \quad \underbrace{\sum_{\ell \in L} P_\ell^{\text{ON}} x_\ell}_{\text{link switch on energy}} \quad + \underbrace{\sum_{\ell \in L} \sum_{\gamma \in \Gamma} \delta_\ell^\gamma \left(\sum_{d=(v_s,v_d,c) \in D} \frac{D_{sd}^c}{C_\ell^{\text{LINK}}} P_\ell^{\max} \right) y_d^\gamma +}_{\text{link bandwidth energy}} \underbrace{\sum_{v \in V^{\text{NFV}}} P_v \, \text{REQ}_v}_{\text{node resource energy}} \; . \quad (3)$$

The constraint set decomposes into several sets of contraints, which are next described. One path per demand

$$\sum_{\gamma \in \Gamma_d} y_d^\gamma = 1 \qquad d = (v_s, v_d, c) \in D. \quad (4)$$

Link capacity

$$\sum_{d=(v_s,v_d,c) \in D} \; \sum_{\gamma \in \Gamma_d} D_{sd}^c \, \delta_\ell^\gamma \, y_d^\gamma \leq x_\ell \, C_\ell^{\text{LINK}} \qquad \ell \in L. \quad (5)$$

Node capacity

$$\sum_{d=(v_s,v_d,c) \in D} \; \sum_{\gamma \in \Gamma_d} D_{sd}^c \left(\sum_{i=1}^{n_c} \Delta_{f_i} a_{v f_i}^\gamma \right) y_d^\gamma \leq \text{REQ}_v \leq C_v^{\text{NODE}} \qquad v \in V^{\text{NFV}}. \quad (6)$$

The *CG-cuts* model. To reduce the integrality gap, we add cuts, which are next described. Inequalities (7) states that, for each node, at least one incident link should always be on. Moreover, at least $n - 1$ links should be on to have a connected network (or different if not all-to-all). Inequality (8) enforce that last condition.

$$\sum_{\ell \in \omega^+(v)} x_\ell \geq 1 \qquad v \in V \quad (7)$$

$$\sum_{\ell \in L} x_\ell \geq n - 1. \quad (8)$$

The *CG-cut+* model. For the second variant, we add the following set of constraints:

$$x_\ell \geq \sum_{\gamma \in \Gamma_d} \delta_\ell^\gamma \, y_d^\gamma \qquad \ell \in L, \gamma \in \Gamma_d \quad (9)$$

Following Equation (4), we have that $\sum\limits_{\gamma \in \Gamma_d} \delta_\ell^\gamma \, y_d^\gamma \leq 1$. This allows us to circumvent the use of a big M formulation at the expense of a large number of constraints.

3.2 Solution Scheme

There is a configuration generator, i.e., pricing problem, for each $d = (v_v, v_d, c) \in D$. Two sets of decision variables are required. First set is made of variables φ_ℓ^i such that $\varphi_\ell^i = 1$ if the provisioning of demand d uses link ℓ in layer i of the layered graph G^{L}, 0 otherwise. Second set contains variables a_v^i such that $a_v^i = 1$ if the ith function (f_i) of chain c for demand $d = (v_v, v_d, c)$ is placed on NFV node v, 0 otherwise.

$$\min \quad \sum_{\ell \in L} P^{\max} \frac{D_{sd}^c}{C^{\text{LINK}}{}_\ell} \sum_{i=0}^{n_c} \varphi_\ell^i \; - \; u_{sd}^{(4)} \; + \sum_{\ell \in L} u_\ell^{(5)} D_{sd}^c \sum_{i=0}^{n_c} \varphi_\ell^i \; + \; D_{sd}^c \sum_{v \in V} u_v^{(6)} \sum_{i=0}^{n_c} \Delta_{f_i} a_v^i. \quad (10)$$

Path computation (flow conservation constraints):

$$\sum_{\ell \in \omega^+(v)} \varphi_\ell^i - \sum_{\ell \in \omega^-(v)} \varphi_\ell^i + a_v^i - a_v^{i-1} = 0 \qquad v \in V, 0 < i < n^c \tag{11}$$

$$\sum_{\ell \in \omega^+(v)} \varphi_\ell^0 - \sum_{\ell \in \omega^-(v)} \varphi_\ell^0 + a_v^0 = \begin{cases} 1 \text{ if } v = v_s \\ 0 \text{ else} \end{cases} \qquad v \in V \tag{12}$$

$$\sum_{\ell \in \omega^+(v)} \varphi_\ell^{n^c} - \sum_{\ell \in \omega^-(v)} \varphi_\ell^{n^c} - a_v^{n^c} = \begin{cases} -1 \text{ if } v = v_d \\ 0 \text{ else} \end{cases} \qquad v \in V. \tag{13}$$

Link capacity

$$D_{sd}^c \sum_{i=0}^{n_c} \varphi_\ell^i \leq C_\ell^{\text{LINK}} \quad \ell \in L. \tag{14}$$

Node capacity

$$D_{sd}^c \sum_{i=0}^{n_c} \Delta_{f_i} a_v^i \leq C_v^{\text{NODE}} \qquad v \in V^{\text{NFV}}. \tag{15}$$

4 GreenChains Heuristic

In [13], we proposed a heuristic algorithm called GreenChains to solve EE-SFCP. We briefly recall it below. Heuristic GreenChains solves EE-SFCP in three steps.

- First, the *routing problem* is to compute a path for each demand, respecting the link capacity constraints.
- Second, the goal of the *service chain placement problem* is to find a placement of the NVF respecting the capacities of the nodes and the order defined by the service chains.
- Last, the *energy saving problem* tries to put into sleep mode as many links and cores as possible to decrease the energy consumption of the network.

Routing Step. Demands are considered one at a time. For each demand, we compute a weighted shortest path on a residual graph. The weight of a link is equal to the inverse of its spare capacity. This favours the links with lower loads. The idea is to spread as much as possible the demands to find a feasible routing. To compute the residual graph, when the path is computed, we reduce the link capacities by the value of the demand. Furthermore, when considering a new demand to be routed, we remove links with a residual capacity smaller than the demand.

Service Chain Placement Step. When the paths for all demands are set, we place the service chains. We consider the different service chains separately. For a given chain, we consider the paths of all the demands with this chain. We then sort the nodes of the networks in the decreasing order of the number of such paths they are involved in. We then try to place all the functions of the chains on the nodes following this order. If it is possible in a node u, we place the functions. We then assign some of the paths to the cores of node u, till its capacity is full. We then actualize the ordered list of nodes considering only the paths not yet assigned.

Energy Saving Step. We attempt to greedily put links into sleep mode. It starts with the graph $G = (V, L)$. It first launches the routing module and then the service chain placement module. If they both succeed, it creates a list U of all links according to their usage (volume of traffic). It then chooses the less loaded link $\ell_{\min}$ as a candidate to be put in sleep mode. It now

Service Chain	Chained VNFs	rate	% traffic
Web Service	NAT-FW-TM-WOC-IDPS	100 kbps	18.2%
VoIP	NAT-FW-TM-FW-NAT	64 kbps	11.8%
Video Streaming	NAT-FW-TM-VOC-IDPS	4 Mbps	69.9%
Online Gaming	NAT-FW-VOC-WOC-IDPS	50 kbps	0.1%

Table 1: Service Chain Requirements [4]

NAT: Network Address Translator, FW: Firewall, TM: Traffic Monitor, WOC: WAN Optimization Controller, IDPS: Intrusion Detection Prevention System, VOC: Video Optimization Controller

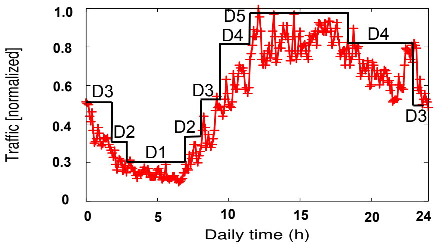


Figure 1: Normalized daily variation of traffic of a France Telecom network link and multi-period approximation.

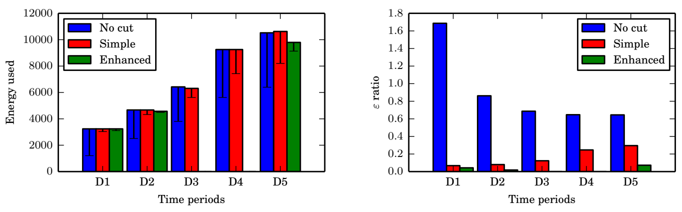
considers the graph $G' = (V, L \setminus \{\ell_{\min}\})$. It launches again the routing and placement modules. If they succeed, $\ell_{\min}$ is definitely put in sleep mode. The list U is updated with the new routing, as well as the less loaded link. If at least one of the two modules fails, GreenChains considers that $\ell_{\min}$ cannot be into sleep mode and the link is definitely kept active in the final solution. The second element of U is then considered. The algorithm goes on till all links have been tried and set either as definitely in sleep mode or definitely active.

5 Numerical Experiments

In this section, we investigate the energy savings obtained by the Column Generation model. We compare the results with ones of the GreenChains heuristic algorithm, proposed in [9]. We first present the data sets we use for the experiments. We then take a look at the precision of the solutions obtained by the Column Generation model and GreenChains. We investigate different improvements of the model presented in Section 3. We then present the energy savings obtained for network topologies of different sizes. Last, we discuss the impact of the solutions on link usage and path lengths.

5.1 Data sets

In networks, each type of flows has to go through a different chain of network services. In our experiments, we consider the four of the most frequent types of flows as presented in Table 1: Video Streaming, Web Service, Voice-over-IP (VoIP), and Online Gaming. The traffic percentage are from [14]. For each one, we give the ordered set of functions required and the bandwidth used. In total, 6 different functions are used, and each function requires a different amount of cores to be executed.



(a) Energy used

(b) $\varepsilon = (\tilde{z}_{\text{ILP}} - z^{\star}_{\text{LP}})/z^{\star}_{\text{LP}}$

Figure 2: Comparison of the performance of different Column Generation Models for *pdh*.

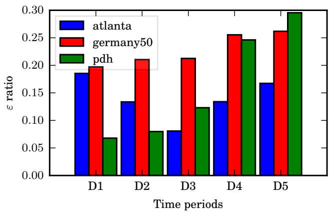


Figure 3: ε ratio for the *pdh*, *atlanta*, and *germany50* networks for varying amount of traffic.

We tested the CG model and GreenChains on three topologies of different sizes from SNDlib [15]: *pdh* (11 nodes and 64 directed links), *alanta* (15 nodes and 44 directed links), and *germany50* (50 nodes and 176 directed links).

For each network, we generate a set of demands from the traffic matrices provided in SNDLib: we divide each aggregate flow from a source to a destination into four demands corresponding to the four different types of traffic. The original load of the flow is conserved and each subflow load is given by the distribution of the last column of Table 1. For example, a flow with a charge of 1 is splitted into a Web Service, a VoIP, a Video Streaming and an Online Gaming sub-flows with a respective charge of 0.182, 0.118, 0.699 and 0.001.

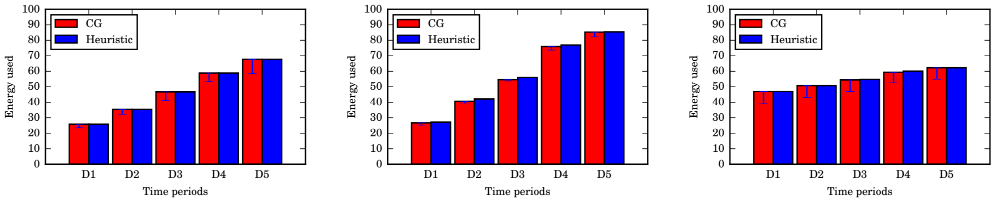
We tested the solution on a daily traffic to see how much energy can be saved during the day or at night. The variations of traffic come from a trace of a typical France Telecom link shown in Figure 1. Previous work [16] shows that most of the energy savings can be obtained by using a small number of configurations during the day. In our case, we considered 5 different levels of traffic called D1 to D5. D1 represents the period with the lowest amount of traffic and D5 the one with the highest.

5.2 Quality of the Column Generation models

In Section 3, we propose three different models using Column Generation, *CG-simple*, *CG-cuts*, and *CG-cut+*. We now study the impact of the different sets of constraints on the final objective value of the Column Generation and on the ratio between the Integer Reduced Master Problem (RMP) and its Fractional counterpart, $\varepsilon = (\tilde{z}_{\text{ILP}} - z^{\star}_{\text{LP}})/z^{\star}_{\text{LP}}$. In Figure 2, we compare the solutions founded by the three CG models for the *pdh* network and for the 5 different levels of traffic. Energy used is given in Figure 2a (Error bars represents the energy used found by the RMP) and ε ratio in Figure 2b. The first observation is that the three models obtain similar energy savings: almost always equal for *CG-simple* and *CG-cuts*, and a little bit better (put percent) for *CG-cut+*. However, the ε ratio is dramatically different. Adding the cuts improves greatly the ratio: ε for *CG-simple* varies between 169% for the D1 period and 64% for the D5 period, when, for *CG-cuts*, ε is between 8 to 30%. The ratio is further improved with *CG-cut+*: between 2 and 7%. As the energy savings are similar for the three models, this shows that the *three CG models provide solutions that are few percent from optimal, but that this is easier for Cplex to prove it for the* CG-cuts *and* CG-cut+ *models.*

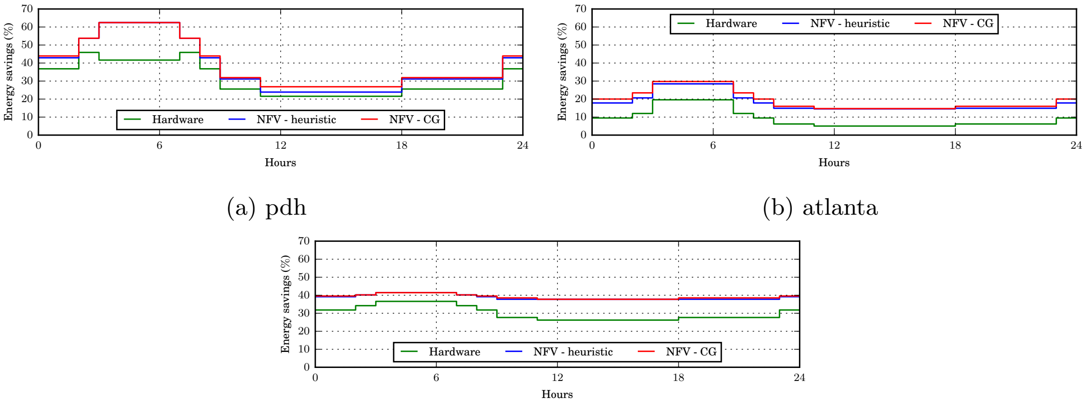
For the following results, we *focus on the* CG-cuts *model*, as its ratio is a lot better than the one of *CG-simple*, and, as *CG-cut+* requires too much memory to solve large networks. We are not able to solve the problem for the *pdh* network for the D3 and D4 period.

We provide in Figure 3 the values of the ε ratio for the three studied topologies from SNDLib.



(a) pdh (b) atlanta (c) germany50

Figure 4: Comparison of the energy used and saved between GreenChains and the Column Generation model for *pdh* network.



(a) pdh (b) atlanta

(c) germany50

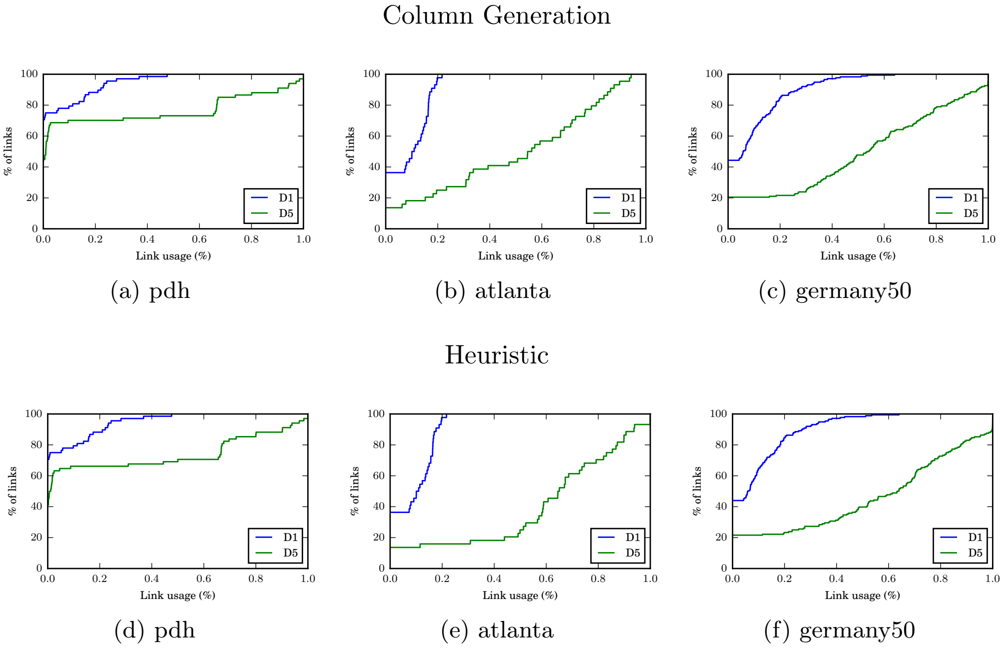
Figure 5: Energy savings during the day.

On *pdh*, the ratio increases from 7% at the D1 period to 17% at the D5 period. For *atlanta*, the ratio decreases from 19% at D1 to 8% at D3, before increasing back to 27% at D5. For *germany50*, the ratio steadily increases from 20 to 26%.

Energy Savings. We compare the savings obtained by our solutions with the *legacy scenario*, which serves as a baseline, and a *hardware scenario*. The *legacy scenario* considers a legacy network which does not implement SDN nor NFV. The routing is thus static and the network functions are not virtualized. They are executed by specific hardware, which are installed at given positions in the network. The demands use the shortest paths between the sources, hardwares implementing the network functions of their services chains, and the destination. The *hardware scenario* corresponds to the one of an SDN (non virtualized) networks in which an operator tries to reduce its energy consumption by adapting the routing to the demands. In this scenario, the network functions are carried out by some specific hardware placed at given positions in the network.

We provide in Figure 4 the energy used for the 5 levels of demands D1, D2, D3, D4 and D5 for the SNDLib networks. The values are normalized: 100 corresponds to the legacy scenario. We also present in Figure 5 the corresponding energy savings during the day for the *germany50* network. We see that we obtained important savings using virtualization: between 25 and 62 % for *pdh*, 15 and 30 % for *atlanta*, and 38 and 42 % for *germany50*.

Heuristic vs the Column Generation model. We compare in Figure 5 the Column Gen-



(a) pdh (b) atlanta (c) germany50

(d) pdh (e) atlanta (f) germany50

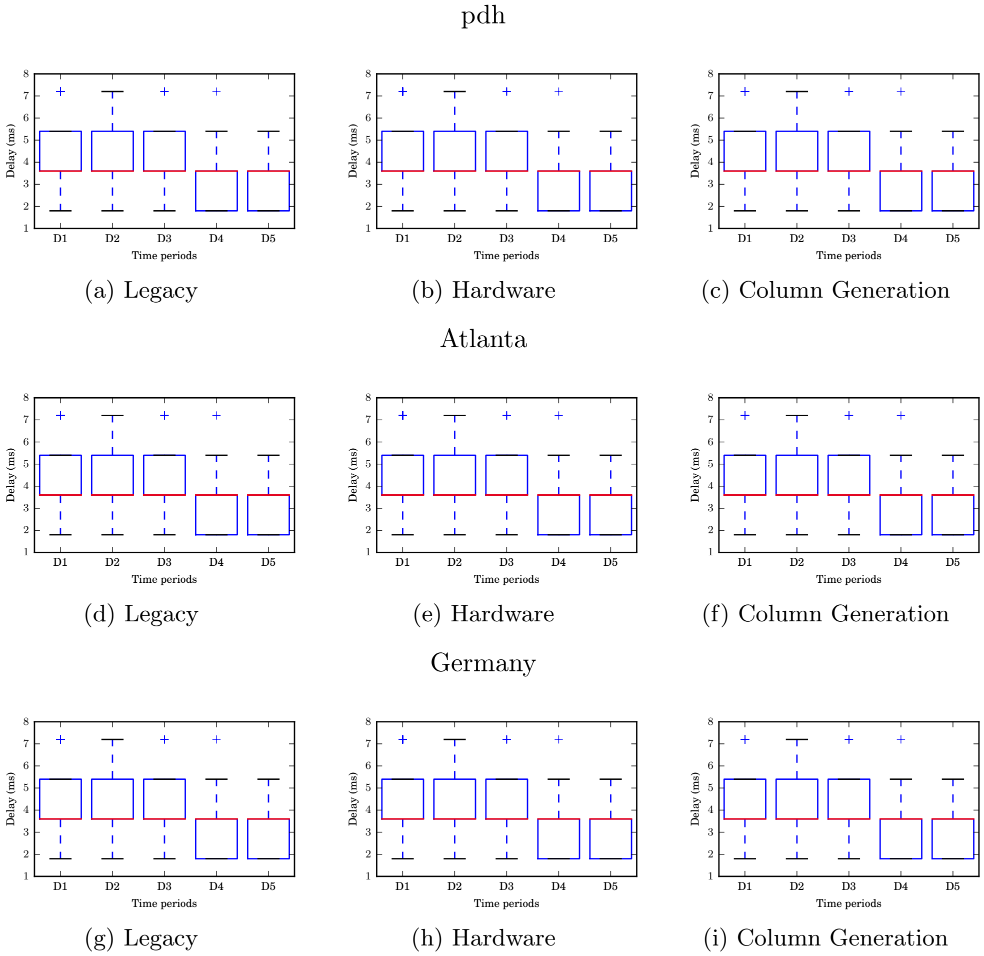
Figure 6: Cumulative distribution function of the links load for the lowest and highest traffic period.

eration model and the heuristic. Error bars on the CG solutions represent the lower bounds, that is the relaxation of the RMP ($z^{\star}_{\mathrm{LP}}$). For all three topologies and for all time periods, the difference between the heuristic and the Column Generation model is small. For the lowest traffic periods (D1 and D2), both methods provide similar solutions. The CG model provides slightly better solutions when the traffic is higher, with a difference of 4, 7 and 5% for *pdh*, *atlanta* and *germany50* respectively in the D5 period. The important thing to note here is that, even though the Column Generation does not provide a much better solution than the heuristic, it does show (by providing the ε ratio) that the heuristic gives good results, regardless of the traffic period.

5.3 Load of links

To reduce the amount of energy used by the network, we reroute some of the flows to be able to put links into sleep. This means that the remaining active links are more loaded. In Figure 6, we look at the link load for the highest and lowest traffic periods. First, we see that for *atlanta* and *germany50*, with no surprise, the number of links with no traffic is higher when the traffic is low, with around 40% of the links that forward no traffic at all. When the network is at its highest utilization it drops to 20% or less. The *pdh* network, due to its higher link density, can have more links put into sleep mode. Indeed, between 45% and 70% of the links have no traffic. Moreover, at the lowest traffic period no links are used at 100%. They are at most used up to 50%, 22% and 63% of their capacity, for *pdh*, *atlanta* and *germany50* respectively. At rush hour, *germany50* and *pdh* have about 5 and 10% at full capacity, while *atlanta* links are at most used at 92%.

5.4 Delay



(a) Legacy (b) Hardware (c) Column Generation

(d) Legacy (e) Hardware (f) Column Generation

(g) Legacy (h) Hardware (i) Column Generation

Figure 7: Distribution of delay for the 5 traffic periods on the pdh, atlanta and germany50. The box represents the 1st and 3rd quartile. The median is represented by the red line and the whiskers represents the 1st and 9th decile.

Again, when some links are put into sleep mode, some of the paths between sources and destinations are becoming longer. We investigate the impact of these changes on the delays in the network. To compute the delays, we assume that they are proportional to the distances traveled along the paths [17]. SNDLib provides the geographical coordinates of the nodes of *germany50*. We thus derived the link lengths based on these coordinates. For the other two networks, no data is available. We thus use the average delay of 1.8 ms founded in *germany* as the link length.

In Figure 7, we look at the delay of the request for each period of time. The median remains the same on *pdh* and *atlanta* all day long. On *germany50*, it slightly decreases for higher traffic periods, as more links are active. At most, requests experience delay of about 7, 14 and 25 on *pdh*, *atlanta* and *germany50* respectively. An important observation is that the maximum delay of every path stays below 50 ms, which is a classic value for the maximum delay in Service Level Agreements [18].

6 Conclusion

In this work, we investigate the potential of network virtualization to reduce the energy consumption of networks. We propose a Column Generation model to solve the problem of minimizing network energy consumption while satisfying the constraints of service chains of virtualized network functions. We obtain between % of energy savings for network topologies of different sizes.

References

[1] D. Matsubara, T. Egawa, N. Nishinaga, V. P. Kafle, M.-K. Shin, and A. Galis, "Toward future networks: a viewpoint from itu-t," *IEEE Communications Magazine*, vol. 51, no. 3, pp. 112–118, 2013.

[2] L. Chiaraviglio, M. Mellia, and F. Neri, "Minimizing isp network energy cost: Formulation and solutions," in *IEEE/ACM Trans. Netw.'11.*

[3] R. Modrzejewski, L. Chiaraviglio, I. Tahiri, F. Giroire, E. Le Rouzic, E. Bonetto, F. Musumeci, R. Gonzalez, and C. Guerrero, "Energy Efficient Content Distribution in an ISP Network," in *IEEE Global Communications Conference (GLOBECOM 2013)*, Atlanta, United States, Dec. 2013.

[4] M. Savi, M. Tornatore, and G. Verticale, "Impact of processing costs on service chain placement in network functions virtualization," in *IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN)*, Nov. 2015, pp. 191–197.

[5] A. Gupta, M. F. Habib, P. Chowdhury, M. Tornatore, and B. Mukherjee, "Joint virtual network function placement and routing of traffic in operator networks," *UC Davis, Davis, CA, USA, Tech. Rep*, 2015.

[6] B. Martini, F. Paganelli, P. Cappanera, S. Turchi, and P. Castoldi, "Latency-aware composition of virtual functions in 5g," in *1st IEEE Conference on Network Softwarization (NetSoft)*, 2015, pp. 1–6.

[7] A. Mohammadkhan, S. Ghapani, G. Liu, W. Zhang, K. Ramakrishnan, and T. Wood, "Virtual function placement and traffic steering in flexible and dynamic software defined networks," in *The 21st IEEE Int. Workshop on Local and Metropolitan Area Networks*, 2015, pp. 1–6.

[8] R. Riggio, A. Bradai, T. Rasheed, J. Schulz-Zander, S. Kuklinski, and T. Ahmed, "Virtual network functions orchestration in wireless networks," in *Network and Service Management (CNSM), 2015 11th International Conference on.* IEEE, 2015, pp. 108–116.

[9] A. Tomassilli, N. Huin, F. Giroire, and B. Jaumard, "Energy-efficient service chains with network function virtualization," pp. 1–17, 2016, Inria Res. Report RR-8979.

[10] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright, "Power awareness in network design and routing," in *IEEE Annual Joint Conference of the IEEE Computer and Communications Societies - INFOCOM*, 2008, pp. 1130–1138.

[11] L. Niccolini, G. Iannaccone, S. Ratnasamy, J. Chandrashekar, and L. Rizzo, "Building a power-proportional software router," in *Presented as part of the 2012 USENIX Annual Technical Conference (USENIX ATC 12)*, 2012, pp. 89–100.

[12] A. Dwaraki and T. Wolf, "Adaptive service-chain routing for virtual network functions in software-defined networks," in *Workshop on Hot topics in Middleboxes and Network Function Virtualization (HotMIddlebox)*, 2016, pp. 32–37.

[13] A. Tomassilli, N. Huin, F. Giroire, and B. Jaumard, "Energy-efficient service chains with network function virtualization," in *submitted*, 2017, pp. 1–6.

[14] C. V. N. Index, “Cisco visual networking index: Forecast and methodology 2015-2020,” *White paper, CISCO)*, 2015.

[15] S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly, “SNDlib 1.0–Survivable Network Design Library,” *Networks*, vol. 55, no. 3, pp. 276–286, 2010.

[16] J. Araujo, F. Giroire, J. Moulierac, Y. Liu, and M. Remigiusz, “Energy efficient content distribution,” *Computer Journal*, pp. 1–18, 2016.

[17] B.-Y. Choi, S. Moon, Z.-L. Zhang, K. Papagiannaki, and C. Diot, “Analysis of point-to-point packet delay in an operational network,” *Computer networks*, vol. 51, no. 13, pp. 3812–3827, 2007.

[18] F. Giroire, A. Nucci, N. Taft, and C. Diot, “Increasing the robustness of ip backbones in the absence of optical level protection,” in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 1. IEEE, 2003, pp. 1–11.

RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399