2010 IEEE/ACM International Conference on Green Computing and Communications & 2010 IEEE/ACM International Conference on Cyber, Physical and Social Computing

Minimizing Routing Energy Consumption: from Theoretical to Practical Results[‡]

Frédéric Giroire*, Dorian Mazauric*, Joanna Moulierac* and Brice Onfroy*
*MASCOTTE, INRIA, I3S (CNRS / Université de Nice-Sophia Antipolis), France
Mail: *first.lastname*@sophia.inria.fr

***Abstract*—Several studies exhibit that the traffic load of the routers only has a small influence on their energy consumption. Hence, the power consumption in networks is strongly related to the number of active network elements, such as interfaces, line cards, base chassis,... The goal thus is to find a routing that minimizes the (weighted) number of active network elements used when routing. In this paper, we consider a simplified architecture where a connection between two routers is represented as a link joining two network interfaces. When a connection is not used, both network interfaces can be turned off. Therefore, in order to reduce power consumption, the goal is to find the routing that minimizes the number of used links while satisfying all the demands. We first define formally the problem and we model it as an integer linear program. Then, we prove that this problem is not in APX, that is there is no polynomial-time constant-factor approximation algorithm. Thus, we propose a heuristic algorithm for this problem and we present a study on specific topologies, such as trees and complete graphs, that provide bounds and results useful for real topologies. We then exhibit the gain in terms of number of network interfaces for a set of existing network topologies: we see that for almost all topologies more than one third of the network interfaces can be spared for usual ranges of operation, leading to a global reduction of approximately 33 MWh for a medium-sized backbone network. Finally, we discuss the impact of energy efficient routing on the stretch factor and on fault tolerance.**

I. Introduction

The minimization of ICT energy consumption has become a priority with the recent increase of energy cost and the new sensibility of the public, governments and corporations towards energy consumption. ICT alone is responsible of 2% to 10% (depending on the estimations) of the world consumption [2]. In this paper, we are interested in the networking part of this energy consumption, and in particular in the routing. It is estimated that switches, hubs, routers account for 6 TWh per year in the US [16].

Some recent studies [1], [14] exhibit that the traffic load of the routers only has a small influence on their energy consumption. Hence, the dominating factor is the number of switched-on network elements: interfaces, platforms, routers,... In order to minimize energy, we should try to use as few network elements as possible.

Nevertheless, in most of networks, PoPs or even routers cannot be turned off. As a matter of fact, first, they are the source or destination of demands; second, they can be part of backup routes to protect the network against failures. For this reason, we consider in this paper a simplified architecture where a connection between two routers is represented by a link joining two network interfaces. We can spare energy by turning off the two network interfaces which are the extremities of the link. The network is represented by an undirected graph and, in that case, the goal in this simplified architecture is *to find a subgraph minimum in number of links to route the demands*. The contributions of this paper are the following:

- We prove that there is no polynomial-time constant-factor approximation algorithm for this problem, even for two demands or if all links have the same capacity.
- We give explicit *close formulas or bounds for specific topologies*, such as trees and complete graphs. They provide limit behaviors and give indications of how the problem behaves for general networks. To the best of our knowledge, we are presenting in this paper the first study of energy-efficient routing solutions on specific topologies.
- We present *heuristics* to find close to optimal solutions for general networks. These heuristics are validated by comparison with theoretical bounds for specific instances of the problem.
- We study the *energy gain* on a set of topologies of existing backbone networks. We exhibit that *at least one third of the network interfaces* can be spared for usual range of demands.
- Finally, we discuss the impact of energy-efficient routing on *route length* and *fault tolerance*.

A. Related Work

Measure of energy consumptions. Several measurement campaigns of network energy consumption have been carried out in the last few years. See for example [13], [14] and [1]. Their authors claim that the consumption of network devices is largely independent of their load. In particular, in [1], the authors were interested by routers' energy consumption. They observe that for the popular Cisco 12000 series, the consumption at a load of 75% is only 2% more than at an idle state (770W vs. 755W). In [14], the authors show through experimentation that the power consumed depends on the number of active ports. Explicitly disabling unused ports on a line card reduces the device power consumption. The values

‡ A long version of this work with all the proofs and additive work has been reviewed for this conference and can be found in [9].

978-0-7695-4331-4/10 $26.00 © 2010 IEEE
DOI 10.1109/GreenCom-CPSCom.2010.147

obtained during experimentation show that the consumption of a linecard 4-port Gigabit ethernet (100W) is approximately one fourth the consumption of the global base system (430W).

Energy minimization. In [1], the authors model this problem as an integer linear program. The objective function is a weighted sum of the number of platforms and interfaces. They show how much energy can be saved on different networks with this model. However, they do not give intuitions, explanations, nor formulas for their results. In [12], the authors propose a rerouting at different layers in IP-over-WDM networks for energy savings while [18] study the impact of the technology for energy efficient routing. In [15], [11], [3], researchers proposed techniques such as putting idle subcomponents (line cards, ports, etc.) to sleep, as well as adapting the rate for forwarding packets depending on the traffic in local area networks. In [4], the authors propose a modulation of the radio configurations in fixed broadband wireless networks to reduce the power consumption.

The remainder of the paper is organized as follows. In Section II, we first present formally the problem and we model it as an integer linear program. In Section III, we recall the complexity of related problems and we prove that the problem cannot be approximated within a constant factor. In Section IV, we design heuristic algorithms and we prove also negative results about greedy and probabilistic heuristic algorithms. Then, specific topologies, such as trees or complete graphs, are discussed in Section V. We show the good performance of our proposed heuristic in terms of power consumption in Section VI, and we show the impact of such energy-efficient solutions on route lengths and network fault tolerance. Finally, we discussed the impact of such algorithms for network operators in Section VII.

II. Problem Modeling

One way to reduce the network power consumption consists in minimizing the number of turned-on network equipments. We model a network topology as an undirected weighted graph $G = (V, E)$, where the weight c_e represents the capacity of edge $e \in E$. We represent the set of demands by $D = \{\mathcal{D}_{st} > 0; (s,t) \in V \times V, s \neq t\}$, where $\mathcal{D}_{st}$ denotes the amount of demand from s to t. A demand $\mathcal{D}_{st}$ has to be routed through an elementary path from node $s \in V$ to $t \in V$. A valid routing of the demands is an assignment of such a path in G for each $\mathcal{D}_{st} \in D$ such that for each edge $e \in E$, the total amount of demands through e does not exceed the capacity c_e. A classical decision problem is to determine if there is such a routing of the demands in G. Formally:

Definition II.1. *Given an undirected weighted graph $G = (V, E)$ and a set of demands D, the* Routing Problem *consists in deciding if there is a valid routing of the demands D in G.*

For our purpose, we study a simplified network architecture in which a link (A, B) connects two routers A and B through 2 network interfaces, one for A and one for B. The degree of a node in the graph corresponds to the number of network interfaces on the router. If a network interface is turned-off, then the other interface at the extremity of the link is not useful anymore and can also be turned-off. Therefore, the objective of our problem is to minimize the number of active links in the network. Formally:

Definition II.2. *Given an undirected weighted graph $G = (V, E)$ and a set of demands D, the* Minimum Edges Routing Problem *consists in finding a minimum cardinality subset $E^* \subseteq E$ such that there is a valid routing of the demands D in $G^* = (V, E^*)$.*

In Section II-A, we present some simple instances of the Minimum Edges Routing Problem and the corresponding solutions. In Section II-B, we describe a linear program for our problem.

A. Examples

Consider the graph $G = (V, E)$ depicted in Figure 1 composed of 14 nodes and 15 edges. Integers on edges represent the different capacities.

In Figure 1(a), there are two demands $D_{s_1 t_1} = 10$ and $D_{s_2 t_2} = 5$. The solution E^* for the Minimum Edges Routing Problem is composed of $|E^*| = 7$ edges. Note that the path from s_i to t_i in $G^* = (V, E^*)$ is composed of 5 edges, whereas the shortest path in G is composed of 4 edges, for $i \in \{1, 2\}$. Indeed these two shortest paths are edge-disjoint whereas, in the optimal routing, the two paths share 3 edges. This simple example shows that the shortest paths routing does not give the optimal solution.

In Figure 1(b), there are also two demands but the amount of demand from s_1 to t_1 is now equal to 12. This increase considerably changes the optimal solution E^*. Indeed the

$G = (V, E)$	Network topology with V the set of vertices (or routers) and E the set of edges (or links).
$\mathcal{D}_{st}$	Volume of traffic of the demand from a source $s \in V$ to a destination $t \in V$. In section V, $\forall (s,t) \in V \times V, \mathcal{D}_{st} = \kappa$.
c_e	Capacity of the edge $e \in E$. In section V, $\forall e \in E, c_e = c$.
λ	Capacity/Demand ratio with $c = \lambda\kappa$.
r_e	Residual capacity of edge e.
OF	Network Overprovisionning Factor. $OF = 1$ means that the capacities c_e of the edges imply the feasibility of the routing of the demands.
f_{uv}^{st}	Flow on edge $e = uv$ corresponding to the demand $\mathcal{D}_{st}$.
x_e	Binary variable which says if edge e is used or not.
$\bar{l}_H(D)$	Average path length in the graph H given by a feasible routing of the demands in D.
$DP_H(D)$	Average disjoint paths in the graph H for the demands in D.
δ	Degree of a vertice and δ_{max} is the maximum degree of the nodes in the graph.
$d_G(i,j)$	Shortest path distance between node i and j in graph G. The notation $d(i,j)$ can be used for shortcut.

Table I
Summary of Notations

demand $D_{s_1t_1}$ must be routed through the shortest path composed of 4 edges because of the edge of capacity 11. Thus E^* is composed of the two previous edge-disjoint shortest paths of length 4. We get $|E^*| = 8$.

In Figure 1(c) and 1(d), we have the demands of the second example plus $D_{s_3t_3} = 2$. Because of the three edges of capacity 16, only two demands can share these 3 edges. Two optimal solutions are depicted in these figures, each one of cost $|E^*| = 9$. In these optimal solutions, the 3 edges of capacity 16 support the demand $D_{s_3t_3}$ and one of the two demands $D_{s_2t_2}$ (for Figure 1(c)) or $D_{s_1t_1}$ (for Figure 1(d)). The other demand is routed through the shortest path.

B. Integer Linear Program

The Minimum Edges Routing Problem can be modeled as a multicommodity *integral* flow problem in which the objective function is the minimization of the number of edges. We note f_{uv}^{st} the flow on edge uv corresponding to the demand $\mathcal{D}_{st}$ flowing from u to v. We note $f_{uv} = \sum_{st \in V \times V} f_{uv}^{st}$. For each edge $e \in E$, we introduce a binary variable x_e which says if the edge e is used or not: $x_e = 0$ if $f_{uv} + f_{vu} = 0$ and $x_e = 1$ if $f_{uv} + f_{vu} > 0$.

The *Objective function* is then $\min \sum_{e \in E} x_e$ subject to:

Flow constraints: $\forall (s,t) \in V \times V$, $\forall u \in V$,

$$\sum_{v \in N(u)} f_{vu}^{st} - \sum_{v \in N(u)} f_{uv}^{st} = \begin{cases} -\mathcal{D}_{st} & \text{if} \quad u = s, \\ \mathcal{D}_{st} & \text{if} \quad u = t, \\ 0 & \text{otherwise.} \end{cases}$$

Capacity constraints: $\forall e = (u,v) \in E$,

$$\sum_{d \in \mathcal{D}} \left(f_{uv}^d + f_{vu}^d \right) \leq x_e c_e.$$

The flow constraints are usual flow conservation. The capacity constraints state that for each edge $e \in E$, the total amount of demands through e does not exceed the capacity c_e.

Table I summarizes the notations used throughout the paper.

III. Impossibility of Approximation

The Minimum Edges Routing Problem presented in this paper is a special case of different well known network optimization problems: minimum cost routing [20], minimum concave flow problem [10], minimum flow problem with step cost functions [8]. In operation research, this problem can be seen as a special case of the Fixed charge transportation problem [19], [5], where the cost of the flow unit on an edge is zero. Note that this problem is NP-Hard [10]: the number of possible subgraphs to test is strongly exponential for most graphs. Moreover, even for a given subgraph (when the set of edges to be used is fixed), the feasibility of a multicommodity integral flow problem has to be assessed. This simpler problem corresponds to the Routing Problem and it is known to be NP-complete even for two commodities [7]. Last, note that it is also the worst case of step-functions as most of the approximations by linearization are very far from a feasible solution.

We prove in [9] that the Minimum Edges Routing Problem is not in APX (and so it is an NP-hard problem) even for two special kinds of instances (Theorem III.1 and Theorem III.2). It means that there is no polynomial-time constant-factor approximation algorithms for the Minimum Edges Routing Problem, unless $P = NP$.

Theorem III.1. *The* Minimum Edges Routing Problem *is not in APX even for two commodities.*

Proof: This proof and all the following ones can be found in [9]. ■

Theorem III.2. *The* Minimum Edges Routing Problem *is not in APX even if each edge has a constant capacity c.*

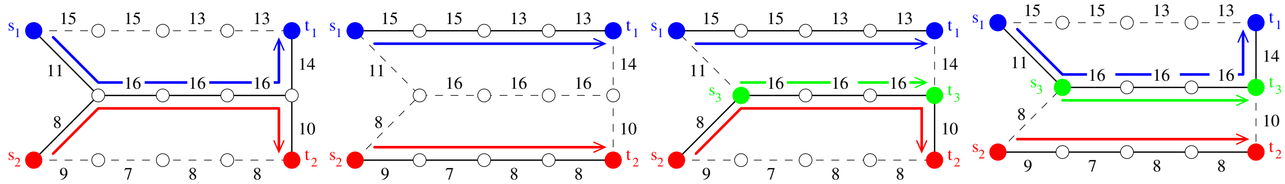
These two negative results motivate the design of heuristic algorithms for the Minimum Edges Routing Problem in Section IV and the study of theoretical bounds for particular instances in Section V that give information for general networks.

IV. Heuristics

As we have seen in the previous section, the Minimum Edges Routing Problem is a problem difficult to solve exactly and is even difficult to approximate to an insured factor in general. Hence, the necessity of proposing good heuristic algorithms for classical real network topologies. We propose in this section two heuristics to find energy-efficient routing, namely Less Loaded Edge Heuristic and Random Heuristic. These heuristics are tested in Section VI.

Algorithm 1 presents a simple heuristic named Less Loaded Edge Heuristic for our problem. We start from the whole network, compute a feasible routing as described in Algorithm 2 and try to remove in priority edges that are less loaded. We believe that it is better to remove these edges that are not involved in many shortest paths than overloaded edges. For the routing, the demands are considered one by one in random order. We compute a shortest path for the demand with the metric $\frac{c_e}{r_e}$ on edges is computed, where r_e is the residual capacity on edge e when the previous demands have been routed. Then, the residual capacity is updated for each edge and the next demand is considered. This routing allows a better load balancing of the demands in the network. Note that finding a feasible routing can also be done with an integer linear program for the Routing Problem for small topologies. Each time an edge is removed in the network, a feasible routing is computed. If no routing exists, then the removed edge is put back and we try to remove another edge that has not been yet considered. The process of removing less loaded edges is done until no more edges can be removed.

The second heuristic, Random Heuristic, is used as a measure of comparison during the simulations. The only difference with the first one is that it selects uniformly at random the links to be removed and not (necessarily) the less



(a) $D_{s_1t_1} = 10$, $D_{s_2t_2} = 5 \rightarrow$ the two routes of the optimal solution are not shortest paths.

(b) $D_{s_1t_1} = 12$, $D_{s_2t_2} = 5 \rightarrow$ the optimal solution corresponds to a shortest paths routing.

(c) $D_{s_1t_1} = 10$, $D_{s_2t_2} = 5$, $D_{s_3t_3} = 2$. $D_{s_1t_1}$ and $D_{s_3t_3}$ are routed through shortest paths.

(d) $D_{s_1t_1} = 10$, $D_{s_2t_2} = 5$, $D_{s_3t_3} = 2$. $D_{s_2t_2}$ and $D_{s_3t_3}$ are routed through shortest paths.

Figure 1. Four different solutions for the MINIMUM EDGES ROUTING PROBLEM.

loaded edges. The routing is performed in the same way as for LESS LOADED EDGE HEURISTIC.

Algorithm 1 LESS LOADED EDGE HEURISTIC

Require: An undirected weighted graph $G = (V, E)$ where each edge $e \in E$ has an initial capacity c_e and a residual capacity r_e (depending on the demands supported on e). A set of demands $\mathcal{D}$, each demand has a volume of traffic $\mathcal{D}_{st}$.

$\forall e \in E, r_e = c_e$
Compute a feasible routing of the demands with Algorithm 2
while Edges can be removed **do**
  Remove the edge e' that has not been chosen once, with the smallest value $\frac{c(e')}{r(e')}$.
  Compute a feasible routing with Algorithm 2
  If no feasible routing exists, then put back e' in G
end while
return the subgraph G.

Algorithm 2 Feasible routing Heuristic for ROUTING PROBLEM

Require: An undirected weighted graph $G = (V, E)$ where each edge $e \in E$ has an initial capacity c_e and a residual capacity r_e (depending on the demands supported on e). A set of demands $\mathcal{D}$, each demand has a volume of traffic $\mathcal{D}_{st}$.

Sort the demands in random order
while $\mathcal{D}_{st}$ is a demand in D with no routing assigned **do**
  Compute a shortest path SP_{st} with the metric $\frac{c_e}{r_e}$ on edges
  Assign the routing SP_{st} to the demand $\mathcal{D}_{s,t}$
  $\forall e \in SP_{st}, r_e = c_e - \mathcal{D}_{st}$
end while
return the routing (if it exists) assigned to the demands in D.

These heuristics are evaluated through simulations in Section VI and compared to the theoretical bounds given in Section V and to the integer linear program described in Section II-B

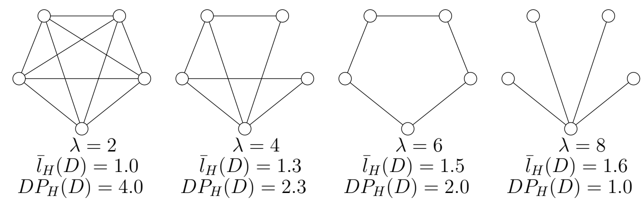


Figure 2. A Toy example, study of the complete graph with 5 vertices: subgraphs with the minimum number of edges, with λ the capacity/demand ratio, $\bar{l}_H(D)$ average route length and $DP_H(D)$ the average number of edge-disjoint paths between two nodes where D contains the all-to-all demands.

V. TOPOLOGY STUDY: EXTREME CASES

We present here the general framework of the studies of the rest of this paper. We then study two extreme cases for general networks, namely trees and complete graphs. They give us the limit behavior of the real networks for different network loads. We also derive an upper bound on the number of links that can be spared in function of the demands.

A. Our Framework

In general networks, the demands vary during the life of the networks, e.g. with the increase of the number of users, with the development of new technologies, or, more simply, according to the time of the day. The goal of the study in this section is to see how much energy can be spared depending on the resources available for the routing.

Given an undirected weighted graph $G = (V, E)$ representing a network, each edge $e \in E$ has the same amount of capacity $c_e = c$. We perform an all-to-all routing with demands of volume, $\forall (s, t) \in V \times V, \mathcal{D}_{st} = \kappa$.

Definition V.1. *The capacity/demand ratio λ expresses the relation between the demand and the edge capacity: $\lambda = c/\kappa$.*

Although an all-to-all routing is not a realistic scenario, it allows to study the effects of such routing in extreme conditions and also to keep the network connected. With such scenario, we can certify that the topology given by our algorithm will be suitable for more realistic traffic matrices. Then, for each topology, we look at the number of links that can be spared for different capacity/demand ratios λ.

A toy example. As an illustration, Figure 2 shows an optimal solution H given by the integer linear program described in

Section II-B for the complete graph G composed of 5 nodes when the capacity/demand ratio λ varies from 2 to 8. When λ equals 2, all the edges of G are needed to perform an all-to-all routing with $\kappa = 1$. The larger the ratio is, the fewer network interfaces are needed until we reach the star graph where no more network interfaces can be removed. The two extreme cases are the complete graph ($\lambda = 2$) and the tree ($\lambda = 8$). The gain in terms of network interfaces can reach 60%, indeed, only 4 edges (or 8 network interfaces) are needed instead of 10 (or 20). We also measure the impact of this energy-efficient routing on delay and failure protection: for each different λ, we give the *average route length*, $\bar{l}_H(D)$, given by a feasible routing of the all-to-all demands on the solution subgraph H, and the *average number of edge-disjoint paths* linking two nodes, $DP_H(D)$. We see that, for this simple example, the route length increases by 60% and that the number of disjoint paths drops from 4 to 1 between the two extreme cases.

Note that an orthogonal way to conduct the study is to fix the number of network interfaces/edges to be turned-on and to compute the *load of the network*, that is the minimum capacity needed to be able to satisfy the all-to-all routing:

Definition V.2 (Load of a Graph)**.** *Let $G = (V, E)$ be an undirected weighted graph and D the set of demands. The load of G is the minimum over all routings (feasible flows $\mathcal{F}$) of the maximum load over all edges:*

$$\min_{f \in \mathcal{F}} \max_{e \in E} f_e.$$

B. General Bounds

Path length lower bounds. The global capacity of the system has to be larger than the global demand. The global flow is minimum when all demands are following the shortest paths between their source and destination. We note $d(s,t)$ the length of a shortest path between a source s and a destination t. Hence, we have

$$\sum_{e \in E} c_e \geq \sum_{st \in V^2; s \neq t} d(s,t) \mathcal{D}_{st}.$$

In particular, when all the edges have the same capacity c and all the couples of nodes have the same demand κ, it becomes

$$c|E| \geq \kappa \sum_{st \in V^2; s \neq t} d(s,t).$$

or equivalently

$$|E| \geq \frac{1}{\lambda} \sum_{st \in V^2; s \neq t} d(s,t).$$

Max flow min cut lower bounds. We present here a generalized max flow min cut argument. For each subset $S \subseteq V$, we must have

$$\sum_{e=uv \in E; u \in S, v \in \bar{S}} c_e \geq \sum_{s \in S, t \in \bar{S}} \mathcal{D}_{st} + \sum_{s \in S, t \in \bar{S}} \mathcal{D}_{ts},$$

where $\bar{S} = V \setminus S$. In particular, when all the edges have the same capacity c and all the couples of nodes have the same demand κ, it becomes

$$c|E_{S\bar{S}}| \geq 2\kappa |S||\bar{S}|,$$

where $|E_{S\bar{S}}|$ the number of edges of the cut between S and $\bar{S}$. The load of a graph can thus be computed by looking at the *minimum cut* of the network that supports the maximum flow.

Minimum bisection cut. A particular example of cut is the minimum *bisection cut* denoted $E'_{S\bar{S}}$. The minimum bisection cut is the minimum cut that divides the network into two (almost) equal-sized regions S and $\bar{S}$: $|S| = \lceil \frac{n}{2} \rceil$ and $|\bar{S}| = \lfloor \frac{n}{2} \rfloor$. The set of edges corresponding to the minimum bisection cut will support the demands exchanged between the nodes of the two regions. This gives a minimum value for the load of the graph which is for most practical networks a good approximation of the real load as shown in Section VI.

In a graph with n vertices, with minimum bisection cut $E'_{S\bar{S}}$, the load of the graph in case of all-to-all demand is at least:

$$\frac{2\kappa}{|E'_{S\bar{S}}|} \lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor.$$

C. Load of the Minimal Subgraph: a Spanning Tree

In the undirected case, the subgraph with the minimum number of edges is a tree, as it is the smallest connected subgraph, see e.g. Figure 2. This minimal configuration can be attained when the capacity is larger than the load given in Lemma V.1.

Lemma V.1. *[Tree and Spanning Tree]*

a) *The load of a tree composed of n nodes is $2\kappa v(n-v)$, where v is the size of the larger branch incident to the tree centroid.*

b) *In a graph with n nodes and of maximum degree δ_{max}, the load of a spanning tree is at least*

$$2\kappa \left\lceil \frac{n-1}{\delta_{max}} \right\rceil \left(n - \left\lceil \frac{n-1}{\delta_{max}} \right\rceil \right).$$

Note that the load of a (spanning) tree mostly depends on the maximum node degree of the underlying graph. On a complete graph where $\delta_{\max} = n-1$, the tree with the lowest load is a star and its load is $2\kappa(n-1)$, to be compared with the load of a path $\kappa \lfloor \frac{n^2}{2} \rfloor$, for which $\delta_{\max} = 2$. Hence, networks with nodes of large degrees tend to attain the minimum configuration for smallest capacities, see Section VI.

D. Complete Graph - Bound on the Number of Spared Edges

We consider here the complete graph K_n composed of n nodes and $n(n-1)/2$ edges. This topology is the *second extreme case* (all possible edges) of our problem. It corresponds to the *design problem* of finding the best network satisfying the load when all possible edges between nodes can be used.

The all-to-all routing is possible on the complete graph as soon as the capacity/demand ratio λ is larger than 2. We also

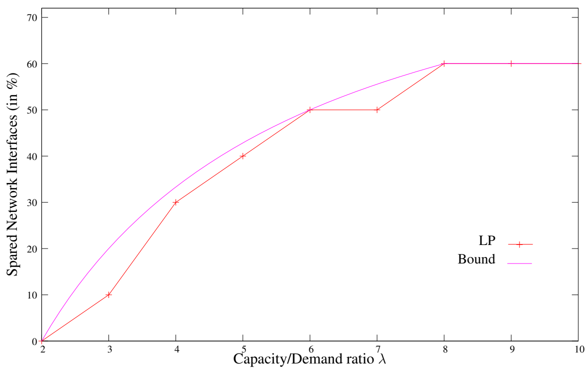


Figure 3. Number of spared edges on the complete graph with 5 nodes. Bound of Lemma V.2 and integer linear program described in Section II-B.

have seen in the previous section that when λ is larger than $2(n-1)$ the routing is possible on the star with only $n-1$ edges, corresponding to the small fraction $2/n$ of the total number of edges of the complete graph. But what happens between these two capacities?

Lemma V.2. *In a complete graph with n vertices, the all-to-all routing uses at least*

$$\max\left(\frac{2\kappa n(n-1)}{c+2}, n-1\right)$$

edges, with c the capacity of the edges.

We validate in Figure 3 the results of the integer linear program described in Section II-B given by CPLEX 10 for the complete graph with 5 nodes. The figure shows that the lower bound given in Lemma V.2 is close to the optimal solution. The capacity/demand ratio λ varies between 2 and 8 as stated before. For $\lambda = 4$, a gain of 30% of networks interfaces is attained, leading to 7 active links instead of 10.

VI. Results on General Networks

We present in this section the results of our proposed heuristics on general networks. We study ten classical network topologies extracted from SNDLib (http://sndlib.zib.de). In our experiments, we explore how many network interfaces can be spared for different ranges of overprovisioning factor. We consider a range of capacity/demand ratio λ starting from the smallest value λ_1 allowing to route all the demands (overprovisioning factor equals 1) to the value λ_{tree} allowing to route on a minimal subgraph, that is a spanning tree (overprovisioning factor equals $\frac{\lambda_1}{\lambda_{\text{tree}}}$). We also study the impact of this energy-efficient routing on the *route lengths* and on the *network fault tolerance*. To this end, we propose in [9] an integer linear program finding *spanners* of the topology having good stretch and two disjoint paths between all pairs of nodes. We compare the number of edges of these spanners with the one of the minimum subgraphs.

A. SNDLib Topologies

We studied ten classical real network topologies extracted from SNDLib (http://sndlib.zib.de). These networks correspond to US (*Atlanta*), European (*Nobel EU*, *Cost266*), or single country (*Nobel Germany, France*) topologies. Their sizes span from 15 to 54 nodes and from 22 to 89 edges, as summarized in Table II. For these 10 topologies, we computed energy efficient routings for different capacity/demand ratios λ. We could only run the integer linear program for the smallest network, *Atlanta*, for which the results are presented in Figure 4. We see that Less Loaded Edge Heuristic performs well, attaining the optimal value most of the time, when the *Random* heuristic needs a larger λ to attain the same value. As a matter of fact CPLEX already takes several hours on *Atlanta* to solve the problem for one capacity/demand ratio. We thus present the results found by the heuristic (which takes only tens of ms) in Tables II and III.

Spared network interfaces. We give the percentage of spared network interfaces in function of the overprovisioning factor (OF) in Table II (a). A factor of 1 means that we use the minimum capacity/demand ratio necessary to route all the demands (corresponding to the value λ_1), when, e.g., a factor of 2 means that we have twice the value λ_1. Note that in most today's backbone networks overprovisioning is heavily used as it is an efficient and easy way to provide protection against failure: links are often used between 30 and 50 % of their capacity.

First, note that on some of the ten topologies, as soon as the routing is feasible ($OF = 1$), some network interfaces can already be turned off (12% for *Norway* and *Nobel EU*). As a matter of fact, in this case, the important edges of the network (the edges in the minimum cut for example) are fully used, but at the same time edges at the periphery are less used and some can be spared. With an overprovisioning factor of two, around one third of the edges can be spared (and even 53% for the *Pioro40* network). With larger factors (3 or 4), the gain is not as important, but still some network interfaces can be saved (e.g. 36% for *Atlanta*). We show in Table II (a), in the 2 last columns, the value of OF for which the tree is attained together with the corresponding spared network interfaces in percentage (SNE). The values are directly linked to the density of the network. For example, *Norway* needs a factor of 4.71 ($\lambda_{\text{tree}} = 4.71 \times \lambda_1$) to reach the tree with 26 links (instead of 51), sparing 49% edges. Hence, the larger the density, the more the network interfaces that can be turned off.

To conclude, for all the studied networks *between one third and one half of the network interfaces can be spared for usual overprovisioning factors*. Furthermore, when the best routing cannot be found by the integer linear program (large topologies), the *proposed heuristic found close to optimal solutions*.

Limit Configurations.

a) Full Network Topology: We report in Table II (b) the minimum capacity/demand ratio λ_1 for which the heuristic can perform an all-to-all routing. As explained in Section V-B, λ_1 depends on the minimum cut of the network. We reported

	$\lvert V\rvert$	$\lvert E\rvert$	Overprovisioning factor 1	2	3	4	Tree OF	%SNE
Atlanta	15	22	0%	32%	36%	36%	2.66	36%
New York	16	49	2.0%	59%	63%	67%	5.2	69%
Nobel Germany	17	26	0%	35%	39%	39%	2.75	39%
France	25	45	0%	42%	44%	47%	3.13	47%
Norway	27	51	12%	43%	47%	47%	4.71	49%
Nobel EU	28	41	12%	32%	34%	34%	2.76	34%
Cost266	37	57	3.5%	32%	35%	37%	3.68	37%
Giul39	39	86	0%	45%	50%	52%	8.25	56%
Pioro40	40	89	0%	53%	54%	55%	5.12	56%
Zib54	54	80	0%	30%	33%	33%	4.71	34%

(a) Gain of network interfaces (in %).

	Simulations λ_1	λ_{tree}	Values given by the bounds λ_1 Cut	Bound	λ_{tree} $\delta_{\max}$	Bound
Atlanta	38	101	3	38	4	88
New York	15	78	12	11	11	56
Nobel Germany	44	121	4	36	5	104
France	67	210	7	45	10	132
Norway	75	354	6	61	6	220
Nobel EU	131	362	3	131	5	264
Cost266	175	644	4	171	4	540
Giul39	85	702	11	70	8	340
Pioro40	153	784	7	115	5	512
Zib54	294	1385	6	243	10	576

(b) Evaluation of the load given by the bounds.

Table II

(a) Gain of network interfaces (in %) depending on the overprovisioning factor and (b) Evaluation of the load given by the bounds in previous section with λ_1 the capacity/demand ratio for overprovisioning factor equals 1 and λ_{tree} for the tree.

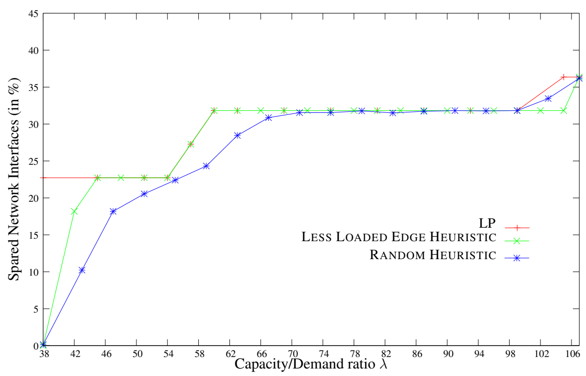


Figure 4. Percentage of spared network interfaces for the *Atlanta* network.

	Overprovisioning 1	2	3	λ_{tree}	Overprovisionning 1	2	3
Atlanta	1.00	1.19	1.25	1.25	2.35	1.09	1.00
New York	1.01	1.24	1.26	1.32	4.90	1.24	1.19
Nobel Germany	1.00	1.11	1.18	1.18	2.35	1.04	1.00
France	1.00	1.10	1.12	1.16	2.48	1.02	1.01
Norway	1.02	1.17	1.18	1.25	2.61	1.14	1.04
Nobel EU	1.08	1.14	1.24	1.25	1.82	1.07	1.00
Cost266	1.04	1.11	1.19	1.32	2.47	1.12	1.07
Giul39	1.00	1.18	1.21	1.50	3.68	1.41	1.14
Pioro40	1.00	1.25	1.32	1.42	4.06	1.12	1.09
Zib54	1.00	1.02	1.07	1.11	2.16	1.05	1.01

(a) Route length / **(b) Fault tolerance**

Table III

Impact of the energy-efficient routing on (a) the route length (average multiplicative stretch factor) and on (b) the network fault tolerance (average number of disjoint paths).

for each network the minimum bisection cut dividing the network into two (almost) equal-sized regions. We computed this minimum bisection cut with an integer linear program. The bound on λ_1 implied by the cut is given in the second column of the table. We see that it gives a very good indication of λ_1 for most of the networks, even if the value is not tight, as the heuristic does not always find the optimal solution. For example, we see that the bound is tight for *Atlanta*, where the minimum cut is of size 3 and splits the network in two sub-networks of sizes 8 and 7. For *Atlanta*, and *Nobel EU*, the capacity/demand ratios evaluated by the minimum cut bound is equal to the values given during the simulations ($\lambda_1 = 38$ for *Atlanta* and 131 for *Nobel EU*).

b) Spanning Trees: As explained in Section V (Lemma V.1), the ratio λ_{tree}, for which we get a spanning tree, depends on the network node degree. On the contrary to regular network, such as the complete graph, the existence of a spanning tree centered on the node of maximum degree with equal-sized branch is not given. Hence, the bounds given in Table II (b) are not tight. Nevertheless, we see that the node degree still is a good indication on what can be achieved on these networks. Graphs with low maximum degree attain the best configuration for lower values of capacity/demand ratio λ_{tree}. For example, the network *Atlanta* has a maximum node degree of 4 (the bound is 88) and λ_{tree} is 101, when for *Zib54* λ_{tree} is 1385 for a maximum degree of 10 (the bound is 576).

B. Impact on the Network

We believe that network operators will implement energy efficient routing only if the impact on other parameters is limited. We discuss here the impact on the route lengths and on the fault tolerance of our proposed heuristic.

Impact on the route lengths. When turning-off some components in a network, we save some energy but, at the same time, we route on longer paths. The *multiplicative stretch* is defined as the ratio between the average route length in the new routing divided by the average route length with the old routing (using all the edges). Results for the SNDLib topologies are given in Table III (a). A stretch of one correspond to the cases where no edge could be spared and thus the routing is not affected. We see that, as expected, the general trend is that when the overprovisioning factor increases, the paths become longer.

Nevertheless we see that the impact on the route lengths is limited. E.g., for the topology *Zib54*, the increase is 11% for the extreme case when routing on a tree (for 34% of turned-off network interfaces). In general, the increase for this extreme case spans from 11% to 50%, and in average 27%. For the network with the larger impact, *Giul39*, the stretch is increased by 50% for the tree, with a saving of 56% of the network interfaces. We see that a saving of already 45% is attained for $OF = 2$, leading to a route length increase of only 18%.

Impact on the fault tolerance. We measure in Table III (b) the network fault tolerance as the *average number of disjoint paths* linking two nodes. We see that the full network topologies have an average number of disjoint paths between 1.82 and 4.90. When routing on the tree, this number is of course 1 as only one route exists, and this value is almost already attained for $OF = 3$. The drop is quick as all the ten networks have a number below 1.41 for $OF = 2$.

Discussion on Technology. When there is a failure between two nodes, it may be necessary to turn on some network interfaces to compute a new routing for some demands. Hence, this study show that the use of such energy efficient solutions is conditioned by the existence of *technologies allowing a quick switching on of network interfaces*. Network interfaces companies are currently working on designing this kind of interfaces [17].

We propose in [9] a solution for the routing with fault-tolerant spanners, such that there are two disjoint paths per demand. Therefore, the impact of links failures on the network will be reduced because a protection path with enough capacity will be available.

VII. Conclusion and Perspectives

In this work, we present through a simplified architecture the problem of minimizing power consumption in networks. We show non-approximation results. The simulations on real topologies show that the gain in energy is significant when some network interfaces can be turned-off.

- *At least one third of the network interfaces can be spared for usual range of demands.*
- For a medium-sized backbone network, this leads to a *reduction of power consumption* of approximately 33MWh per year (for *Cost266* with 37% of spared interfaces). For this estimation, we consider a scenario where the turned-off interfaces of our simplified architecture are 4-port Gb Ethernet linecards. We believe it corresponds to a reasonable capacity for backbone networks. We use the consumption values given in [1]: 100W for these linecards.
- The *route lengths* increase, but not too much: in average 27% for almost all studied topologies.
- *Fault tolerance* can be achieved with the use of *fast switching-on technologies* or by adding *disjoint path constraints* to the problem so that the network remains γ-connected, allowing a tolerance of $\gamma - 1$ failures.
- The bounds for specific topologies are useful for general networks to evaluate the overprovisioning factors needed for such energy-efficient solutions.

As part of future work, we plan to study a detailed cost function for a more complex router architecture. Moreover, we will carry on lab experiments on small network topologies to measure in practice the performance of the proposed energy-efficient routing.

Acknowledgements

This work has been partly funded by ANR DIMA-GREEN [6] (DesIgn and MAnagement of GREEN networks with low power consumption) and Région PACA.

References

[1] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright. Power awareness in network design and routing. In *IEEE INFOCOM*, pages 457–465, 2008.

[2] L. Chiaraviglio, E. Leonardi, and M. Mellia. How much can Internet be greened? In *GreenComm*, 2009.

[3] L. Chiaraviglio, M. Mellia, and F. Neri. Reducing power consumption in backbone networks. In *ICC*, 2009.

[4] D. Coudert, N. Nepomuceno, and H. Rivano. Power-efficient radio configuration in fixed broadband wireless networks. *Computer Communications*, 2010. To appear.

[5] G. Dantzig. *Linear programming and extensions*. Princeton Univ Pr, 1963.

[6] DIMAGREEN. http://www-sop.inria.fr/teams/mascotte/contrats/dimagreen/wiki/.

[7] S. Even, A. Itai, and A. Shamir. On the complexity of time table and multi-commodity flow problems. In *16th Annual Symposium on Foundations of Computer Science, 1975.*, pages 184–193, 1975.

[8] V. Gabrel, A. Knippel, and M. Minoux. Exact solution of multicommodity network optimization problems with general step cost functions. *Operations Research Letters*, 25(1):15–23, 1999.

[9] F. Giroire, D. Mazauric, J. Moulierac, and B. Onfroy. Minimizing Routing Energy Consumption: from theoretical to practical results. Research Report inria-00464318, INRIA, May 2010.

[10] G. Guisewite and P. Pardalos. Minimum concave-cost network flow problems: Applications, complexity, and algorithms. *Annals of Operations Research*, 25(1):75–99, 1990.

[11] M. Gupta and S. Singh. Greening of the internet. In *SIGCOMM*, 2003.

[12] F. Idzikowski, S. Orlowski, C. Raack, H. Woesner, and A. Wolisz. Saving energy in IP-over-WDM networks by switching off line cards in low-demand scenarios. In *ONDM*, 2010.

[13] P. Mahadevan, P. Sharma, S. Banerjee, and P. Ranganathan. Energy Aware Network Operations. In *HP Labs*, 2008.

[14] P. Mahadevan, P. Sharma, S. Banerjee, and P. Ranganathan. A Power Benchmarking Framework for Network Devices. In *IFIP NETWORKING 2009*, pages 795–808, may 2009.

[15] L. Nedevschi, S.and Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall. Reducing Network Energy Consumption via Rate-Adaptation and Sleeping. In *NSDI*, 2008.

[16] B. Nordman and K. Christensen. Reducing the energy consumption of networked devices. www.csee.usf.edu/˜christen/energy/lbnl_talk.pdf.

[17] W. Paper. Energy efficiency for network equipment: Two steps beyond greenwashing. Technical report, Juniper Networks Inc., 2008.

[18] B. Puype, W. Vereecken, D. Colle, M. Pickavet, and P. Demeester. Power Reduction Techniques in Multilayer Traffic Engineering. In *ICTON*, 2009.

[19] K. Spielberg. On the fixed charge transportation problem. In *Proceedings of the 1964 19th ACM national conference*, pages 11.101–11.1013, New York, NY, USA, 1964. ACM.

[20] B. Yaged Jr. Minimum cost routing for static network models. *Networks*, 1(2):139–172, 1971.