Energy-Aware Routing in Carrier-Grade Ethernet using SDN Approach

Rihab Maaloul*, Raouia Taktak* Lamia Chaari*and Bernard Cousin†

*Laboratory of Technology and Smart Systems (LT2S/CRNS), University of Sfax, Tunisia

†University of Rennes 1, IRISA, France

***Abstract*—Soft-Defined Networking (SDN) is a new approach that enables operators to easily manage all the network elements. In this paper, we address the problem of energy-aware routing in SDN-based carrier-grade Ethernet networks. Our approach is based on turning off network nodes and links to reduce energy consumption, while respecting the rule space capacity for each Openflow switch, and maintaining an allowable maximum link utilization. The problem of identifying the optimal set of network elements to be turned off is NP-hard. We first present an exact model based on an Integer Linear Programming formulation for the problem. Then, we describe a set of *first-fit* heuristic algorithms suitable for large-sized networks. The exact and heuristic approaches are tested on SNDlib-based instances. Experimentations show the efficiency of both exact and heuristic methods for different network topologies. In particular, our heuristic algorithms are able to achieve a good balance between energy consumption, resource utilization, and network performance.**

***Index Terms*—Energy-Aware Routing, Carrier-Grade Ethernet, Software-Defined Networking (SDN), Routing Optimization.**

I. Introduction

ENERGY optimization in carrier-grade networks is becoming a concern in networking. Studies have shown that the energy consumed by carrier-grade networks may reach 50% of the total network power by 2020 [1], [2]. Therefore, reducing the energy consumption of carrier-grade networks has attracted an increasing interest. The energy consumption in a network generally depends on the technology used and elements' power profile. Carrier Ethernet elements present an ON-OFF power profile. This profile fully empowers network devices when they may later be turned on [3], [4]. Consequently, a constant amount of power is consumed when a device is on, regardless of its traffic load. In the case of an ON-OFF power profile, it would be more energy efficient to aggregate traffic on a small set of network devices (line cards and a router chassis) to allow the maximal set to be turned off. Accordingly, Energy Aware Routing (EAR) mechanisms constitute a potential solution to energy consumption minimization. EAR can be implemented and integrated over two architectures (centralized and distributed). Distributed architectures exploit limited amounts of data, relying on multiple agents which are able to locally adjust the sleeping decision. Compared to the distributed architectures, centralized ones dispose of a central controller. Sleeping decisions are carried out in a coordinated way by a central entity who has a global network knowledge. The implementation of an energy-aware routing within an SDN (Software Defined Network) logically centralized architecture can be easily achieved. Carrier-grade network operators specify the need for creating an SDN architecture to facilitate the management and increase the flexibility of their networks [5], [6]. In fact, for the optical transport networks, the Optical Transport working group of the Open Networking Foundation (ONF) [7] emphasizes the improvements in the flexibility of control and management by leveraging virtualization and SDNs.

SDN implementations, in particular using Openflow , focus on carrier Ethernet to optimize its operational expenditures. A detailed description of how Openflow promotes carrier Ethernet advances is provided in [8], [9].

Openflow switches can either be pure or hybrid. Pure switches do not support legacy control protocols and only rely on the Openflow controller for routing decisions, while hybrid switches integrate both. In [10], the authors demonstrate an effective use of SDN for traffic engineering especially when SDN is incrementally introduced into an existing networks. This can be ensured using hybrid switches which are the most deployed in carrier-grade Ethernet [5], [11].

Openflow architecture makes energy-aware routing algorithms less complex due to its logically centralized controller. The Openflow controller can learn network topology and network devices' states, and then compute the best paths in terms of energy savings.

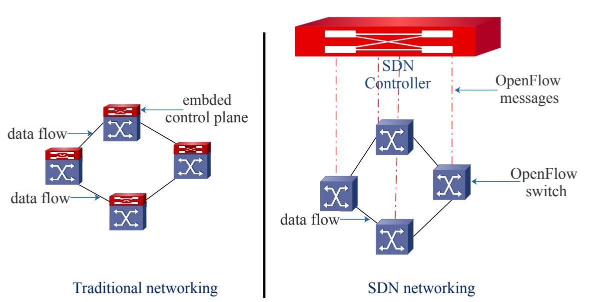


Fig. 1: Traditional networking versus SDN networking

In traditional networks, as illustrated in Fig. 1, the strong coupling between the data and control planes makes the deployment of energy-aware routing algorithms very difficult. It may also become very costly when the numerous devices come from different manufacturers, or when they use different programming interfaces or different protocols. In fact, this would imply a modification of the control plane for all the network devices which act as a closed-system. In contrast, an SDN-based architecture decouples the control plane from the data plane to produce an external entity which is called the SDN controller or the Network Operating System. The logically centralized architecture has the advantage of being consistent with energy-aware traffic engineering. However, to enable energy savings in the Openflow controller, specific features must be controllable by adding extra messages such as the port power status on/off and the adaptive line rate [12]. These messages and their processing add overhead to the control plane and increase the communication delay between the controller and the forwarding devices. Furthermore, the performance of the control plane depends on the size of the flow table embedded in the openflow switches. In our work, we neglect the message exchange issues and consider only the the limitation of flow-table size. In this paper, we use optimization techniques to achieve SDN-based energy-aware routing in carrier Ethernet networks. We first give an Integer Linear Programming (ILP) formulation for the problem that takes into account the rule space capacity constraint, as well as flow conservation and resource utilization constraints. We then introduce a heuristic method that provide near-optimal solutions in a reduced amount of time. As there exists a tradeoff between power savings and network quality of service provisioning, we evaluate the efficiency of our proposed algorithms using diverse performance metrics. These include the network connectivity, the average path length, the average traffic load, and the fairness of traffic distribution.

The sequel is organized as follows. We present related works in the next section. In Section III, we formally describe the problem and model it as an ILP formulation. In Section IV, we describe heuristic algorithms. A performance analysis of the proposed resolution methods is presented in Section V. Finally, Section VI is devoted to giving concluding remarks and new directions for future works.

II. Related Works

A. Energy-Aware Routing in traditional architectures

Energy-Aware Routing strategy refers to smartly routed traffic based on energy-saving objectives. A typical example of EAR consists in modifying the network protocol and turning off unused elements, in order to route traffic over energy efficient paths. Dabaghi et al. [13] categorize EAR approaches that use sleeping techniques into two main types: (i) traffic-unaware algorithms that ignore the network traffic; and (ii) traffic-aware approaches that consider a network traffic matrix in a sleeping decision. Only the works [14], [15], [16], [17], [18] and [19] have considered type (i) of the problem. Although these approaches are able to achieve high energy conservation, they may impact the traffic routing and imply an important congestion on transiting elements especially during high traffic periods.

Type-(ii) EAR approaches, which are the most common, offer a satisfactory level of QoS while achieving a considerable energy efficiency. Typically, a type-(ii) EAR problem in the network is modeled as a graph composed of a set of nodes that are interconnected by a set of directed or undirected links. In this context, using integer linear programming or mixed integer linear programming, the energy saving is formulated as an objective function, while the network's technical requirements are modeled through mathematical constraints. As the EAR problem is NP-hard [20], various heuristics are typically proposed. Chiaraviglio et al. [21] provide a basic formulation of the EAR problem as a capacitated multi-commodity flow (CMCF) problem with continuous flow variables (splittable flows). They propose different heuristics, where a single routing path is considered, based on several sorting policies for turning off both links and nodes. Another variant of sleeping routing algorithms involves turning off both links and nodes, which is considered in [22]. The authors consider the case of flow-based, fully-splittable routing. They propose a MILP-based heuristic that efficiently configures the link weights of an Interior Gateway Protocol to reduce both power consumption and network congestion. As in [22], Moulierac et al. [23] consider an EAR that takes into account link weights optimization. The authors use in addition robust optimization techniques to deal with multi-period traffic variations. In [24], Capone et al. propose an optimization model based on the traditional Multiple Spanning Tree Protocol (MSTP) used by carrier Ethernet networks. They optimize both network congestion and energy consumption (on both links and nodes). The main shortcoming of this approach is the use of MSTP, which can no longer meet the needs of modern carrier Ethernet networks. The aforementioned EAR approaches are assumed to be performed in a coordinated way by a centralized entity. However, none of them have discussed an actual deployment on SND-based architecture networks.

B. Energy-Aware Routing for SDN

In [25], Heller et al. develop the so-called ElasticTree which is one of the most popular approaches that achieve energy efficiency in data center networks. It is implemented on a testbed consisting of Openflow switches. The idea is to turn off links and switches based on the amount of traffic load. The authors show that the traffic flows can be consolidated through a small set of links and switches which are sufficient to serve the bandwidth requests for most of the time. The work in [26] proposes an EAR solution inside Openflow protocol with Green Abstraction Layer (GAL) [27], a recently approved standard of the European Telecommunications Standards Institute (ETSI). This integration permits internal communication between network devices to interchange their power states. In this way, the Openflow controller becomes aware of the energy consumption of each network component. In [28], an extension of the work presented in [26] is proposed including more power states instead of simple ON-OFF states. The authors consider an Openflow protocol that integrates further energy-aware capabilities and power management primitives of the hardware components, line cards, nodes and logical resources. Authors in [29] take the advantage of SDN to create their power management model by collecting real time information about network traffic and users' demands. They propose an ILP formulation that guarantees energy savings for both links and nodes while considering QoS requirements in terms of delay and link utilization constraints. In order to solve the problem in polynomial time, the authors propose global and alternative greedy heuristics. However, they do not consider the limitation of the flow table size, which is one of the main constraints in our model. Typically, EAR approaches assume that the node routing/forwarding table (router/switch) can hold an infinite number of routing rules. However, this assumption does not fit with reality since the actual number of rules in the hardware node is bounded by the Ternary Content Addressable Memory (TCAM) size. In this context, a new EAR approach [30] for SDN-based networks allows only links to be turned off when the rule space constraint is considered. The authors first model the problem in terms of ILP. They also propose a greedy heuristic based on one sorting criterion that iteratively selects the minimally loaded link as a candidate to be turned off.

Recall that using an SDN-based network for EAR offers the major advantage of logically centralized operation. SDN approaches also allow low operating expenses and the flexibility to manage the network and to improve the QoS. In this work, we focus on using Openflow to deploy energy-aware routing in carrier-grade Ethernet networks. Our work can be seen as an extension of [29] considering the rule space capacity, and an extension of [30] offering the possibility to save energy on both links and nodes.

C. Optimizing rule space in Openflow forwarding node

In an Openflow network, the forwarding node contains one or more separated flow tables for handling packets. Starting from version 1.1 and thereafter, Openflow supports a pipeline process consisting of multiple flow tables [5], [11]. Each flow table consists of a set of flow entries that are created by the controller, and that determine how flows will be processed. Each entry in the table corresponds to a routing rule associated with an appropriate action. A flow entry can be divided into three parts: (1) a matching rule that may contain packet header information (e.g., source and destination MAC/IP addresses, and the ingress port); (2) an action to be executed on matching packets (e.g., to output the frame to a specific interface or flood it to all interfaces, to discard the frame, etc.); (3) a counter used to keep statistics on the matching packets. Large tables which are powerful for storing an important number of rules, provide fine-grained flow control and efficient energy-aware traffic engineering. However, it is worth noting that these rules are installed in a TCAM on-chip that is expensive and has limited space to hold a great number of rules. Hence, it would be interesting to optimize the number of rules installed in forwarding devices. TCAM-based energy-aware SDN issues received significant attention as shown in [31]. Some of the works address the problem of rule placement without considering energy savings, see [32], [33] and [34]. In other works, such as [30] and [35], both rule space capacity and energy consumption are optimized. Giroire et al. [30] come with idea of using a default rule to deal with the rule capacity limitation. They have proposed an energy-aware routing algorithm that optimizes the rule placement of an Openflow router in backbone networks. In [35], the authors propose to reduce the size of flow entries and manage large-sized SDN flows, while optimizing only the power consumption induced by the TCAM (without turning off network elements). The authors introduce the Flow-ID concept to enable a new TCAM lookup process that reduces the TCAM power cost.

Our main contribution in this work is to model SDN-based, energy-aware routing in carrier Ethernet networks while respecting the memory limitations in an Openflow switch, which is also known as rule of space capacity. Consequently, it is important to route flows on a single path when the maximum number of rules that can be installed at each node is limited. We use the default rule for optimizing flow tables as in [30]. To the best of our knowledge, the previous works that are the closest to ours are [21], [29] and [30]. Our work is an extension of [29], [30] and [21]. TABLE I gives in details the main common points and differences between our work and those proposed in [21], [29] and [30].

III. Problem statement and formulation

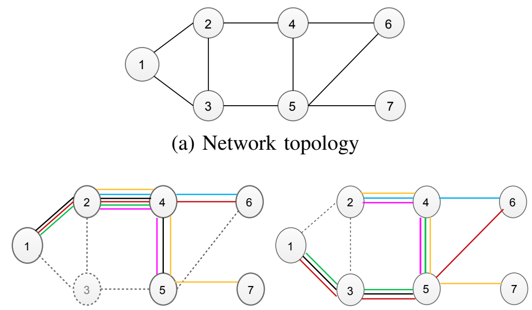
A. Problem statement

As an example of EAR, we consider the network topology shown in Fig. 2a. The capacity of each link is $7Gbps$. There are six traffic demands. Each demand is given by a pair of nodes (the source and destination nodes): $D = \{(1,6),(1,5),(1,4),(2,6),(2,5),(2,7)\}$. All demands have a volume of 1 $Gbps$. When rule space constraints on the flow table are not considered, an optimal EAR routing is obtained as shown in Fig. 2b. In this solution, each demand is routed through its shortest path as follows:

(1,6) : 1-2-4-6 ; (1,5) : 1-2-4-5 ; (1,4) : 1-2-4 ; (2,6) : 2-4-6 ; (2,5) : 2-4-5 ; (2,7) : 2-4-5-7

Fig. 2b illustrates how EAR allows energy savings by turning off node 3 and four links (*i.e.*, (3,1), (3,2), (3,5) and (5,6)). In the obtained solution, the flow table of node 2 stores three routing rules, the flow table of node 4 stores four rules, and the flow table of node 5 stores only one rule.

Now, if we assume that the flow table for each node can store, at most, three routing rules, then node 4 cannot route demands



(a) Network topology

(b) EAR with no rule space constraint (c) EAR with rule space constraint

Fig. 2: Example of EAR

(2,5) and (2,7). Similarly, demands (2,6) and (2,5) cannot be routed via node 2. Note that demand (1,4) does not need to be stored in node 4's flow table as node 4 is a destination. As a consequence, the best EAR solution with the rule space constraint is shown in Fig. 2c and is as follows.

(1,6) : 1-3-5-6 ; (1,5) : 1-3-5 ; (1,4) : 1-3-5-4 ; (2,6) : 2-4-6 ; (2,5) : 2-4-5 ; (2,7) : 2-4-5-7

As shown in Fig.2c, EAR can turn off only two links. Note that, links (1,3) and (3,5), can never be turned off. TABLE II shows the routing rules used by nodes 1 to 5, *i.e.,* each node's flow table contains at most three rules. The flow table of node 6 and node 7 are not reported because they have no demands (rules) to handle.

To address the space limitation issue, one can use, as in [30], default rule to optimize the flow-table size and to enhance the EAR solution. For instance, if we come back to the example in Fig. 2a and apply the default rule to the node flow tables (see Fig. 3 which contains the flow table for node 4), then the routing solution produces exactly the same topology as the one described in Fig. 2b.

In the given example of Fig. 3, before reducing the number of entries in the flow table, we cannot route more than 5 demands according to the available space. To address a large number of flow demands, port 5 is defined as a default port because it initially carried the largest number of rules. Assume that, after shrinking the rule space, we have ten flow demands to route.

TABLE I: Similarities and differences between our work and the closest ones

		Our contribution	**[21]**	**[29]**	**[30]**
Assumptions	**Rule space capacity**	✓	-	-	✓
	Asleep elements	Nodes/Links	Nodes/Links	Nodes/Links	Links only
	Traffic routing	Unsplittable flow (ILP and heuristic)	Splittable flow (MILP) Unsplittable flow (heuristic)	Unsplittable flow (ILP and heuristic)	Unsplittable flow (ILP and heuristic)
Resolution methods	**Exact methods**	ILP (binary variables)	MILP (continuous/binary variables)	ILP (binary variables)	ILP (binary variables)
	Heuristic methods	Sorting policies for network elements (random; least-flow; most-power)	Sorting policies for network elements (random; least-flow; most-power; least-links)	Sorting policies for demands (priority order of delay)	Sorting policies for network elements (least-flow)

TABLE II: Routing rules for Fig. 2c (where each node can store at most three rules)

Node 1		Node 2		Node 3		Node 4		Node 5	
Rule	Action	Rule	Action	Rule	Action	Rule	Action	Rule	Action
(1,6)	port 3	(2,6)	port 4	(1,6)	port 5	(2,6)	port 6	(1,6)	port 6
(1,5)	port 3	(2,5)	port 4	(1,5)	port 5	(2,5)	port 5	(1,4)	port 4
(1,4)	port 3	(2,7)	port 4	(1,4)	port 5	(2,7)	port 5	(2,7)	port 7

A feasible solution will match 4 demands with 4 distinct rules, and the 6 remaining demands will match the default one.

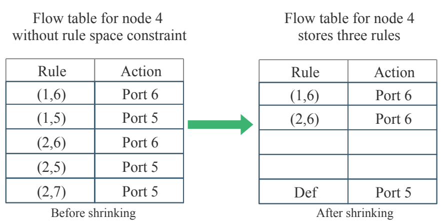


Fig. 3: Stored rules in node 4

B. Binary integer linear programming model

The EAR problem, with the rule space constraint, is formulated as a binary integer linear program. The notations used are shown in TABLE III.

$$\min \sum_{e \in E} \mathcal{E}_e x_e + \sum_{u \in V} \mathcal{E}_u y_u \tag{1}$$

$$\sum_{v \in N_G(u)} [(f_{uv}^{st} - f_{vu}^{st}) + (g_{uv}^{st} - g_{vu}^{st})] = \begin{cases} -1 & \text{if u=s,} \\ 1 & \text{if u=t,} \\ 0 & \text{if u} \neq \text{s,t,} \end{cases} \qquad \forall u \in V, \ \forall (s,t) \in D, \tag{2}$$

$$\sum_{(s,t) \in D} d^{st}(f_{uv}^{st} + f_{vu}^{st} + g_{uv}^{st} + g_{vu}^{st}) \leq \mu C_e x_e \qquad \forall e = (u,v) \in E, \tag{3}$$

$$f_{uv}^{st} + f_{vu}^{st} + g_{uv}^{st} + g_{vu}^{st} \leq 1 \qquad \forall (u,v) \in E, \ \forall (s,t) \in D, \tag{4}$$

$$\sum_{d^{st} \in D} \sum_{v \in N_G(u)} f_{vu}^{st} \leq (R_u - 1) y_u \qquad \forall u \in V, \tag{5}$$

$$\sum_{v \in N_G(u)} k_{uv} \leq 1 \qquad \forall u \in V, \tag{6}$$

$$g_{uv}^{st} \leq k_{uv} \qquad \forall (u,v) \in E, \ \forall (s,t) \in D, \tag{7}$$

$$\sum_{e \in \delta_G(u)} x_e \leq M y_u \qquad \forall u \in V. \tag{8}$$

TABLE III: Summary of notations

Notation	Description
G=(V,E)	Undirected graph where V is the set of vertices (nodes) and E is the set of edges (links)
$\|V\|$, $\|E\|$	$\|V\|$ is the size of V, $\|E\|$ is the size of E
$\mathcal{E}_e$	Power consumption of link e $\in E$
$\mathcal{E}_u$	Power consumption of node u $\in V$
C_e	Capacity of link e $\in$ E
R_u	Maximum number of rules that can be installed in node u $\in$ V
D	Set of all traffic demands $D = \{(s,t), s \in V, t \in V\}$
d^{st}	Traffic demand from node s to t
x_e	1 if link e is in use, 0 otherwise
y_u	1 if node u is in use, 0 otherwise
f_{uv}^{st}	1 if flow (s,t) goes through link (u,v) by a distinct rule, 0 otherwise
g_{uv}^{st}	1 if flow (s,t) goes through link (u,v) by the default rule, 0 otherwise
k_{uv}	1 if the default port of node u goes to v, 0 otherwise
F_u	Set of distinct flows
G_u	Set of default flows
V'	Set of nodes used to route the traffic
E'	Set of links used to route traffic
μ	$\mu \in]0,1]$; maximum tolerated link utilization
$N_G(u)$	Set of neighboring nodes of $u \in V$
$\delta_G(u)$	Incident links to $u \in V$
M	A non-negative, big enough constant

Objective function (1) minimizes the total energy consumed by links and nodes. Constraint (2) expresses the classical flow conservation. It ensures that incoming and outgoing flows are equal for each node except for the source and destination. Inequality (3) says that the sum of traffic for all demands routed through link $e = (u,v)$ must not exceed the tolerated link capacity μC_e. Inequality (4) ensures that the flow passing through link (u,v) is routed using only one rule, which can be either a distinct or a default rule. It also guarantees that the flow for a demand (s,t) is routed in one direction on link (u,v), which can either be from u to v or from v to u. Inequality (5) limits the rule space to a maximum allowed rule space capacity at each node, while keeping only one rule as the default rule. Inequalities (6) and (7) are used to restrict the default port for each node to one. Finally, inequality (8) ensures that when a node u is turned off, none of its incident links can be turned on.

Note that the choice of parameter M is crucial for the experiments. M should be greater than or equal to $\max_{u \in V} |\delta_G(u)|$, or largely $M \geq |V| - 1$.

It is very challenging, and sometimes impossible, to achieve an optimal solution using the previous ILP formulation for large topologies and dense instances. In fact, formulation (1) - (3) falls into the class of multi-commodity *integral* flow problems (see [36]). According to [37], the multicommodity flow problem, with *continuous* flow variables, can be solved in a polynomial time. However, when flow variables are integers, the corresponding decision problem is NP-complete even when considering only two demands and unitary capacities (see [38]). Moreover, if we omit all the coefficients, variables and constraints related to rule space and energy optimization, then we obtain the problem studied in [20], which is proven to be NP-hard. Thus, solving the previous ILP using only exact methods for the resolution is expected to be inefficient. As a consequence, for large topologies, we choose to tackle the problem using heuristic methods.

IV. Heuristic Algorithms

We present a set of *first-fit* heuristic-based algorithms that are practical for large-sized networks. The *first-fit* heuristic is an efficient heuristic that is widely used to solve bin-packing-like problems. It was chosen for this case because it is a straightforward greedy approximation algorithm that can provide a feasible solution in polynomial-time. For more details about the bin-packing optimization problem and the *first-fit* heuristic, the reader may refer to [39], [40].

We propose a centralized implementation of the heuristic algorithms into an Openflow controller. First, the controller collects information on the network topology and the user traffic demands. Then, the controller runs the heuristic to find a subset of selected nodes and links to route traffic demands. In Fig. 4, we present the software architecture running inside SDN-based network. There are three layers in an SDN architecture; (i) *Application layer* transfers requirements to the controller using an open application programming interface (north-bound API) that allows a better orchestration of network resources, (ii) *Control layer* maps the application requirements to the network resources, (iii) *Infrastructure layer* (data plane), consists of heterogeneous network devices that support an open Southbound API, *i.e.*, Openflow protocol. Note that implementing energy saving heuristic algorithms will mainly involve the application modules (Topology, EAR, users' requests, statistics information).

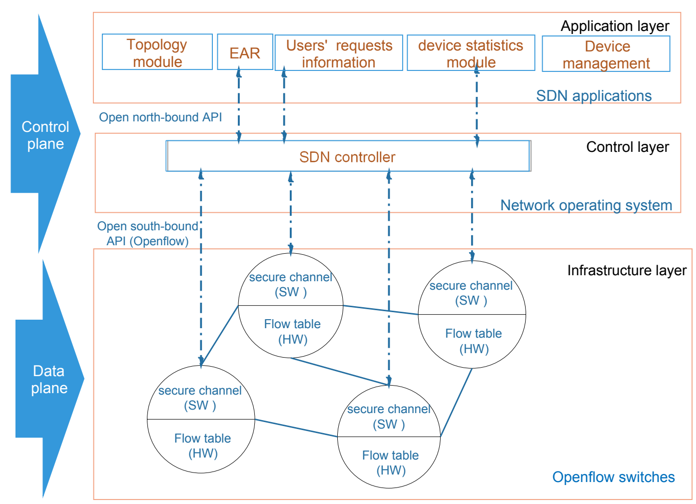


Fig. 4: Software architecture running inside SDN-based network

Fig. 5 contains a diagram description of our proposed heuristics. Step1 uses Dijkstra's algorithm [41] to route traffic demands through the shortest paths; it requires $O(|D||E|.log|V|)$. Step2 sorts the elements according to a

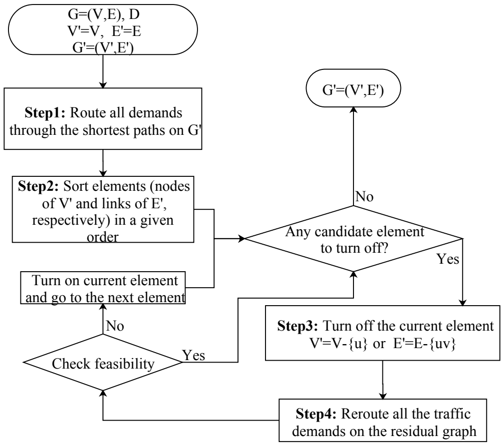


Fig. 5: Diagram of our heuristic.

given criterion and has a complexity of $O(|E| + |V|)$. Step3 requires $O(1)$ because the candidate element for being turned off can be found using the list head from Step2. Step4 uses Dijkstra's algorithm at most $|D|$ times.

Note that a crucial step for this *first-fit* heuristic is the way the elements are sorted. In our algorithms, we choose three criteria to sort nodes and links:

1) First-Fit Most-Power (MP): iteratively selects the element with the highest power consumption.
2) First-Fit Least-Flow (LF): iteratively selects the element with the smallest amount of traffic already routed through it. This selection criterion is used by [30] to sort candidate links.
3) First-Fit Random (R): randomly selects an element. Here, Step2 is neglected because it does not need to sort the network elements.

TABLE IV summarizes the combined node/link sorting policies. The columns correspond to the nodes' criteria and the rows to the links' criteria.

TABLE IV: Combination of sorting criteria for the first-fit heuristics

links \ nodes	**MP**	**LF**	**R**
LF	MP-LF	LF-LF	R-LF
MP	MP-MP	LF-MP	R-MP

Input: G=(V,E), initial flow tables and rule capacity R_u for all $u \in V$, link capacity C_e for all $e \in E$, and a set D of demands with traffic requirements d^{st} for all $(s,t) \in D$.
Output: G'=(V',E'): the output graph containing only elements used to route the demands.

```
initially, the remaining link capacity Cr_e = C_e for all e ∈ E;
/*Node optimization*/
sort nodes according to a predefined order in node-list;
for (i=1; i <= |V|; i++) do
    turn off (node-list[i]);
    for each (s,t) ∈ D do
        path(s,t)=compute the best possible path from s to t ;
        if !path (s,t) then
            turn on (node-list);
        else
            update the graph and flow tables using Algorithm 2 ;
        end
    end
end
/*Link optimization*/
sort links according to a predefined order in node-list;
for ( j=1 ; j<= |E|; j++) do
    turn off (link-list[j]) ;
    for each (s,t) ∈ D do
        path(s,t)=compute the best possible path from s to t ;
        if !path (s,t) then
            turn on (link-list[j]) ;
        else
            update the graph and flow tables using Algorithm 2 ;
        end
    end
end
```

Algorithm 1: First-fit heuristic-based algorithms.

For example, the MP-MP heuristic selects respectively the node and the link that consumes the highest amount of power as a candidate to be powered off. Hence, V and E are sorted according to decreasing values of $\mathcal{E}_u$, $\mathcal{E}_e$ respectively. The LF-LF heuristic turns off elements (nodes and links) with increasing values of traffic that was already routed through each element. Algorithm 1 describes, in detail, the different steps of our heuristics.

We start from the whole network by considering the initial

Input: A subgraph G'' G computed during the turning off step, the path p(s,t), rule capacity R_u, the default port def(u) for all $u \in V$, remaining link capacity Cr_e, and link capacity C_e for all $e \in$ E.
Output: Updated flow tables and updated sets of distinct F_u and default G_u flows.

1 assign the route p(s,t) to the demand (s,t) ;
2 update $Cr_e = Cr_e - d^{st}$ for all $e \in$ p(s,t) ;
3 **for** *each* $u \in p(s,t)$ **do**
4   **if** $|F_u| == R_u$ **then**
5   adjust the flow table of the node u as illustrated in Fig. 3;
6   **end**
7   **for** *each* $v \in N''_G(u)$ **do**
8   **if** *((u,v)* $\in$ *p(s,t) AND def(u) ==v)* **then**
9             $G_u = G_u \cup\ (s,t)$;
10   **else**
11   **if** *((u,v)* $\in$ *p(s,t) AND def(u)*$\neq$ *v)* **then**
12                 $F_u = F_u \cup\ (s,t)$;
13   **end**
14   **end**
15   **end**
16 **end**

Algorithm 2: Updating flow tables, F_u, and G_u

flow tables and assuming that all elements are turned on. After sorting the elements based on a given criteria, we next apply the following procedure for nodes and then for links. At each iteration, we remove (*i.e.*, turn off) the first element in the ordered set. Then, we compute, for each demand (s,t), the best possible path along the residual network topology as described in Algorithm 1. The best path is the shortest path that satisfies inequalities (2)-(4). If no path exists, then the removed element is put back into the network. For the sake of simplicity and without loss of generality, when routing we consider that the weights of all links are equal to one. When a shortest path is found, the remaining capacity of the links is updated as described in Algorithm 2. Recall that, for each node u, the two sets F_u and G_u denote distinct and default flows respectively (see TABLE III). Initially, flow entries are created without hindrance until the flow table becomes full, and then there is no available space to assign a new rule. Then, the flow table is adjusted (line 4, Algorithm 2) by selecting the port that carries the largest number of flows, as the default port. This step has been previously described in Fig. 3.

V. Performance analysis

In this section, we evaluate the ILP formulation and the heuristic-based algorithms. First, we describe the considered performance metrics and the experimental scenarios. Our goal is to accomplish the following evaluations:

1) a general performance analysis of the ILP model on different network instances that consider different rule space capacities;
2) a comparison of the solutions obtained using the ILP formulation with those obtained using the heuristics on the same network instances;
3) a general performance analysis of the heuristic solutions for large networks.

A. Performance metrics

The performance of the proposed resolution approaches is evaluated using five performance metrics. The first two metrics indicate the percentage of energy savings that can be obtained.

- $\eta_{L_{off}}$ is the percentage of energy savings related to the links turned off by our EAR algorithms. It is computed as follows:

$$\eta_{L_{off}} = \frac{\sum\limits_{e \in E} \mathcal{E}_e - \sum\limits_{e \in E'} \mathcal{E}_e}{\sum\limits_{e \in E} \mathcal{E}_e} \times 100. \tag{9}$$

- $\eta_{N_{off}}$ is the percentage of energy savings related to the nodes turned off by our EAR algorithms. It is computed as follows:

$$\eta_{N_{off}} = \frac{\sum\limits_{u \in V} \mathcal{E}_u - \sum\limits_{u \in V'} \mathcal{E}_u}{\sum\limits_{u \in V} \mathcal{E}_u} \times 100. \tag{10}$$

The third metric, denoted by $\lambda_2(G)$, represents an important characteristic of graphs, which is the connectivity. This parameter can be computed using the Laplacian matrix of the undirected graph G , denoted by L_G [42]. In graph theory, L_G is equal to the difference between the degree matrix D_G

and the adjacency matrix A_G, *i.e.*, $L_G = D_G - A_G$. A_G is a square binary matrix $|V| \times |V|$, where the generic matrix element a_{ij} indicates if vertices i and j are adjacent in the graph. The degree matrix D_G of G is the diagonal matrix such that $d_{ii} = \sum_{j \in V} aij$. The Laplacian matrix of an undirected graph is symmetric with real eigenvalues. The eigenspectrum $\lambda(G)$ of L_G is defined as the set of its $|V|$ eigenvalues, which can be ordered sequentially in an ascending order $(\lambda_1(G) \leq \lambda_2(G) \leq \ldots \leq \lambda_V(G))$. For a connected graph G, $\lambda_2(G) > 0$. The second smallest eigenvalue λ_2 is called the *algebraic connectivity* of the graph [43].

In our case, the computation of λ_2 enables to control the connectivity of the active part of the network.

As load balancing is a requirement that should be fulfilled in carrier Ethernet, the fourth metric is devoted to measuring the fairness of traffic distribution on the active links E'. The fairness index FI measures if the traffic load is fairly distributed among all the links. In our performance analysis, we use Jain's Fairness Index [44], which is given by:

$$FI = \frac{(\sum_{e \in E'} l_e)^2}{|E'| \times \sum_{e \in E'} {l_e}^2}, \tag{11}$$

where l_e is the percentage of traffic utilization of link $e \in E'$.

Note that, when $FI = 1$, the traffic is distributed in a fair way.

The last metric to be introduced is related to the increase of route length. Consider a demand $(s,t) \in D$, then we define $\phi^{st} = L_2^{st} - L_1^{st}$, where L_1^{st} is the length of path routing demand (s,t) using the shortest path without considering EAR. L_2^{st} is the length of the path routing (s,t) using our EAR algorithms. L_1^{st} and L_2^{st} are given in terms of hops. Note that for $(s,t) \in D$, $L_2^{st} \geq L_1^{st}$. This is obvious as EAR algorithms may turn off some elements of the graph, which may increase the length of paths.

B. Experimental context

We solve the ILP model using the solver CPLEX with Concert Technology (C++) [45]. Note that Cplex is a solver that uses exact methods of resolution to solve integer, mixed integer and quadratic programs [46]. The time limit is set to 3 hours (10800 seconds), and M parameter is set to $|V|-1$. The heuristic algorithms are implemented using MATLAB. All the experiments are performed on a PC with 2.6 GHz Intel Core i7 and 8GB RAM.

Data for the real network topology used by ISPs are considered confidential, so they are not easily revealed. Consequently, we consider realistic network instances collected from SNDlib [48]. TABLE V presents the main properties of the used network topologies.

TABLE V: Properties of network topologies

Network instance	$\lvert V \rvert$	$\lvert E \rvert$	$\lvert D \rvert$	**Traffic matrix origin origin**	**Link capacity (units)**
Abilene	12	15	132	6:00 am of Sept 04^{th} 2004	[2480-9920]
Atlanta	15	22	210	given by SNDlib	[575000-3200000]
Di-yuan	11	42	22	given by SNDlib	[8200-159300]
France	25	45	300	given by SNDlib	2500
Germany50	50	88	662	6:00 am of Feb 15^{th} 2005	[4150-3290]
Nobel-germany	17	26	121	6:00 am of Feb 02^{nd} 2005	600
Nobel-us	14	21	91	given by SNDlib	[3580-20350]
Pdh	11	34	24	given by SNDlib	1920
Polska	12	18	66	given by SNDlib	[4260-6804]

We consider two main types of traffic matrices:

- $TM1$: is a meshed traffic matrix, *i.e.,* every node of the network appears at least in one demand as a source or destination. $TM1$ is nothing but the traffic matrix provided by SNDlib for the chosen networks.
- $TM2$: is generated from $TM1$ so that some randomly chosen nodes (from 10% to 15% of $|V|$) are assumed to be pass-through nodes (transit nodes, *i.e.*, neither source nor destination of any demand). To generate the traffic matrix TM2, we first begin by choosing the set of nodes that will be considered as pass-through, i.e. transit nodes. The corresponding demands in TM1 are then removed and replaced in TM2 by new ones, randomly generated, in order to maintain the same number of demands for each

topology. TABLE VI presents the percentage of through-pass nodes.

TABLE VI: Percentage of pass-through nodes for $TM2$

Abilene	Atlanta	Di-yuan	France	Germany50	Nobel-germ.	Nobel-us	Pdh	Polska
10%	10%	12%	10%	10%	12%	15%	10%	15%

We assume that the daily traffic patterns have the shape of Fig. 6 taken from [49]. Note that the traffic matrices found in SNDlib are collected at 6:00 a.m. In order to fit the best to reality and represent the daily traffic levels, we scale $TM1$ and $TM2$ with parameter γ ranging from $[0.25, 2.5]$.

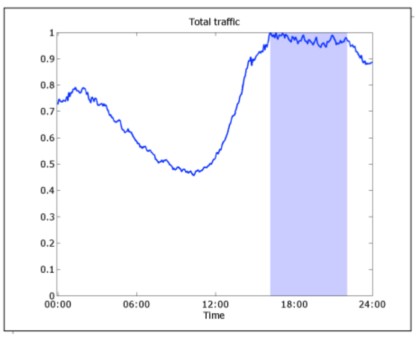


Fig. 6: Daily traffic for different networks

We also assume, as in [30], that the rule capacity of each flow table is $R_u = (\rho \times |D|)$ where $\rho \in]0, 1]$.

In all the experiments, we use the same estimation of the power consumption as in [24]. The power consumption of a single line card is 150 Watts, therefore, the power consumption of a link e is $\mathcal{E}_e$= 300 Watts. While the consumption of node v is assumed to be $\mathcal{E}_v$=(1200 + $|\delta(v)|$) Watts, where $\delta(v)$, we eval is the degree of v.

C. Computational results

In this section, we present the performance results to confirm the effectiveness of our algorithms. We start with a demonstration on the smallest test instance (*i.e.*, Abilene network). Then, we compare the performance of the ILP model with the heuristic algorithms on nine different network topologies. Finally, we present a substantial evaluation of the heuristics with respect to the different network performance previously defined.

1) Optimal vs. heuristic solutions for Abilene network

As a first experimental evaluation, we consider the ILP model and the heuristics solutions for Abilene Network ($|V| = 12, |E| = 15, |D| = 132$), using $TM1$ and varying the rule space capacity. Fig. 7 and Fig. 8 present the produced topologies after applying the ILP and MP-MP heuristic algorithms to the Abilene Network instances with rule capacities $\rho = 9\%$, $\rho = 20\%$, and $\rho = 100\%$ respectively. In Fig. 7 and Fig. 8, the continuous lines represent the links used in the final solution to route all demands. The dashed lines are links that appeared in the original graph and that have been turned off during the optimization process. For the different values of ρ, both algorithms (ILP and MP-MP heuristic) give solutions with always 26.5% of links turned off. However, we notice through Fig. 7 and Fig. 8, that the obtained solutions for the different rule spaces are not the same. In fact, the produced sub-graphs are different for the various rule spaces. This is obvious because when the rule capacity value ρ changes, the flow table size changes as well, therefore producing different routing solutions for the same instance.

We also notice that, for all the cases, the obtained sub-graphs are always full-covering trees. Recall that, for this first set of experiments, we use a fully-meshed traffic matrix (*i.e.*, $TM1$), which implies that all the nodes must be turned on for all the solutions. All the obtained solutions are full-covering trees, which means that we succeed in routing all the demands using the minimum number of links that guarantee network connectivity (*i.e.*, $|E'| = |V| - 1$).

Fig. 9a and Fig. 9b illustrate the distribution of metric ϕ computed for Abilene instances ($\rho = 9\%$, $\rho = 20\%$ and $\rho = 100\%$) using ILP and MP-MP algorithms respectively. Obviously, using EAR algorithms increases the routing path lengths, which can, for some few demands, reach 9 extra hops compared to the shortest path routes. However, more than 70%

of the demands have a reasonable number of extra hops that ranged from 0 to 4.

In summary, for the first experiment, the ILP and heuristic algorithms performed similarly. Both achieve the maximum possible energy savings without violating any operational constraint.

(a) Small flow table ρ = 9%

(b) Medium flow table ρ = 20%

(c) Large flow table ρ = 100%

Fig. 7: The Abilene Network using the ILP model

(a) Small flow table ρ = 9%

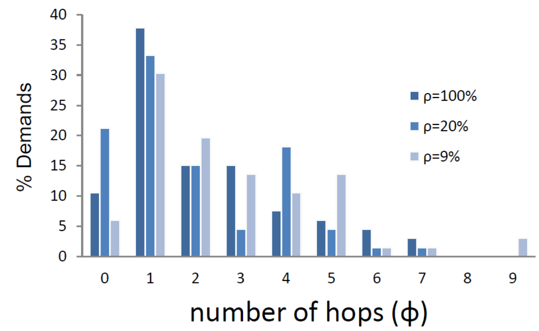
(b) Medium flow table ρ = 20%

(c) Large flow table ρ = 100%

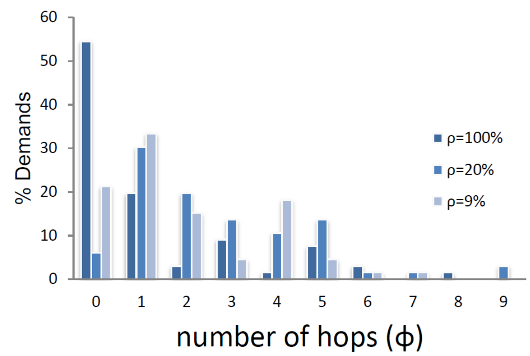
Fig. 8: The Abilene Network using MP-MP heuristic

2) *Optimal vs. heuristic solutions for various network topologies*

To thoroughly compare the ILP and heuristic-based algorithms, we evaluate their performances on nine different network instances using the two traffic matrices $TM1$ and



(a) ILP model



(b) MP-MP heuristic

Fig. 9: Paths hops increase for Abilene Network

$TM2$. As known in practice, network operators do not run their networks at full utilization to avoid transient congestion. In our work, the maximum allowed utilization of links is set to 70% (μ = 0.7), which guarantees normal network operation. Results are reported in TABLE VII, TABLE VIII, TABLE IX, and TABLE X. Entries for the tables are the following.

The first column indicates the network instance characteristics. The second column gives the rule capacity ρ which is set to the three values 9%, 20%, and 100%. The optimum column indicates if the optimal solution is found (only in TABLE VII and TABLE IX). The sorting criteria column indicates the sorting policies used to run the heuristic (only in TABLE VIII and TABLE X). The energy savings column reports the percentage of turned off nodes $\eta_{N_{off}}$ and edges $\eta_{L_{off}}$. $\lambda_2(G)$ and $\lambda_2(G')$ columns report the network connectivity before and after running the EAR algorithms. In other words, $\lambda_2(G)$ is the initial graph connectivity, and $\lambda_2(G')$ is the computed graph connectivity. $\lambda_2(G)$ and $\lambda_2(G')$ are computed only for the fully meshed matrix, which is the case of $TM1$ (TABLE VII and TABLE VIII). The gap column is computed as the ratio (UB-LB)/LB, where UB is the upper bound on power consumption, (the power consumption of the sub-graph solution), and LB is the lower bound on power consumption (the power consumption of the linear relaxation). Finally, the time column gives the computation time in seconds.

TABLE VII and TABLE VIII report the computational results obtained by running the ILP and heuristic algorithms respectively for $TM1$. First, note that for all the instances, the percentage of nodes turned off using both algorithms is $\eta_{N_{off}} = 0\%$. This is obvious since the traffic matrix $TM1$ is fully meshed; therefore, no node can be turned off.

During the experiments for all network topologies except for France and Germany50, we remark that the number of links used to route the traffic is $|V| - 1$. As discussed earlier, this is the minimum number of links needed to route a fully meshed traffic matrix (such as $TM1$). We also observe that, when the original graph is dense (*i.e.*, $\lambda_2(G)$ is high), the percentage of turned off links is important (see, for instance, Di-yuan and Pdh networks).

The impact of rule space can be noticed particularly for the France and Germany50 instances. Clearly, we notice that $\eta_{L_{off}}$ increases when ρ increases as well. We can explain this by the fact that, when providing more rule space, routing the demands would be more flexible and would use fewer links. Having more rule space also makes it easier to test instances. For example, with Atlanta or Nobel-us networks, when the rule space is scarce ($\rho = 9\%$), the ILP cannot reach optimality within the time limit. However, the same networks, when $\rho = 20\%$ and $\rho = 100\%$ are solved to optimality before reaching the time limit.

In TABLE VIII, we report the results obtained using the heuristic-based algorithms and all the possible combinations of the sorting criteria given in TABLE IV. In particular, we report the best obtained solutions, in terms of energy savings and computation times, among all the combinations of sorting criteria. Note, however, that we obtain the same energy savings for the majority of combinations, but sometimes with different sub-graph solutions, (*i.e.*, different values of $\lambda_2(G')$).

As a first observation, the heuristic algorithms represent encouraging results in terms of execution times. In addition, for France and Gemany50 networks, our heuristics achieve a higher percentage of energy savings compared to those achieved with the ILP model (the ILP model is stopped before reaching optimality due to the large network size).

TABLE IX and TABLE X report computational results obtained by running the ILP and heuristic algorithms respectively using $TM2$. Note that for these tables, we do not report the values of graph connectivity, *i.e.*, $\lambda_2(G)$ and $\lambda_2(G')$ because the latter are not significant in this case. In fact, since $TM2$ is a sparse traffic matrix, some nodes act as pass-through nodes in the routing process, and hence, turning off these nodes improves the energy conservation. We notice that a significant gain of energy saving is achieved with both algorithms. For $TM2$ like $TM1$, the impact of rule space is also noticed for France and Germany50 networks. As expected, the resulting energy savings increase when the rule space also increases.

When analyzing the results reported in TABLE VII to TABLE X, we can state that the heuristic algorithms provided energy saving values better than or equal to those obtained with the ILP model within reasonable computation times. Moreover, the heuristic results, especially those obtained for France and Germany50, demonstrate the efficiency of our heuristics on large-sized instances. Through the obtained results we also observe that the performance of our heuristics is influenced by the number of demands, such as Atlanta ($|D|$=210), France ($|D|$=300) and Germany50 ($|D|$=662). This is obvious since the heuristic algorithms are based on a demand re-routing process after turning off selected nodes/links at each iteration.

3) Heuristics performances analysis

In what follows, we evaluate our heuristics with France, Germany50, and Nobel-germany networks using $TM1$ and based on the performance metric ϕ. The increase in the path lengths for these networks using the MP-MP heuristic algorithm is reported in Fig. 10.

We first remark that a significant fraction of demands (30% to 50%) is not affected by path length increase ($\phi = 0$). However, the increase in path length reaches for a small fraction of demands an important number of hops. An example is Germany50 Network, where the path length increases by 20 hops. Consequently, restrictions on the maximum number of hops should be considered in the future using additional constraint especially for large-sized networks.

We can limit the path length (in terms of hop) by adding the following constraint:

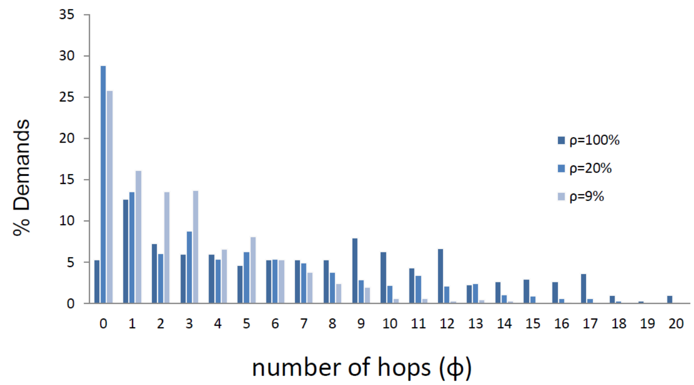
$$\sum_{(u,v)\in E} (f_{uv} + f_{vu} + g_{uv} + g_{vu}) \leq L^{st} \quad \forall (s,t) \in D. \tag{12}$$

We also can limit paths length increase using the following path delay constraint:

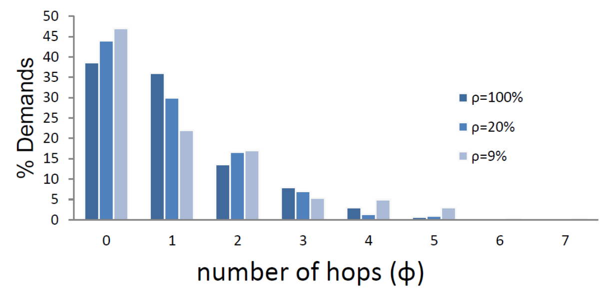
$$\sum_{(u,v)\in E} (f_{uv} + f_{vu} + g_{uv} + g_{vu}).lat_{uv} \leq latency_{st} \quad \forall (s,t) \in D, \tag{13}$$

where lat_{uv} is the edge delay and $latency_{st}$ is the delay of the demand (s,t).

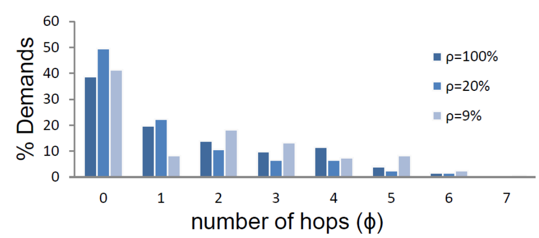
Further analysis is needed to evaluate the different sorting criteria used for the heuristic algorithms that are always applied to France, Germany50 and Nobel-germany networks. To this end, we evaluate the different heuristics performances when the maximum link utilization μ on the network varies. In Fig. 11, we present the percentage of turned off links using the heuristic algorithms when considering different combinations of sorting criteria. We observe that all combinations show identical results for $\mu \geq 0.6$. Otherwise, MP-MP and MP-LF prove to be the most efficient heuristics. For $\mu > 0.65$ no improvements in terms of energy savings is noticed because the traffic demand requirements imply a limitation of the number of links that can be turned off.



(a) Germany50 Network



(b) France Network



(c) Nobel-germany Network

Fig. 10: Paths hops increase by MP-MP heuristic using $TM1$

To evaluate the heuristics performance for the daily variations in traffic between day and night, we scale the traffic matrices ($TM1$ and $TM2$) by γ while setting $\mu = 0.7$ for all network instances. We use the MP-MP heuristic, which gives, in most cases, the best results among all combinations

TABLE VII: ILP formulation using $TM1$

Network	Rule capacity (ρ%)	Optimum	Energy Saving $\eta_{N_{off}}$(%)	Energy Saving $\eta_{L_{off}}$(%)	Graph connectivity $\lambda_2(G)$	Graph connectivity $\lambda_2(G')$	Optimality gap (%)	Power consumption Upper bound UB (W)	Execution Time (s)
Abilene	9	yes	0	26.65	0.309	0.13	0	17730	2.22
	20	yes				0.176	0		1.93
	100	yes				0.086	0		1.83
Atlanta	9	no	0	36.35	0.422	0.0467	2.6	22244	10800
	20	yes				0.0706	0		1961.49
	100	yes				0.0642	0		1893.13
Di-yuan	9	no	0	76.15	5.793	0.1023	7.4	16284	10800
	20	no				0.0741	7.1		10800
	100	no				0.0938	7.5		10800
France	9	no	0	33.33	0.350	0.1267	7.9	39090	10800
	20	no	0	35.55		0.0416	7.7	38790	10800
	100	no	0	37.75		0.0423	7.1	38490	10800
Germany50	9	no	0	31.8	0.182	0.029	15.7	78476	10800
	20	no	0	32.95		0.055	12.9	77876	10800
	100	no	0	34.05		0.046	10.07	77576	10800
Nobel-germany	9	yes	0	38.45	0.301	0.037	0	25252	942.541
	20	yes				0.063	0		1076.54
	100	yes				0.049	0		10248.3
Nobel-us	9	no	0	38.05	0.7326	0.113	2.1	20742	10800
	20	yes				0.064	0		5013.33
	100	yes				0.018	0		942.541
Pdh	9	no	0	70.55	2.524	0.127	5.5	16268	10800
	20	no				0.1857	4.6		10800
	100	no				0.145	5.4		10800
Polska	9	yes	0	38.85	0.7125	0.0805	0	17736	78.6207
	20	yes				0.1318	0		34.6595
	100	yes				0.126	0		42.713

TABLE VIII: Heuristic algorithms using $TM1$

Network	Rule capacity (ρ%)	Sorting criteria	Energy Saving $\eta_{N_{off}}$(%)	Energy Saving $\eta_{L_{off}}$(%)	Graph connectivity $\lambda_2(G)$	Graph connectivity $\lambda_2(G')$	Power consumption Upper bound UB (W)	Execution Time (s)
Abilene	9	R-MP	0	26.65	0.309	0.269	17730	<0.60
	20	R-MP				0.258		
	100	R-MP				0.070		
Atlanta	9	R-MP	0	36.35	0.422	0.046	22244	< 135
	20	R-LF				0.070		
	100	R-LF				0.064		
Di-yuan	9	R-LF	0	76.15	5.793	0.162	16284	< 82
	20	R-LF				0.128		
	100	R-LF				0.137		
France	9	R-LF	0	46.65	0.350	0.095	37290	< 1161
	20	R-MP				0.087		
	100	R-LF				0.095		
Germany50	9	R-LF	0	44.3	0.182	0.090	74876	< 9310
	20	R-LF				0.011		
	100	R-LF				0.034		
Nobel-germany	9	R-LF	0	38.45	0.301	0.087	25252	< 112
	20	R-MP				0.082		
	100	R-MP				0.056		
Nobel-us	9	R-LF	0	38.05	0.7326	0.186	20742	< 53
	20	R-LF				0.123		
	100	R-LF				0.171		
Pdh	9	R-MP	0	70.55	2.524	0.154	16268	< 45
	20	R-MP				0.185		
	100	R-LF				0.026		
Polska	9	R-MP	0	38.85	0.7125	0.117	17736	< 7
	20	R-LF				0.092		
	100	R-LF				0.092		

TABLE IX: ILP formulation using $TM2$

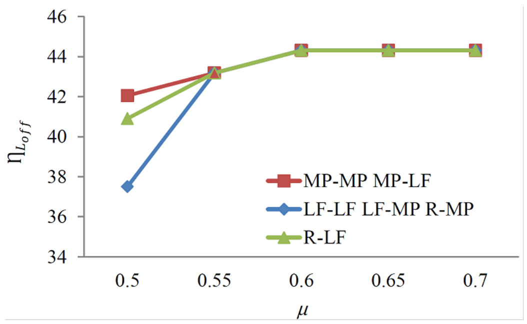
Network	Rule capacity (ρ%)	Optimum	Energy Saving $\eta_{N_{off}}$(%)	Energy Saving $\eta_{L_{off}}$(%)	Optimality gap (%)	Power consumption Upper bound UB (W)	Execution Time (s)
Abilene	9 20 100	yes	8.33	33.33	0	16528	1.25 0.54 0.13
Atlanta	9 20 100	yes	6.66	40.90	0	20740	171.56 635.06 3514.68
Di-yuan	9 20 100	yes	09.09	78.57	0	14776	3791 10800 10800
France	9 20 100	no	8 8 8	46.66 44.44 44.44	4.4 3.8 3.8	35183 34883 34883	10800 10800 10800
Germany50	9 20 100	no	10 6 6	42.04 39.77 39.77	27.1 27.5 12.9	70668 70668 69454	10800 10800 10800
Nobel-germany	9 20 100	yes	11.76	46.15	0	22244	14.63 18.22 14.51
Nobel-us	9 20 100	yes	14.28	47.61	0	17736	370.08 210.08 1387.4
Pdh	9 20 100	yes	9.09	73.52	0	14762	6222.73 6474.38 4384.39
Polska	9 20 100	yes	16.66	50	0	14728	19.92 14.55 3.42

TABLE X: Heuristic algorithms using $TM2$

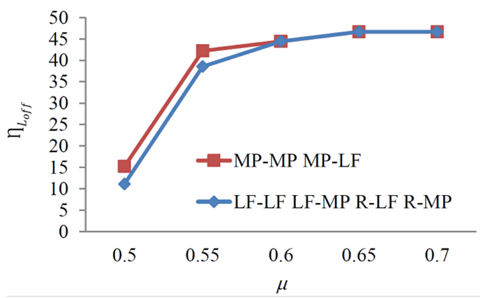
Network	Rule capacity (ρ%)	Sorting criteria	Energy Saving $\eta_{N_{off}}$(%)	Energy Saving $\eta_{L_{off}}$(%)	Power consumption Upper bound UB (W)	Execution Time (s)
Abilene	9 20 100	MP-LF MP-LF LF-LF	8.33	33.33	16528	< 1
Atlanta	9 20 100	MP-LF LF-LF MP-LF	6.66	40.90	20740	< 25
Di-yuan	9 20 100	R-MP R-MP LF-MP	09.09	78.57	14776	< 40
France	9 20 100	MP-LF MP-LF MP-MP	8	51.11	34577	< 743
Germany50	9 20 100	MP-LF MP-LF MP-LF	10	50	67359	< 11282
Nobel-germany	9 20 100	LF-LF MP-MP R-MP	11.76	46.15	22244	< 10
Nobel-us	9 20 100	MP-LF MP-LF MP-LF	14.28	47.61	17736	< 27
Pdh	9 20 100	R-MP MP-MP MP-LF	9.09	73.52	14762	< 49
Polska	9 20 100	MP-MP LF-LF MP-LF	16.66	50	14728	< 15

of the sorting criteria. Fig. 12 reports the percentage of turned off links using $TM1$. While Fig. 13 and Fig. 14 report the percentage of turned off links and nodes, respectively, using $TM2$.

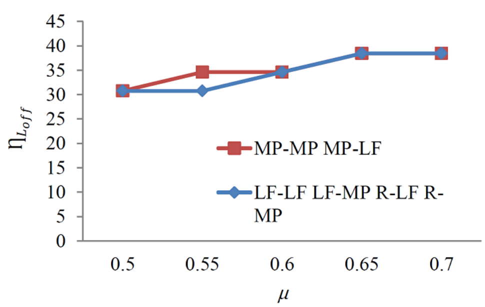
In Fig. 12 and Fig. 13, the obtained results are as expected. When the matrix factor increases for France, Nobel-germany, Nobel-us, and Polska networks, the energy savings are reduced. However, for the other networks except Abilene, the percentage of turned off links remains almost the same for the different values of γ. This is obviously due the fact that link capacities for the these networks are sufficient to satisfy the high-valued traffic demands. Only for Abilene Network, no feasible energy savings can be achieved for high-valued traffic demands. In Fig. 14, we notice that, for



(a) Germany50 Network



(b) France Network



(c) Nobel-germany Network

Fig. 11: Turned off links versus μ using different combinations of sorting criteria for $TM1$

France, Germany50, Nobel-germany, Nobel-us, and Polska networks, some pass-through nodes have to be turned on to route a large number of traffic demands. For the other networks except Abilene Network, it is possible to route a large number of traffic demands with the same number of nodes. Concerning Abilene instances, no feasible solution can be found when $\gamma \geq 2$ because of the link capacity constraint.

Finally, we evaluate the impact of rule space capacity in terms of load balancing. Fig. 15 describes the fairness index FI behavior as a function of ρ for the nine networks running the MP-MP heuristic. When the rule capacity decreases, the traffic demand is routed through the allowed ports according to the matching rule in the flow table, and consequently an unfair traffic distribution is resulted. Based on the results of Fig. 15, we observe that the heuristic solutions maintain a good fairness index that ranged from 0.45 to 0.8.

VI. Conclusion

In this paper, we present an energy-aware routing solution that is compliant with SDN-based carrier Ethernet networks. We first propose a binary linear programming formulation for the EAR problem that maximizes the number of network elements to be turned off, while respecting traffic demand and rule space constraints. Since identifying the optimal set of nodes and links to be turned off is an NP-hard problem, along with the ILP model, we propose a set of *first-fit* heuristic algorithms to reduce the computation time. We also discuss some EAR implementations in an SDN controller. Both ILP algorithm and heuristics are tested on nine realistic network topologies from SNDlib taking into account the rule space constraint. Our algorithms balance between saving energy and link utilization constraints while respecting the size limitation of flow tables. Experiments prove that the heuristics are appropriate for achieving energy efficient routing in carrier-grade networks. Based on the obtained results, which are encouraging, we aim, as a next step, to implement the proposed heuristics via a network emulator (using a POX controller). In addition, as a future work, it would be interesting to include restrictions on the maximum length of paths, which can be ensured by the delay or the hop constraints. Moreover, one could improve the deployment of EAR by considering the so-called reliability constraint which is one of the crucial requirements for carrier Ethernet networks.

References

[1] HINTON, Kerry, BALIGA, Jayant, FENG, Michael, et al. "Power consumption and energy efficiency in the internet," IEEE Network, 2011,

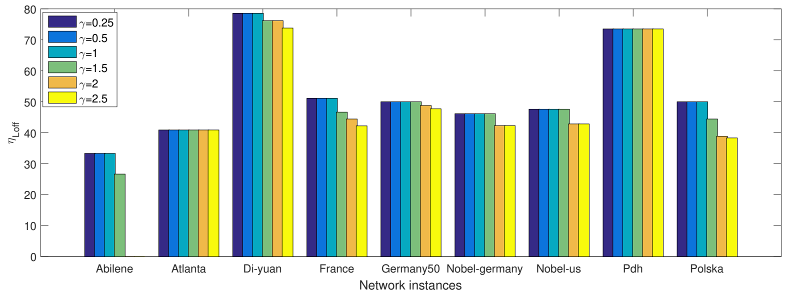


Fig. 12: Turned off links $\eta_{L_{off}}$ using $TM1$ scaled by γ with the MP-MP heuristic

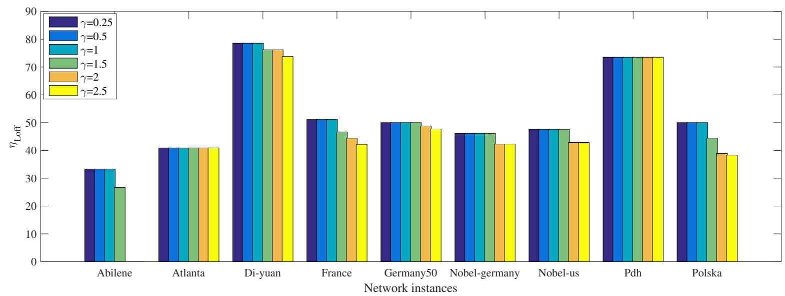


Fig. 13: Turned off links $\eta_{L_{off}}$ using $TM2$ scaled by γ with the MP-MP heuristic

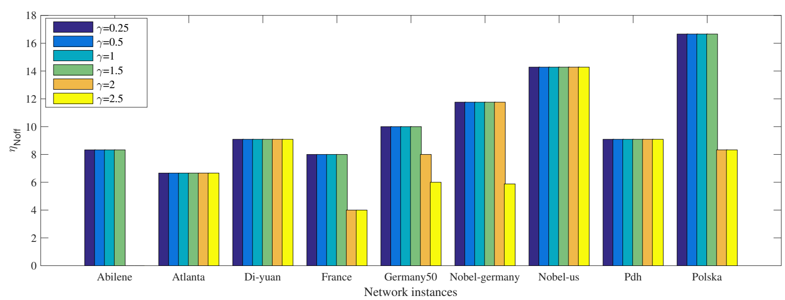


Fig. 14: Turned off nodes $\eta_{N_{off}}$ using $TM2$ scaled by γ with the MP-MP heuristic

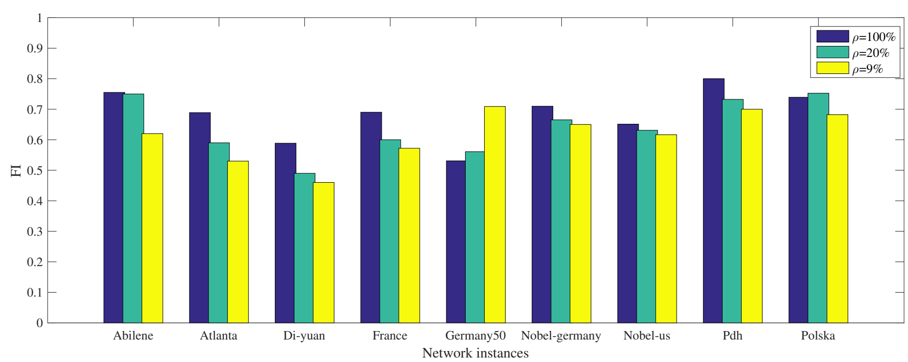


Fig. 15: Fairness index versus ρ using $TM1$ with the MP-MP heuristic

vol. 25, no 2.

[2] LANGE, Christoph, KOSIANKOWSKI, Dirk, WEIDMANN, Rainer, et al. "Energy consumption of telecommunication networks and related improvement options," IEEE Journal of selected topics in quantum electronics, 2011, vol. 17, no 2, pp. 285-295.

[3] MAALOUL, Rihab, CHAARI, Lamia, et COUSIN, Bernard. "Energy Saving Carrier-Grade Networks: A Survey," Computer Standards & Interfaces, 2017.

[4] RESTREPO, JC Cardona, GRUBER, Claus G., et MACHUCA, C. Mas. "Energy profile aware routing," In : Communications Workshops, 2009.

ICC Workshops 2009. IEEE International Conference on. IEEE, 2009. pp. 1-5.

[5] NUNES, Bruno Astuto A., MENDONCA, Marc, NGUYEN, Xuan-Nam, et al. "A survey of software-defined networking: Past, present, and future of programmable networks," IEEE Communications Surveys & Tutorials, 2014, vol. 16, no 3, pp. 1617-1634.

[6] DI MARTINO, Catello, MENDIRATTA, Veena, et THOTTAN, Marina. "Resiliency Challenges in Accelerating Carrier-Grade Networks with SDN," In : Dependable Systems and Networks Workshop, 2016, 46th Annual IEEE/IFIP International Conference on. IEEE, 2016. pp. 242-245.

[7] O. N. Foundation, "Software-defined networking: the new norm for networks (white paper)," 2012. [Online]. Available: https://www.opennetworking.org

[8] VAISHAMPAYAN, Chinmay, BIDKAR, Sarvesh, MEHTA, Saurabh, et al. "Demonstrating openflow over a carrier ethernet switch router (cesr)-a services perspective," In : Networks and Optical Communications (NOC), 2012 17th European Conference on. IEEE, 2012. pp. 1-5.

[9] BIDKAR, Sarvesh, GUMASTE, Ashwin, GHODASARA, Puneet, et al. "Field trial of a software defined network (SDN) using carrier ethernet and segment routing in a tier-1 provider," In : Global Communications Conference (GLOBECOM), 2014 IEEE. IEEE, 2014. pp. 2166-2172.

[10] AGARWAL, Sugam, KODIALAM, Murali, et LAKSHMAN, T. V. "Traffic engineering in software defined networks," In : INFOCOM, 2013 Proceedings IEEE. IEEE, 2013. pp. 2211-2219.

[11] KREUTZ, Diego, RAMOS, Fernando MV, VERISSIMO, Paulo Esteves, et al. "Software-defined networking: A comprehensive survey," Proceedings of the IEEE, 2015, vol. 103, no 1, pp. 14-76.

[12] STAESSENS, Dimitri, SHARMA, Sachin, COLLE, Didier, et al."Software defined networking: Meeting carrier grade requirements," In : Local & Metropolitan Area Networks (LANMAN), 2011, 18th IEEE Workshop on. IEEE, 2011. pp. 1-6.

[13] DABAGHI, Fahimeh, MOVAHEDI, Zeinab, et LANGAR, Rami. "A survey on green routing protocols using sleep-scheduling in wired networks," Journal of Network and Computer Applications, 2017, vol. 77, pp. 106-122.

[14] CIANFRANI, Antonio, ERAMO, Vincenzo, LISTANTI, Marco, et al. "An energy saving routing algorithm for a green OSPF protocol,". In : INFOCOM IEEE Conference on Computer Communications Workshops, 2010. IEEE, 2010. pp. 1-5.

[15] CIANFRANI, Antonio, ERAMO, Vincenzo, LISTANTI, Marco, et al. "An OSPF enhancement for energy saving in IP networks," In : Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on. IEEE, 2011. pp. 325-330.

[16] CUOMO, Francesca, ABBAGNALE, Anna, et PAPAGNA, Sabino. ESOL: "Energy saving in the Internet based on Occurrence of Links in routing paths,". In : World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2011 IEEE International Symposium on a. IEEE, 2011. pp. 1-6.

[17] CUOMO, Francesca, CIANFRANI, Antonio, POLVERINI, Marco, et al. "Network pruning for energy saving in the Internet," Computer Networks, 2012, vol. 56, no 10, pp. 2355-2367.

[18] MATSUURA, Hiroshi. "Energy-saving routing algorithm using Steiner tree, " In : Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on. IEEE, 2013. pp. 378-386.

[19] MAALOUL, Rihab, CHAARI, Lamia, et COUSIN, Bernard. "Energy-aware forwarding strategy for Metro Ethernet networks," In : Computer Systems and Applications (AICCSA), 2015 IEEE/ACS 12th International Conference of. IEEE, 2015. pp. 1-7.

[20] GIROIRE, Frederic, MAZAURIC, Dorian, MOULIERAC, Joanna, et al. "Minimizing routing energy consumption: from theoretical to practical results, " In : Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'l Conference on & Int'l Conference on Cyber, Physical and Social Computing (CPSCom). IEEE, 2010. pp. 252-259.

[21] CHIARAVIGLIO, Luca, MELLIA, Marco, et NERI, Fabio. "Minimizing ISP network energy cost: Formulation and solutions, " IEEE/ACM Transactions on Networking (TON), 2012, vol. 20, no 2, pp. 463-476.

[22] AMALDI, Edoardo, CAPONE, Antonio, et GIANOLI, Luca G. "Energy-aware IP traffic engineering with shortest path routing," Computer Networks, 2013, vol. 57, no 6, pp. 1503-1517.

[23] MOULIERAC, Joanna et PHAN, Truong Khoa. "Optimizing IGP link weights for energy-efficiency in multi-period traffic matrices," Computer Communications, 2015, vol. 61, pp. 79-89.

[24] CAPONE, Antonio, CORTI, Daniele, GIANOLI, Luca, et al. "An optimization framework for the energy management of carrier ethernet networks with multiple spanning trees, ". Computer Networks, 2012, vol. 56, no 17, pp. 3666-3681.

[25] HELLER, Brandon, SEETHARAMAN, Srinivasan, MAHADEVAN, Priya, et al. "ElasticTree: Saving Energy in Data Center Networks," In : Nsdi. 2010. pp. 249-264.

[26] BRUSCHI, Roberto, LOMBARDO, Alfio, MORABITO, Giacomo, et al. "Green extension of OpenFlow," In : Teletraffic Congress (ITC), 2014 26th International. IEEE, 2014. pp. 1-6.

[27] BOLLA, Raffaele, BRUSCHI, Roberto, DAVOLI, Franco, et al. "The green abstraction layer: A standard power-management interface for next-generation network devices," IEEE Internet Computing, 2013, vol. 17, no 2, pp. 82-86.

[28] BOLLA, Raffaele, BRUSCHI, Roberto, DAVOLI, Franco, et al. "Fine-grained energy-efficient consolidation in SDN networks and devices," IEEE Transactions on Network and Service Management, 2015, vol. 12, no 2, pp. 132-145.

[29] WANG, Rui, JIANG, Zhipeng, GAO, Suixiang, et al. "Energy-aware

routing algorithms in software-defined networks,". In : World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014 IEEE 15th International Symposium on a. IEEE, 2014. pp. 1-6.

[30] GIROIRE, Frederic, MOULIERAC, Joanna, et PHAN, Truong Khoa. "Optimizing rule placement in software-defined networks for energy-aware routing," In : Global Communications Conference (GLOBECOM), 2014 IEEE. IEEE, 2014. pp. 2523-2529.

[31] TUYSUZ, Mehmet Fatih, ANKARALI, Zekiye Kubra, et GOZUPEK, Didem. "A survey on energy efficiency in software defined networks," Computer Networks, 2017, vol. 113, pp. 188-204.

[32] KANG, Nanxi, LIU, Zhenming, REXFORD, Jennifer, et al. "Optimizing the one big switch abstraction in software-defined networks," In : Proceedings of the ninth ACM conference on Emerging networking experiments and technologies. ACM, 2013. pp. 13-24.

[33] KANIZO, Yossi, HAY, David, et KESLASSY, Isaac. Palette: "Distributing tables in software-defined networks, ". In : INFOCOM, 2013. pp. 545-549.

[34] RIFAI, Myriana, HUIN, Nicolas, CAILLOUET, Christelle, et al. "Too many SDN rules? Compress them with MINNIE, " In : Global Communications Conference (GLOBECOM), 2015 IEEE. IEEE, 2015. pp. 1-7.

[35] KANNAN, Kalapriya et BANERJEE, Subhasis. Compact TCAM: "Flow entry compaction in TCAM for power aware SDN," In : International Conference on Distributed Computing and Networking. Springer, Berlin, Heidelberg, 2013. pp. 439-444.

[36] AHUJA, Ravindra K., MAGNANTI, Thomas L., et ORLIN, James B. "Network flows: theory, algorithms, and applications," 1993.

[37] ADDIS, Bernardetta, CAPONE, Antonio, CARELLO, Giuliana, et al. "Energy management in communication networks: a journey through modeling and optimization glasses,". Computer Communications, 2016, vol. 91, pp. 76-94.

[38] GAREY, Michael R. et JOHNSON, David S. "A Guide to the Theory of NP-Completeness," WH Freemann, New York, 1979, vol. 70.

[39] JOHNSON, David S. "Fast algorithms for bin packing," Journal of Computer and System Sciences, 1974, vol. 8, no 3, pp. 272-314.

[40] XIA, Binzhou et TAN, Zhiyi. "Tighter bounds of the First Fit algorithm for the bin-packing problem," Discrete Applied Mathematics, 2010, vol. 158, no 15, pp. 1668-1675.

[41] DIJKSTRA, Edsger W. "A note on two problems in connexion with graphs," Numerische mathematik, 1959, vol. 1, no 1, pp. 269-271.

[42] MOHAR, Bojan, ALAVI, Y., CHARTRAND, G., et al. "The Laplacian spectrum of graphs," Graph theory, combinatorics, and applications, 1991, vol. 2, no 871-898, pp. 12.

[43] FIEDLER, Miroslav. Algebraic connectivity of graphs. Czechoslovak mathematical journal, 1973, vol. 23, no 2, pp. 298-305.

[44] JAIN, Raj. "The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling," John Wiley & Sons, 1990.

[45] IBM ILOG. IBM ILOG CPLEX 12.4 user's Manual; 2012

[46] [Online]. Available: https://http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/index.html

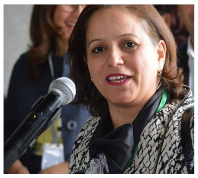
[47] GUNLUK, Oktay. "A branch-and-cut algorithm for capacitated network design problems," Mathematical Programming, 1999, vol. 86, no 1, pp. 17-39.

[48] ORLOWSKI, Sebastian, WESSALY, Roland, PIORO, Michal, et al. "SNDlib 1.0–Survivable network design library," Networks, 2010, vol. 55, no 3, pp. 276-286.

[49] TELKAMP, Thomas. "Best Practices for Determining the Traffic Matrix in IP Networks," V 2.0.

Rihab Maaloul is currently working toward the Ph.D. degree in computer systems engineering in University of Sfax, National Engineering School of Sfax. She is also a researcher at Technology and Smart Systems Laboratory (LT2S) of Digital Research Center of SFAX (CRNS). Her current research interests are energy aware routing in Carrier-Grade Ethernet networks and communication network optimization.

Raouia Taktak was born in Sfax, Tunisia, in 1983. She received the engineering degree from Tunisia Polytechnic School, Tunisia, in 2007, and the M.Sc and Ph.D. degrees in Operations Research and Computer Science (combinatorial optimization) from Paris- Dauphine University, France, in 2008 and 2013, respectively. Since 2014, she is an assistant professor of Computer Science in the Higher Institute of Computer Science and Multimedia, Sfax University, Tunisia. Her research interests include mixed-integer linear problems and methods, combinatorial optimization and polyhedra, exact and heuristic approaches of resolution.



Lamia Chaari was born in Sfax, Tunisia, in 1972. She received the engineering and Ph.D. degrees in electrical and electronic engineering from SFAX National Engineering School (ENIS) in TUNISIA. She received in 2011 HDR telecommunications from ENIS. Actually she is an associate professor at multimedia and informatics higher institute, SFAX University, TUNISIA. She is also a researcher at electronic and technology information laboratory (LETI). Her scope of research are new generation networks, communications, wireless networking, multiple access and MAC design, QoS provisionning, IoT, M2M, SDN, ICN. She is the author or co-author of many journal papers and conferences papers in these areas.

Bernard Cousin since 1992, is a Professor of Computer Science at the University of Rennes 1, in France. Bernard Cousin received, in 1987, his PhD degree in computer science from the University of Paris 6. Between 1985 and 1992, he has been successively Assistant Professor at University of Paris 6 and Associate Professor at University of Bordeaux 1. Currently, he is at the head of a research group on Advanced Networking. He is, member of IRISA (a CNRS-University joint research laboratory located at Rennes in France). He has co-authored 150+ scientific papers published in international journals and conferences. His research interests include next generation Internet, green networking, all-optical networks, wireless networks, dependable networking, high speed networks, traffic engineering, multicast routing, network QoS management, network security and multimedia distributed applications.