

# STREETE : Une ingénierie de trafic pour des réseaux de cœur énergétiquement efficaces

Radu CÂRPA, Olivier GLÜCK, Laurent LEFEVRE and Jean-Christophe MIGNOT

Inria Avalon - LIP Lab., École Normale Supérieure de Lyon, University of Lyon, France  
Email : {radu.carpa, olivier.gluck, jean-christophe.mignot}@ens-lyon.fr, laurent.lefevre@inria.fr

---

## Résumé

La consommation d'énergie est devenue un facteur limitant pour le déploiement d'infrastructures distribuées à grande échelle. Ce travail<sup>1</sup> a pour but l'amélioration de l'efficacité énergétique des réseaux de cœur en éteignant un sous-ensemble des liens grâce à une approche SDN (Software Defined Network). Nous présentons l'architecture logicielle STREETE (Segment Routing based Energy Efficient Traffic Engineering) qui adapte dynamiquement le nombre de liens du réseau qui sont allumés en fonction de la charge du réseau. Le cœur de la solution repose sur le protocole SPRING, aussi appelé Segment Routing, un protocole novateur en cours de standardisation par l'IETF. Les algorithmes ont été implémentés et évalués en utilisant le simulateur OMNET++. Les résultats expérimentaux montrent que le nombre de liens allumés peut être réduit de 44% tout en gardant une haute qualité de service.

**Mots-clés :** efficacité énergétique, réseaux de cœur, SDN, ingénierie de trafic, SPRING, Segment Routing, MPLS.

---

## 1. Introduction

Le réseau joue un rôle de plus en plus important dans notre monde hautement connecté et son trafic devrait être augmenté d'un facteur 3 dans les 5 années qui viennent<sup>2</sup>. Cela engendre une croissance de sa consommation énergétique. Heureusement, de nombreuses améliorations sont possibles car l'énergie consommée par les périphériques réseau est souvent indépendante de leur utilisation. Les réseaux de cœur, par exemple, consomment de nombreux KWh même s'ils ne font rien. Rendre la consommation dépendante de l'utilisation pourrait fournir un moyen d'économies substantielles.

Parmi les techniques permettant de réduire la consommation énergétique des réseaux de cœur on trouve le re-routage du trafic et l'utilisation de modes basse consommation. Les techniques de re-routage de données sur des routes alternatives pour décharger certains liens sont possibles car les opérateurs surdimensionnent les réseaux afin de supporter les pics de consommation et pour offrir une haute qualité de service. Même pendant les pics de consommation,

---

1. Ce travail est réalisé dans le cadre du projet européen CHIST-ERA STAR "SwiTching And tRansmission" (<http://www.chistera.eu/projects/star>), qui a pour objectif la réduction d'un facteur 100 de la consommation des réseaux de cœur à l'horizon 2020.

2. <http://ciscovni.com/forecast-widget/index.html>

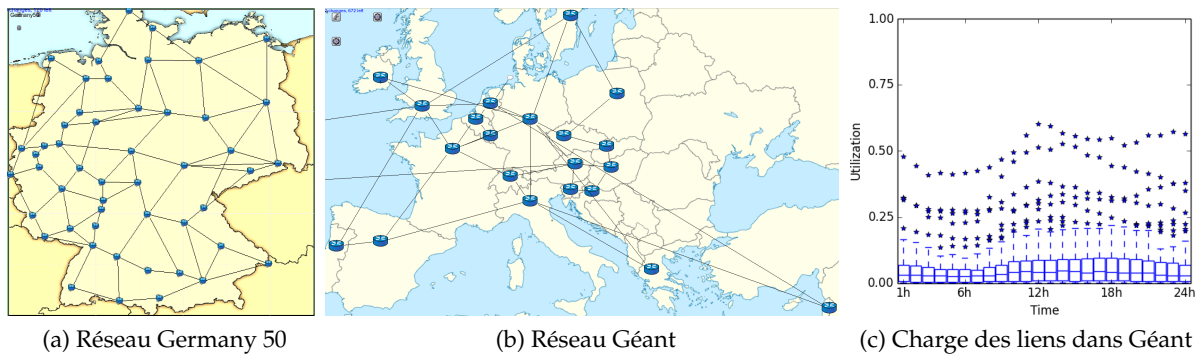


FIGURE 1 – Les topologies utilisées et la charge de liens dans Géant

les liens sont rarement utilisés à plus de 50% de leur capacité [9]. La figure 1c montre la distribution des utilisations des liens du réseau Géant. On remarque que la charge de la plupart de liens n'excède pas 25% de leur capacité même pendant les heures de pointe.

Dans cet article, nous nous intéressons au problème de la consommation d'énergie dans les réseaux de cœur en changeant l'état des ports des routeurs et des transpondeurs aux deux extrémités d'un lien. L'état de ces composants est mis en mode veille dès qu'un lien ne doit plus transférer de données ; il est remis en mode opérationnel quand nécessaire. Dans la suite, nous utiliserons les termes allumage et extinction des liens.

La plupart des travaux qui cherchent à adapter le nombre de liens allumés à l'importance du trafic se limitent au niveau conceptuel. Ils cherchent à minimiser le nombre de liens actifs en résolvant des formulations en programmation linéaire ou en utilisant des heuristiques. Peu nombreux sont ceux qui vont jusqu'aux détails de l'implémentation et ils considèrent peu, ou pas du tout, les conditions réelles. Nous avons implémenté notre solution dans le simulateur d'événements discrets OMNET++. Nos expérimentations en considérant de vraies topologies de réseaux (Germany50 et Géant, voir Figure 1) et de vraies matrices de trafic nous permettent de quantifier le compromis entre économie d'énergie et performance.

Pour re-router le trafic, nous utilisons le nouveau protocole SPRING géré par un contrôleur SDN (Software Defined Networks), contrairement à la plupart des autres travaux qui utilisent MPLS+RSVP-TE ou la variation dynamique des métriques IP. SPRING est particulièrement bien adapté à la reconfiguration dynamique des réseaux. De plus, son utilisation rend l'implémentation plus facile. A notre connaissance, SPRING n'a jamais été utilisé dans la littérature pour réduire la consommation d'énergie dans les réseaux.

La solution implémentée [4] est proposée sous la forme d'une architecture logicielle que nous nommons STREETE (SegmentT Routing based Energy Efficient Traffic Engineering). Elle permet d'éteindre/allumer les liens dynamiquement en fonction de la charge du réseau. La genericité de notre approche permet des améliorations graduelles par une simple modification des algorithmes la composant.

Le reste de cet article est composé de la manière suivante. La section 2 est consacrée à l'état de l'art. La section 3 décrit nos hypothèses et les concepts comme SDN et le protocole SPRING. Nous présentons ensuite notre solution dans la section 4 et l'analyse des résultats de nos simulations dans la section 5. La section 6 conclut l'article.

## 2. État de l'art

Gupta *et al.* [8] ont été parmi les premiers à penser à mettre des interfaces en mode veille pour économiser de l'énergie. Cependant, dans les réseaux de cœur, les liens ne peuvent pas être éteints sans perdre de données. Il a été montré que même des interruptions de quelques microsecondes sont durs à réaliser du fait du très petit délai inter-paquets [2]. Pour éviter ce problème, des transmissions de paquets en rafale ont été envisagées [14]; ceci est malheureusement quasiment impossible avec les réseaux modernes à 100 Gbps.

Une solution distribuée reposant sur le reroutage des flux a été proposée par Vasic *et al.*. Ils ont réussi à éteindre 21% des liens dans un réseau issu du monde réel [18]. Cependant, cette étude reste conceptuelle et les détails de l'implémentation ne sont pas publiés. Une autre équipe [1] propose que chaque nœud observe l'utilisation ses liens adjacents et décide s'il doit les éteindre ou pas. Des mécanismes d'apprentissage sont utilisés pour éviter d'éteindre des liens dont l'extinction a provoqué des congestions dans le passé. Cette solution repose sur IGP-TE (Interior Gateway Protocol - Traffic Engineering), mais sous-estime son surcoût. Après une étude détaillée, nous avons découvert que du fait de la mauvaise estimation de la fréquence des échanges IGP-TE [15] le surcoût annoncé de 0.52% est en fait une augmentation de 30% du nombre d'inondations faites pas IGP-TE. Cela n'est pas négligeable car les inondations sont coûteuses en termes de messages et de puissance de calcul. De plus, le fait d'utiliser le routage IP classique plutôt que MPLS peut induire des instabilités lors des reconfigurations.

Zhang *et al.* ont proposé une approche centralisée basée sur MPLS+RSVP-TE [19]. Leur évaluation s'appuie sur des matrices de trafic estimées ce qui a été prouvé relativement peu efficace [7]. Ils ont opté pour une approche hybride où MPLS+RSVP-TE est utilisé parallèlement à un routage par le plus court chemin. Notre travail utilise le protocole SPRING qui a les mêmes avantages et une complexité moindre.

Les résultats sur l'économie d'énergie varient grandement selon les solutions proposées. Par exemple, dans certains travaux, les performances d'approches gloutonnes telles que "sélectionner le lien le moins congestionné" sont annoncées être proches de l'optimal en termes de consommation d'énergie (Kamola *et al.* [11]). Par opposition, dans d'autres travaux, des techniques d'optimisations stochastiques semblent être supérieures à des approches gloutonnes (Bonetto *et al.* [3]). Nous reviendrons sur ces résultats dans la section 5.2.1 de cet article.

## 3. Hypothèses et contexte

Nous considérons que les variations de trafic dans les réseaux de cœur suivent un profil journalier et sont lentes du fait de l'agrégation des nombreux flux [10]. Le tableau 1 montre la puissance consommée par les interfaces des équipements de cœur [17]. Dans notre modèle, nous considérons

TABLE 1 – Consommation des ports des routeurs

Vitesse du port (Gbps)	Consommation (W)	
	Carte du port	Transpondeur
10	10	50
40	35	100
100	135	150
400	335*	300*

\* projection mathématique

qu'éteindre un lien correspond à l'extinction d'une partie du circuit intégré et du transpondeur à chaque extrémité (éteindre un lien 100Gbps =  $2(135 + 150) = 570$  W).

### 3.1. SDN

Dans les réseaux traditionnels, les plans de contrôle et de données sont situés dans les nœuds réseau, et les ajustements de la politique réseau nécessitent souvent l'accès et la configuration de tous les équipements. SDN permet de simplifier cette gestion en séparant les plans de

contrôle et de données. Le plan de contrôle contient l'intelligence du réseau qui détermine les chemins des flots de données et les programme sous forme de plan de données. Les commutateurs du plan de données sont uniquement responsables du transfert des paquets qui arrivent sur un port d'entrée de l'équipement vers le bon port de sortie.

Les contrôleurs SDN exploitent leur connaissance de la globalité du réseau pour effectuer leurs calculs et effectuer les changements des plans de données des nœuds via une API prédéfinie. La flexibilité de SDN offre de grandes possibilités pour la gestion des économies d'énergie.

Comparés aux réseaux des centres de données, les réseaux optiques de cœur posent des problèmes pour lesquels l'implémentation actuelle de SDN, OpenFlow, n'est pas prête. Cependant des chercheurs ont réussi à transférer le concept SDN vers les réseaux de cœur [16]. Dans cet article, nous faisons l'hypothèse de la disponibilité des plans de données que nous pouvons modifier à notre guise, par exemple, en modifiant leur code source.

### 3.2. Le protocole SPRING

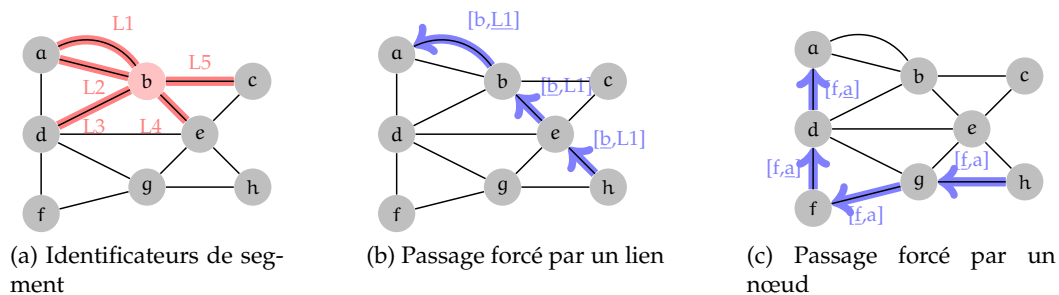


FIGURE 2 – Le protocole SPRING

Le protocole SPRING [6], aussi appelé Segment Routing, est un draft IETF commencé en 2013 avec pour but de remplacer le tandem MPLS+RSVP-TE pour l'ingénierie du trafic. Il combine la puissance du routage par la source (flexibilité de la gestion du trafic) et le routage par le plus court chemin (moins de signalisation que MPLS et moins d'entête que Myrinet).

Le protocole SPRING utilise le même concept de commutation par étiquette que MPLS. Il peut utiliser le plan de données de MPLS sans modification. En revanche, le plan de contrôle nécessite une refonte complète. Par exemple, les étiquettes appelées Segment Identifiers (SID) ont une portée globale, contrairement à MPLS où leur portée est locale. Dans notre étude, nous nous intéressons à 2 types d'identificateurs présentés sur la figure 2a : nodaux (*nodal SID*) et d'adjacence (*adjacency SID*).

- Un identificateur nodal est unique pour tout le réseau et identifie un nœud (a, b, c, ..., h).
- Un identificateur d'adjacence est local à un nœud et identifie une interface sortante (dans le cas du nœud b, les interfaces sortantes sont identifiées par L1, L2, L3, L4 et L5).

Contrairement à MPLS, la distribution des étiquettes est faite en utilisant une extension de IGP plutôt qu'en utilisant des protocoles spéciaux (LDP/RSVP-TE). Après la découverte du réseau, envoyer un paquet au nœud a en utilisant le chemin le plus court se résume à lui ajouter l'étiquette a dans l'en-tête. Cependant, une gestion beaucoup plus fine du trafic est possible :

- En ajoutant l'en-tête [b, L1], le nœud h force le paquet à aller au nœud a en utilisant le lien L1 (voir Figure 2b).
- Sur la Figure 2c, si h veut envoyer un paquet à a en passant par f, il lui ajoutera l'entête [f, a]. Le paquet prendra alors le plus court chemin vers f, puis continuera jusqu'à a.

SPRING est un bon choix pour effectuer du re-routage dynamique aussi bien avec SDN que tout seul. Étant un protocole de routage par la source, il permet une mise en place rapide des

flux et une facile reconfiguration des circuits virtuels, le tout avec un surcoût minimal. Les changements ne sont à appliquer que sur les équipements d'entrée. Il n'y a pas de perte de temps ni de signalisation à faire pour re-configurer les équipements intermédiaires. C'est particulièrement appréciable quand beaucoup de changements de politique de routage sont nécessaires. Nous utilisons aussi SPRING du fait des problèmes de convergence qui peuvent arriver avec MPLS+RSVP-TE en cas de rupture d'un lien<sup>3</sup>. En pratique, ce problème est repoussé en limitant le nombre de canaux virtuels utilisés. De plus, une extension spéciale existe dans RSVP-TE pour accélérer la convergence. Cependant, dans notre cas, du fait du re-routage des canaux virtuels sur un sous-ensemble des chemins et de l'extinction des autres liens, nous augmentons la charge sur certains routeurs et le nombre de chemins virtuels (LSP : Label Switched Path) par routeur. Pour ces raisons, SPRING convient mieux à nos besoins car il est plus robuste pour un grand nombre de chemins virtuels nécessitant une reconfiguration fréquente.

## 4. L'architecture de STREETE

### 4.1. Analyse

Dans cette section, nous formalisons le problème de l'efficacité énergétique de l'ingénierie de trafic. Le but est d'éviter de perdre la connectivité entre deux nœuds du réseau. Nous utiliserons le terme **contrainte de connectivité** pour désigner la conservation de l'intégrité du réseau. Nous présentons notre approche en 3 étapes :

- Sélection des liens à allumer ou éteindre,
- Calcul des nouvelles routes en tenant compte des allumages ou extinctions,
- Envoi des routes vers les équipements du réseau et allumage ou extinction physique des liens.

#### 4.1.1. Sélection des liens à éteindre ou allumer

Nous implémentons la politique d'extinction du lien le moins congestionné car c'est la plus utilisée dans la littérature pour évaluer les résultats. Cette étape consiste à trier les liens en fonction de leur vitesse de transmission instantanée. La complexité est en  $O(E \cdot \log(E))$ , où  $E$  est le nombre de liens ( $E$  pour *edges* en anglais). La suite consiste à essayer d'éteindre les nœuds qui transmettent le moins de données. Cela est fait en testant, pour chaque lien, si l'extinction viole la contrainte de connectivité ou pas.

Dans le pire cas, le coût est  $O(E \cdot (E + V))$  :  $V$  étant le nombre de nœuds ( $V$  pour *Vertices* en anglais), on effectue  $E$  recherches en largeur d'abord, pour tester la connectivité du réseau. Dans la suite de cet article, nous utiliserons le terme **<SelectLinksToOff/On>** pour nommer l'étape de sélection des liens à éteindre/allumer.

#### 4.1.2. Calcul des nouvelles routes pour les flux de données

Avant d'effectivement éteindre ou allumer les liens sélectionnés par **<SelectLinksToOff/On>**, nous devons calculer les nouvelles routes pour chaque flux les traversant. Il peut arriver que le réseau ne soit pas capable de router tous les flux quand les liens sélectionnés sont éteints. De plus, laisser chaque nœud re-calculer les chemins les plus courts à travers le réseau n'est pas forcément judicieux, car cela peut quand même mener à des congestions : une congestion peut apparaître même si un lien est allumé, par exemple quand il fait partie des plus courts chemins pour un grand nombre de nœuds.

Pour calculer les nouvelles routes, le contrôleur SDN utilise sa connaissance globale du trafic du réseau et résout ce problème connu sous le nom de *Multicommodity Flow Problem*. C'est un pro-

3. <http://www.ietf.org/mail-archive/web/bmwg/current/msg01502.html>

blème NP-complet mais pour lequel des techniques d'approximation existent [13]. Pour cela, le contrôleur SDN doit d'abord connaître la bande passante utilisée entre toutes les extrémités du réseau *i.e.* la matrice de trafic. Des statistiques doivent être collectées sur les commutateurs SDN d'entrée dans le domaine de routage. Notons qu'à cette étape, l'algorithme peut aussi calculer des chemins de secours [12].

Dans nos simulations, nous implémentons une technique qui calcule le plus court chemin entre chaque paire de nœud du réseau et teste la possibilité de router toutes les données sans congestion. La complexité de ce calcul est en  $O(V \cdot (E + V \cdot \log V))$  et utilise l'algorithme de Dijkstra pour chaque nœud. Cette complexité pourrait être réduite en utilisant un algorithme optimisé [5].

En ce qui concerne le rallumage, notre algorithme est des plus simples. Il cherche l'utilisation du nœud le plus utilisé et, si sa charge est supérieure à 75%, tous les liens sont rallumés et les chemins les plus courts recalculés. Nous appelons cet algorithme **<ComputeNewRoutes>**.

### 4.1.3. Re-routage et extinction/allumage des liens

---

#### Algorithm 1: SDN centralisé : <RerouteAndSwitchOff>

---

**Data:** List-Of-Links :[(A-B),(C-D),...] et les nouvelles routes. /\* Fournies par  
<SelectLinksToOff> et <ComputeNewRoutes> respectivement \*/

```
1 begin
2   Le contrôleur fournit à tous les routeurs les nouveaux chemins à utiliser pour les chemins virtuels ;
3   Le contrôleur demande aux nœuds adjacents aux liens (A,B,...) de les éteindre ;
4   Les nœuds A,B,C,... envoient un accusé de réception (ACK) au contrôleur ;
5   Le contrôleur SDN envoie une inondation IGP : le lien A-B est déconnecté /* l'inondation
      est nécessaire car à la ligne 2 il se peut qu'on ne contacte pas tous
      les liens */
6   Tous les routeurs appliquent les changements aux tables d'acheminement pour re-router sur les
      nouveaux chemins ;
7   Les nœuds A,B,C,... arrêtent d'envoyer les messages IGP HELLO ;
8   Les liens (A-B,C-D,...) entrent en mode veille / négocient l'arrêt ;
```

---

---

#### Algorithm 2: Main loop

---

```
1 while L'ingénierie du trafic énergétiquement efficace est
   active do
2   List-Of-Links = <SelectLinksToOn>();
3   if <ComputeNewRoutes>(All-links-that-are-ON ∪
   List-Of-Links) then
4     | <RerouteAndSwitchOn>(List-Of-Links);
5   List-Of-Links = <SelectLinksToOff>();
6   if <ComputeNewRoutes>(All-links-that-are-ON \
   List-Of-Links) then
7     | <RerouteAndSwitchOff>(List-Of-Links);
```

---

Une fois les nouvelles routes calculées, le contrôleur informe chaque nœud des routes qui doivent être introduites dans sa table d'acheminement. Après cela, les liens sont éteints ou allumés. Les étapes nécessaires à l'extinction des liens sont décrites dans l'algorithme 1. Dans notre exemple, les liens entre les nœuds A et B, C et D etc. ont été sélectionnés pour extinction. L'algorithme présente les étapes pour éteindre ces liens.

Le surcoût de cette solution en termes d'échanges de messages est inférieur aux solutions distribuées. Dans les cas distribués, les nœuds doivent se synchroniser afin d'éviter d'éteindre un lien qui pourrait rompre la contrainte de connectivité. Dans la version centralisée, s'il n'y a pas de défaillance réseau, ce risque n'existe pas. Notons qu'il est possible de rendre l'algorithme tolérant aux pannes en plaçant des contraintes de protection dans <SelectsLinksToOff> et <ComputeNewRoutes>. Nous n'aborderons pas cette question dans cet article.

#### 4.1.4. La boucle principale

<MainLoop>, présenté dans l'algorithme 2, montre comment les 3 parties composantes de notre solution interagissent. L'algorithme cherche les liens qui peuvent être éteints ou allumés. S'il en trouve, il appelle <RerouteAndSwitchOn> ou <RerouteAndSwitchOff>.

## 5. Évaluation des performances

### 5.1. Environnement expérimental

L'approche de STREET a été implémentée dans le simulateur OMNET++. Nous avons aussi développé une version expérimentale du protocole SPRING pour re-router dynamiquement les données dans le réseau afin d'éviter les liens éteints. Aux extrémités du réseau, lors de la classification des paquets et de leur affectation à des circuits virtuels, les nœuds entrants recueillent des statistiques sur le volume de données envoyé vers l'intérieur du réseau. Le contrôleur SDN collecte ces statistiques et génère la matrice de trafic utilisée par nos algorithmes.

Nous avons testé notre solution sur les réseaux résumés dans la table 2. Le réseau Germany50 est un vrai réseau de cœur, *i.e.* le trafic est l'agrégation d'un grand nombre de flux et reste stable dans le temps. Bien que Géant soit l'interconnexion de réseaux de recherche, le degré d'agrégation est assez important et le profil du trafic est similaire à celui des réseaux de cœur.

Nous avons évalué STREETE en simulant de vrais réseaux avec de véritables matrices de trafic issues de SNDLib<sup>4</sup> qui ont été mises à l'échelle pour prendre en compte la rapidité plus élevée des réseaux actuels. Les liens du réseau sont éteints un par un tant qu'aucun d'eux n'atteint une charge supérieure à 60%. Dès que cela arrive, nous arrêtons l'extinction des liens. Dès que l'utilisation d'un lien dépasse 75%, le réseau est entièrement rallumé.

Les matrices de trafic de SNDLib contiennent, par tranche de 5 minutes, la quantité de données transférée sur les 5 minutes précédentes. Cela donne 288 (24h \* 60m / 5m) matrices de trafic par jour. Dans notre simulation, on compresse chaque 5 minutes en 2 secondes de simulation. Cela a permis de réduire le temps de simulation sans impacter les résultats. Cela parce qu'on s'intéresse aux moments de changement des matrices de trafic et pas aux 5 minutes de stabilité durant l'utilisation de la même matrice.

Le délai de propagation est calculé comme étant le temps mis par la lumière pour traverser la fibre optique. La distance entre les nœuds est la distance à vol d'oiseau. Les temps de traitement des équipements sont simulés par OMNET++.

### 5.2. Resultats

#### 5.2.1. Consommation d'énergie

Nous avons simulé une journée d'utilisation pour chaque réseau. Sur la figure 3, les lignes vertes représentent le pourcentage de liens laissés allumés par notre algorithme. En utilisant les valeurs du tableau 1 pour des liens à 100Gbps, nous obtenons les réductions de consommation d'énergie du tableau 3.

TABLE 2 – Topologies des réseaux étudiés.

Réseau	Nombre de nœuds	Nombre de liens
Géant (Fig. 1b)	22*	36
Germany50 (Fig. 1a)	50	88

\*Le nœud de New-York n'est pas visible sur la figure 1b, mais les 2 liens le connectant au reste du réseau le sont.

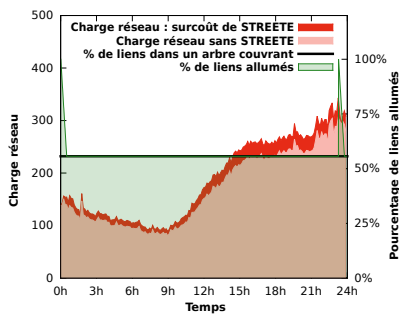
4. <http://sndlib.zib.de/>

En début de journée, quand notre algorithme commence à être exécuté, les deux réseaux commencent à éteindre des liens les uns après les autres et convergent rapidement vers un arbre recouvrant. L'extinction de liens supplémentaires romprait la contrainte de connectivité. Dans le cas du réseau Germany50, nous observons un rallumage à 23 :00h. Il s'explique par une grande augmentation de la demande entre deux nœuds. Nous ne connaissons pas la raison de cette augmentation inhabituelle. Notre algorithme a réagi de manière appropriée en rallumant le réseau, lui restituant ainsi ses pleines capacités.

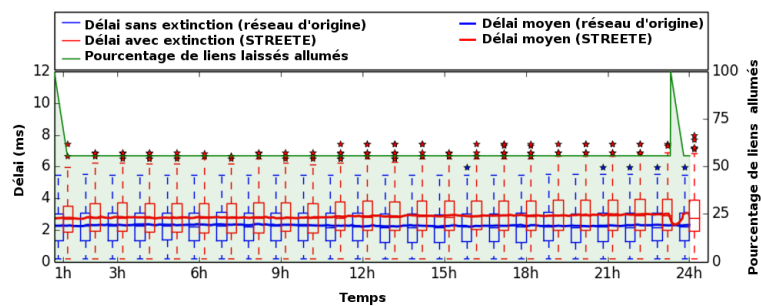
Nous nous sommes rendu compte que, contrairement à ce qui est affirmé dans [3], même l'algorithme glouton utilisé dans cet article pour éteindre les liens permettait d'atteindre de bonnes économies d'énergie. Cela s'explique par le très grand surdimensionnement des réseaux étudiés. Une étude détaillée *a posteriori* de [3] montre qu'ils comparent une approche gloutonne agissant au niveau des liens physiques (coupure brutale de la fibre, i.e. toutes les longueurs d'onde) avec des algorithmes stochastiques agissant au niveau d'une seule longueur d'onde.

TABLE 3 – Économies d'énergie pour une journée de simulation.

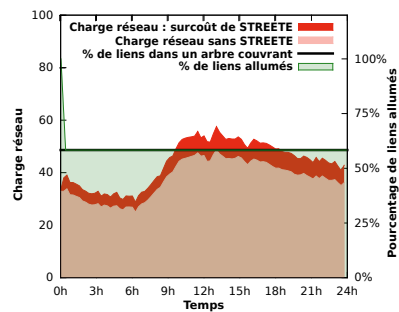
Réseau	Durée moyenne d'extinction des liens	Économie d'énergie (kWh)
Géant	14.88 sur 36	204
Germany50	38.18 sur 88	522



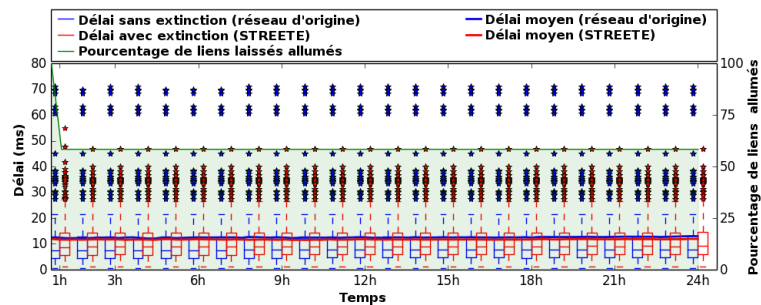
(a) Impact sur la charge des liens



(b) Impact sur le délai de bout en bout



(c) Impact sur la charge des liens



(d) Impact sur le délai de bout en bout

FIGURE 3 – Résultats des simulations sur les réseaux Germany50 (en haut) et Géant (en bas)

### 5.2.2. Impact sur la charge réseau

Le fait de re-router des données sur des chemins qui ne sont pas les plus courts induit des surcoûts dans la charge du réseau. Le surcoût de notre solution est montré sur les figures 3a et 3c. Son maximum est de 18.56%. Il est à noter que même avec ce surcoût le réseau converge vers un arbre couvrant. On ne peut plus éteindre de liens sans violer la contrainte de connectivité. Comme on pouvait s'y attendre, quand tous les liens sont allumés, le surcoût de notre solution en termes de charge réseau est nul. Cela peut être remarqué sur les figures 3a et 3c à travers le fait que la courbe rouge foncée est superposée à la rose. Ce phénomène peut être remarqué au



début des simulations et, dans le cas du réseau Germany50, juste après l'allumage de 23h.

### 5.2.3. Impact sur le délai de bout en bout

Les figures 3b et 3d montrent l'impact de notre approche sur les délais de bout en bout. Pour chaque nœud source et destination, nous calculons le délai moyen de bout en bout sur une durée d'une heure. Les boîtes à moustache représentent la distribution des moyennes entre chaque couple (source, destination) durant l'heure précédente. Les boîtes bleues représentent les réseaux d'origine, les rouges représentent les réseaux où les liens ont été éteints. Les délais moyens sont aussi représentés sous forme de courbes continues pour les mêmes réseaux avec les mêmes couleurs.

Concernant le réseau Germany50, les délais augmentent de 37% dans le pire des cas mais dans la plupart des cas l'augmentation est plutôt de l'ordre des 20%. Le réseau Géant présente un résultat aberrant. Comme prévu, la plupart du temps le délai est légèrement augmenté. Néanmoins, on observe de manière étrange à première vue que certaines valeurs diminuent ! Cela s'explique par le routage en nombre minimum de sauts. Le réseau Géant a une topologie irrégulière et il s'avère que le routage en nombre de sauts minimum ne lui est pas bien adapté : dans certains cas, le trafic entre deux nœuds européens passe par New-York ! L'extinction d'un des 2 liens incriminés force les flux à passer par de multiples sauts, mais ils restent en Europe. Cela induit une réduction du délai pour certains couples (source, destination).

### 5.2.4. Impact sur la perte de paquets

La simulation n'a montré aucune perte de paquet liée à nos algorithmes. Dans le cas des réseaux de cœur, les flux agrégés varient lentement ou, pour le moins, pas assez rapidement pour que notre solution induise une perte de paquets. L'algorithme a le temps de rallumer les liens éteints dès que le dépassement de seuil est détecté, redonnant au réseau son entière capacité.

## 6. Conclusion

Nous avons présenté l'implémentation d'une technique d'ingénierie de trafic permettant d'économiser de l'énergie dans les réseaux de cœur. L'ingénierie de trafic prenant en compte l'énergie avait été étudiée dans des travaux précédents mais aucun n'avait présenté d'implémentation réelle. Ne pas tester la faisabilité des solutions peut conduire à de mauvaises estimations ou à des solutions instables (voir fin de la section 3.2). Nous avons implémenté un prototype fonctionnel dans le simulateur OMNET++.

Les réseaux actuels tendent de plus en plus vers des architectures centralisées et l'utilisation de SDN est en constante augmentation dans les réseaux des centres de données de production. Nous pensons que la philosophie de SDN peut être étendue aux réseaux de cœur. La simplicité de l'implémentation de notre solution montre que SDN peut aussi être un bon moyen de faire facilement des économies d'énergie dans les dispositifs réseau.

Contrairement aux travaux précédents, nous proposons d'utiliser le protocole de transport SPRING pour améliorer la stabilité des solutions de gestion de trafic efficaces en énergie. A notre connaissance, nos travaux sont les premiers à utiliser cette approche pour les réseaux de cœur. La flexibilité de ce protocole de routage est particulièrement adaptée aux nombreux changements de routes qui apparaissent quand on éteint et rallume des liens. Notons que SPRING est particulièrement bien adapté à SDN puisqu'il a été conçu pour lui.

En ayant recours à des simulations, nous avons montré qu'il était possible d'éteindre jusqu'à 44% des liens pour faire des économies. Nous avons aussi découvert en utilisant de vraies matrices de trafic que les réseaux de cœurs étaient largement surdimensionnés. Même des ap-

proches gloutonnes peuvent facilement atteindre le niveau maximum de réduction d'énergie. En fait, le goulot d'étranglement pour réaliser des économies d'énergie est la contrainte de connectivité.

## Bibliographie

1. Bianzino (A.), Chiaraviglio (L.) et Mellia (M.). – Grida : A green distributed algorithm for backbone networks. – In *GreenCom, 2011 IEEE*, pp. 113–119, Sept 2011.
2. Bolla (R.), Bruschi (R.) et al. – Energy efficiency in optical networks. – In *Telecommunications Network Strategy and Planning Symposium, 2012 XVth International*, pp. 1–6, Oct 2012.
3. Bonetto (E.), Chiaraviglio (L.), Idzikowski (F.) et Le Rouzic (E.). – Algorithms for the multi-period power-aware logical topology design with reconfiguration costs. *Optical Communications and Networking, IEEE/OSA Journal of*, vol. 5, n5, May 2013, pp. 394–410.
4. Carpa (R.), Glück (O.) et Lefevre (L.). – Segment Routing based Traffic Engineering for Energy Efficient Backbone Networks. – In *ANTS2014 : IEEE International Conference on Advanced Networks and Telecommunications Systems*, New Delhi, India, décembre 2014.
5. Demetrescu (C.) et Italiano (G. F.). – A new approach to dynamic all pairs shortest paths. *J. ACM*, vol. 51, n6, novembre 2004, pp. 968–992.
6. Filsfils (C.), Previdi (S.) et Bashandy (A.). – Segment routing architecture", draft-filsfils-spring-segment-routing-02 (work in progress), May 2014.
7. Gunnar (A.), Johansson (M.) et Telkamp (T.). – Traffic matrix estimation on a large ip backbone : A comparison on real data. – In *ACM SIGCOMM '04*, pp. 149–160, NY, USA, 2004.
8. Gupta (M.) et Singh (S.). – Greening of the internet. – In *SIGCOMM '03*, NY, USA, 2003.
9. Hassidim (A.), Raz (D.), Segalov (M.) et Shaqed (A.). – Network utilization : The flow view. – In *INFOCOM, 2013 Proceedings IEEE*, pp. 1429–1437, April 2013.
10. Juva (I.), Susitaival (R.) et Peuhkuri (M.). – Traffic characterization for traffic engineering purposes : analysis of funet data. – In *NGIN, 2005*, pp. 404–411, april 2005.
11. Kamola (M.) et Arabas (P.). – Shortest path green routing and the importance of traffic matrix knowledge. – In *Digital Communications - Green ICT (TIWDC), 2013 24th Tyrrhenian International Workshop on*, pp. 1–6, Sept 2013.
12. Liu (V. Y.) et Tipper (D.). – Spare capacity allocation using shared backup path protection for dual link failures. *Comput. Commun.*, vol. 36, n6, mars 2013, pp. 666–677.
13. Madry (A.). – Faster approximation schemes for fractional multicommodity flow problems via dynamic graph algorithms. – In *Proceedings of the Forty-second ACM Symposium on Theory of Computing, STOC '10, STOC '10*, pp. 121–130, New York, NY, USA, 2010. ACM.
14. Nedeveschi (S.), Popa (L.) et al. – Reducing network energy consumption via sleeping and rate-adaptation. – In *NSDI'08*, pp. 323–336, Berkeley, USA, 2008.
15. Salsano (S.), Botta (A.), Iovanna (P.), Intermite (M.) et Polidoro (A.). – Traffic engineering with ospf-te and rsvp-te : Flooding reduction techniques and evaluation of processing cost. *Comput. Commun.*, vol. 29, n11, juillet 2006, pp. 2034–2045.
16. Siqueira (M.), Oliveira (J.) et al. – An optical sdn controller for transport network virtualization and autonomic operation. – In *Globecom Workshops '13*, pp. 1198–1203, Dec 2013.
17. Van Heddeghem (W.), Idzikowski (F.) et al. – Power consumption modeling in optical multilayer networks. *Photonic Network Communications*, vol. 24, n2, 2012, pp. 86–102.
18. Vasić (N.) et Kostić (D.). – Energy-aware traffic engineering. – In *Proceedings of e-Energy '10*, pp. 169–178, Passau, Germany, 2010. ACM.
19. Zhang (M.), Yi (C.), Liu (B.) et Zhang (B.). – Greente : Power-aware traffic engineering. – In *Network Protocols (ICNP), 2010 18th IEEE International Conference on*, pp. 21–30, Oct 2010.