

Modèle de routage écoénergétique dans les réseaux définis par logiciel

par

Samy ZEMMOURI

MÉMOIRE PRÉSENTÉ À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
COMME EXIGENCE PARTIELLE À L'OBTENTION DE LA MAÎTRISE EN GÉNIE
AVEC MÉMOIRE
CONCENTRATION : RÉSEAUX DE TÉLÉCOMMUNICATIONS
M.Sc.A.

MONTRÉAL, LE 1 JUIN 2017

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC



Samy ZEMMOURI, 2017



Cette licence Creative Commons signifie qu'il est permis de diffuser, d'imprimer ou de sauvegarder sur un autre support une partie ou la totalité de cette oeuvre à condition de mentionner l'auteur, que ces utilisations soient faites à des fins non commerciales et que le contenu de l'oeuvre n'ait pas été modifié.

PRÉSENTATION DU JURY

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE:

M. Mohamed Cheriet, Directeur de Mémoire
Département de génie de la production automatisée à l'École de Technologie Supérieure

M. Zbigniew Dziong, Président du Jury
Département de génie électrique à l'École de Technologie Supérieure

M. Mohamed Faten Zhani, membre du jury
Département de génie logiciel et des TI à l'École de Technologie Supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 17 MAI 2017

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

REMERCIEMENTS

La réalisation de ce mémoire a été possible grâce au concours de plusieurs personnes à qui je voudrais témoigner toute ma reconnaissance.

Je voudrais tout d'abord adresser toute ma gratitude à Mr. Mohammed Cheriet, professeur titulaire à l'école de technologie supérieure et directeur du laboratoire Synchronmedia. pour sa patience, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter ma réflexion.

Je voudrais aussi témoigner mes remerciements, à Monsieur Kim Nguyen, Professeur assistant et associé de recherche, au laboratoire Synchronmedia, pour son assistance et son dévouement tout au long de ces deux années afin de nous procurer un meilleur apprentissage.

J'adresse mes vifs remerciements et ma gratitude aux membres du jury qui m'ont fait l'honneur d'accepter d'évaluer mon travail.

Mes remerciements s'adressent également aux membres du laboratoire Synchronmedia, à ma famille, ma fiancée ainsi que mes amis qui ont été toujours présents pour moi et qui m'ont toujours soutenu et encouragé.

Enfin que tous ceux qui ont contribué de près ou de loin à l'élaboration de ce mémoire trouvent ici toute notre gratitude.

MODÈLE DE ROUTAGE ÉCOÉNERGÉTIQUE DANS LES RÉSEAUX DÉFINIS PAR LOGICIEL

Samy ZEMMOURI

RÉSUMÉ

La croissance exponentielle des utilisateurs du réseau et leurs demandes de communication ont conduit à un accroissement considérable de la consommation d'énergie dans les infrastructures réseau. Un nouveau paradigme de réseautage appelé *Software Defined Networking* (SDN) a récemment vu le jour, dans lequel le plan de données, responsable du transfert de paquets est découplé du plan de contrôle responsable de la prise de décision.

Le SDN simplifie la gestion du réseau en offrant la programmabilité des équipements réseau et permet la redirection rapide des flux. Il est évident que le SDN ouvre de nouvelles opportunités prometteuses pour améliorer la performance du réseau en général et l'efficacité énergétique en particulier. Dans ce mémoire, compte tenu de la charge de trafic actuelle d'un réseau basé sur le paradigme SDN, nous exploitons les techniques d'adaptation dynamique des débits des ports de transmission des données, afin de réduire la consommation d'énergie du réseau.

L'idée principale de ce mémoire est de trouver une distribution des flux sur des chemins pré-calculés qui permet de réduire la consommation énergétique du réseau par l'adaptation dynamique des débits de transmission des liaisons. Nous formulons d'abord le problème comme un problème de programmation linéaire entière (ILP). Ensuite, nous présentons quatre différents algorithmes efficaces du point de vue de temps de traitement à savoir, l'algorithme First Fit, l'algorithme Best Fit, l'algorithme Worst Fit et un algorithme génétique (AG) pour résoudre le problème pour le cas d'une topologie réseau réaliste.

Les résultats de la simulation montrent que la méthode basée sur l'AG surpasse systématiquement les autres heuristiques proposées et qu'en appliquant cet algorithme, entre 32 % et 47 % de l'énergie peut être économisée en fonction de la taille et de la densité de la topologie du réseau.

Mots clés: SDN, Algorithmes génétiques, Heuristiques, Adaptive Link Rate, Optimisation

ENERGY-AWARE ROUTING FOR SOFTWARE-DEFINED NETWORKS

Samy ZEMMOURI

ABSTRACT

The exponential growth of network users and their communication demands has led to a tangible increment of energy consumption in network infrastructures. A new networking paradigm called *Software Defined Networking* (SDN) was recently emerged, in which the packet forwarding data plane is decoupled from decision making control plane.

SDN simplifies network management by offering programmability of network devices, monitoring the real-time traffic rates and ability of fast rerouting. It also assists to lower link data rates via rate-adaptation technique which reduces considerably the power consumption of network. It is deemed that SDN opens new promising opportunities to improve network performance in general and energy efficiency in particular. In this paper, given current traffic load of an SDN based network, we exploit the rate adaptation techniques in SDN-enabled devices in order to reduce the energy consumption.

The main idea behind this thesis is to find a distribution of flows over pre-calculated paths which allows to adapt the transmission rate of maximum links into lower states. We first formulate the problem as an Integer Linear Programming (ILP) problem. Then, we present four different computationally efficient algorithms namely greedy first fit, greedy best fit, greedy worst fit and meta-heuristic Genetic Algorithm (GA) based method to solve the problem for a realistic network topology.

Simulation results show that the GA based method consistently outperforms the others proposed greedy algorithms and by applying this algorithm, between 32 % to 47 % of the energy can be saved depending on the size and density of the network topology.

Keywords: SDN, Genetic algorithms, Greedy algorithms, Adaptive Link Rate, Optimisation

TABLE DES MATIÈRES

	Page
0.1 INTRODUCTION	1
CHAPITRE 1 REVUE DE LA LITTÉRATURE	11
1.1 Introduction	11
1.2 Axes de recherche du Green Networking	11
1.2.1 Consolidation des ressources	11
1.2.2 Traitement proportionnel à l'énergie	12
1.2.3 Virtualisation des ressources	14
1.2.4 Connectivité sélective	14
1.3 Stratégies du Green Networking	15
1.3.1 Critères de classification	15
1.3.1.1 Délai de prise des décisions	16
1.3.1.2 Champs d'application	17
1.3.1.3 Couche réseau	17
1.3.1.4 Données en entrée	17
1.3.2 Stratégies écoénergétiques : locales	18
1.3.2.1 Adaptation dynamique à la charge de trafic	18
1.3.2.2 Proxy d'interface	22
1.3.3 Stratégies écoénergétiques : globales	23
1.3.3.1 Routage écoconscient	24
1.3.3.2 Applications écoénergétiques	26
1.4 Software defined networks (SDN)	27
1.4.1 Introduction	27
1.4.2 Architecture SDN	29
1.4.3 Économie d'énergie dans les réseaux SDN	30
1.4.3.1 Applications écoénergétiques	31
1.4.3.2 Routage écoconscient	34
1.5 Avantages et inconvénients des stratégies d'économie d'énergie	38
1.6 Conclusion	39
CHAPITRE 2 MÉTHODOLOGIE ET OBJECTIFS	41
2.1 Introduction	41
2.2 Hypothèses du problème	41
2.3 Objectifs	42
2.4 Formulation du problème	44
2.4.1 Modélisation de la consommation énergétique du réseau :	46
2.4.2 Modélisation des reconfigurations de routage :	48
2.5 Solutions proposées	50
2.5.1 Approche formelle	50
2.5.2 Approche heuristique	51
2.5.2.1 Algorithme First Fit (FF)	53

2.5.2.2	Algorithme Best Fit (BF)	55
2.5.2.3	Algorithme Worst Fit (WF)	57
2.5.3	Approche métaheuristique	58
2.5.3.1	Algorithme génétique (AG)	59
2.6	Conclusion	64
CHAPITRE 3 EXPÉRIMENTATION ET RÉSULTATS		65
3.1	Introduction	65
3.2	Protocole expérimental	65
3.3	Scénarios des expérimentations et étude de sensibilité	68
3.3.1	Scénario 1 : mesure des performances des algorithmes pour (K=1)	68
3.3.2	Scénario 2 : mesure du temps de traitement de l'outil Cplex pour l'approche formelle	72
3.3.3	Scénario 3 : mesure de la performances de l'approche métaheuristique pour (K>1)	74
3.4	Cas d'étude	81
3.4.1	Architecture d'implémentation	81
3.4.1.1	Module de collecte des données	81
3.4.1.2	Module d'optimisation	83
3.4.1.3	Module de contrôle	83
3.5	Conclusion	84
CHAPITRE 4 CONCLUSION		85
ANNEXE I ALGORITHME GÉNÉTIQUE (AG)		89
ANNEXE II OUTILS DE SUPERVISION ET CONTRÔLE DU RÉSEAU		99
ANNEXE III ENVIRONNEMENT DE SIMULATION		105
BIBLIOGRAPHIE		117

LISTE DES TABLEAUX

		Page
Tableau 1.1	Critères de classification des stratégies Tirée de Bianzino <i>et al.</i> (2012)	16
Tableau 2.1	Table des notations	45
Tableau 3.1	Modèle de consommation d'énergie Tirée de Tang Jian (2012)	66
Tableau 3.2	Paramètres établis pour l'évaluation de l'AG	69
Tableau 3.3	Taux d'économie d'énergie moyen et maximal des algorithmes Firsts Fit, Best Fit et Worst Fit	69
Tableau 3.4	Taux d'économie d'énergie moyen et maximal de l'AG testé pour une probabilité de mutation = 0,02 et différents nombres de croisements	69
Tableau 3.5	Taux d'économie d'énergie moyen et maximal de l'AG testé avec une probabilité de mutation = 0,05 et différents nombres de croisements	70
Tableau 3.6	Paramètres de variation de la charge de trafic pour les {10, 30, 50} flux générés	72
Tableau 3.7	Nombre d'itérations de l'AG pour différentes matrices de trafic, différents nombre de chemins et différentes topologies (Abilene, NSF)	77

LISTE DES FIGURES

	Page
Figure 0.1	Diagramme des chapitres 9
Figure 1.1	Exemple d'un comportement hebdomadaire du trafic Tirée de Chiaraviglio <i>et al.</i> (2009) 12
Figure 1.2	Différents profils de consommation d'énergie Tirée de Bianzino <i>et al.</i> (2012) 13
Figure 1.3	Link Sleep vs Link Rate Switching 20
Figure 1.4	Mécanismes et politiques de la technique ALR Tirée de Bilal <i>et al.</i> (2013) 21
Figure 1.5	Les paquets dans un bloc sont organisés par destination Tirée de Sergiu <i>et al.</i> (2008) 26
Figure 1.6	Architecture SDN Tirée de ONF (2014) 28
Figure 1.7	EQVMP : Energy-efficient with QoS-aware VM Placement algorithm Tirée de Shao-Heng <i>et al.</i> (2014) 34
Figure 1.8	Banc d'essai : Elastic Tree Tirée de Brandon <i>et al.</i> (2010) 35
Figure 1.9	CARPO : correlation-aware power optimization Adaptée de Xiaodong <i>et al.</i> (2012) 37
Figure 2.1	Exemple de chemins précalculés Tirée de Athanasiou (2013) 43
Figure 2.2	Schéma représentant les différentes méthodes d'optimisation 51
Figure 2.3	Organigramme représentant le trie qui est fait au début des trois algorithmes de l'approche heuristique 52
Figure 2.4	Organigramme représentant l'approche heuristique : First Fit (FF) 54
Figure 2.5	Organigramme représentant l'approche heuristique : Best Fit (BF) 56
Figure 2.6	Organigramme représentant l'approche heuristique : Worst Fit (WF) 57
Figure 2.7	Représentation des gènes et des chromosomes 60
Figure 2.8	Cadre gris : population initiale de la commodité $k=1$ 61

Figure 2.9	Croisement et validation des individus	62
Figure 2.10	Mutation et validation des individus	63
Figure 3.1	Topologie NSF Tirée de nsf.gov (2008)	67
Figure 3.2	Topologie Abilene Tirée de Cs.stanford.edu (2003)	67
Figure 3.3	Temps de traitement moyens des algorithmes Firsts Fit, Best Fit, Worst Fit et AG testés avec différents nombre de flux	70
Figure 3.4	Impact du nombre de flux et de la variation du trafic sur le temps de traitement du Cplex (3 chemins précalculés).....	73
Figure 3.5	Impact du nombre de chemins précalculés sur le temps de traitement du Cplex (3 chemins précalculés).....	73
Figure 3.6	Utilisation maximale et moyenne des liens dans le réseau Abilene Adaptée de Mingui <i>et al.</i> (2010)	75
Figure 3.7	Échantillon de comportement du trafic échangé entre deux paires (O-D) du réseau Abilene	76
Figure 3.8	Nombre de transition des états des liaisons résultantes de l'AG et l'outil Cplex pour différentes matrices de trafic, différents nombres de chemins et différentes topologies (Abilene, NSF)	77
Figure 3.9	Nombre de reroutage de flux résultants de l'AG et l'outil Cplex pour différentes matrices de trafic, différents nombres de chemins et différentes topologies (Abilene, NSF)	78
Figure 3.10	Taux d'économie d'énergie de l'AG et l'outil Cplex pour différentes matrices de trafic, différents nombres de chemins et différentes topologies (Abilene, NSF)	79
Figure 3.11	Temps de traitement de l'AG pour différentes matrices de trafic, différents nombre de chemins et différentes topologies (Abilene, NSF).....	80
Figure 3.12	Architecture proposée	82
Figure 3.13	Génération de la topologie par le contrôleur Opendaylight Tirée de opendaylight.org (2017)	83

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

ALR	Adaptive Link Rate
LS	Link Sleep
LRS	Link Rate Switching
IP	Internet Protocol
QoS	Quality of Service
SDN	Software-Defined Networking
NFV	Network Functions Virtualization
AG	Algorithme Génétique
CPU	Central Processing Unit
ADN	Acide Désoxyribonucléique
VM	Virtual Machine
API	Application Programming Interface
OPEX	Operational Expenditures
CAPEX	Capital Expenditures
DVFS	Dynamic Voltage and Frequency Scaling
ILP	Integer Linear Programming
TCAM	Ternary Content Addressable Memory
TIC	Technologies de l'Information et de la Communication
MPLS	Multi-protocol Label Switching

0.1 INTRODUCTION

Contexte

Le début des années deux mille a vu une croissance explosive dans l'utilisation des équipements des technologies de l'information et de la communication (TIC) dans toutes les sphères de la vie qui incluent l'industrie, le commerce et le secteur résidentiel. Les TIC sont désormais une partie intégrante de la vie quotidienne et les activités humaines y sont fortement dépendantes.

Par le passé, l'intérêt engendré par les TIC était axé sur la performance et le coût. Peu d'attention a été accordée à l'énergie consommée par les équipements et l'impact sur l'environnement. Cependant aujourd'hui, les TIC sont l'un des acteurs majeurs du réchauffement climatique et de la pollution environnementale causés par leur utilisation croissante. En effet, en 2012, la consommation moyenne annuelle d'électricité dans les industries des TIC était de plus de 200 GW où l'infrastructure et les dispositifs de télécommunications représentaient 25 % (Chih-Lin *et al.* (2014)).

Par conséquent, les techniques d'économie d'énergie en général et le concept des réseaux écologiques ou *Green Networking* en particulier attirent une grande attention de l'industrie des télécommunications et de la communauté scientifique. Sur le plan industriel, certaines grandes entreprises du secteur ont déjà entamé une révolution verte par le développement d'équipements réseau « Green » ou écoénergétique. Parmi ces entreprises, telles que Cisco avec sa suite « EnergyWise » (Sandhu *et al.* (2012)), Alcatel- Lucent à travers le consortium mondial « GreenTouch » (Vetter *et al.* (2013)) qui fournit de nouvelles technologies permettant de multiplier par 10 000 l'efficacité énergétique des réseaux mobiles ou encore IBM avec son projet « Big-Green » (Olson (2008)) qui décrit une approche en cinq étapes pour améliorer l'efficacité énergétique ; diagnostiquer, construire, visualiser, gérer, refroidir (cooling).

De nombreuses études sont unanimes pour indiquer que les progrès technologiques au niveau des équipements devraient permettre l'amélioration de l'efficacité énergétique au cours des dix prochaines années. L'intérêt de l'industrie et celui des chercheurs à l'amélioration de l'effi-

capacité énergétique ont permis l'émergence du *Green Networking*. La consommation d'énergie élevée de l'infrastructure de télécommunication, le coût élevé engendré par cette énorme consommation, ainsi que l'empreinte carbone élevée engendrée représentent les principaux facteurs de l'introduction des techniques du *Green Networking*. En effet, des études ont montré que plus de 4,7 % de l'énergie mondiale est consommée par les TIC (Awad *et al.* (2017)). L'empreinte carbone des TIC représente quant à elle jusqu'à 10 % des émissions mondiales de CO₂ (Dabaghi-Zarandi et Movahedi (2017)). Le taux d'augmentation annuelle des émissions de CO₂ engendré par les dispositifs de télécommunication (par exemple, les routeurs, les commutateurs) est estimé à plus de 12% (Wu *et al.* (2017)). Avec une telle augmentation, les prédictions estiment qu'à l'horizon 2020 les dispositifs de télécommunication vont représenter 22 % des émissions de CO₂ à l'échelle mondiale (Wu *et al.* (2017), Dabaghi *et al.* (2017)).

Une grande partie de l'énergie consommée par les TIC est due aux dispositifs physiques utilisés. En effet, ces derniers contribuent à hauteur de 30 % à 37 % des émissions de gaz à effet de serre (GES) produit par le secteur des TIC (Dabaghi *et al.* (2017)). Les auteurs de (Fouli et Maier (2009)) ont observé que les besoins en bande passante doublent tous les 18 mois pour les nouvelles applications réseau. Ces applications entraînent une augmentation de la charge du trafic IP qui se répercute par une augmentation du nombre d'équipements. De plus, les éléments physiques du réseau sont dimensionnés et installés pour gérer le trafic à sa charge crête et doivent être tolérants aux pannes, ce qui entraîne le surdimensionnement et la redondance d'équipements.

Par conséquent, les équipements qui interconnectent l'infrastructure du réseau sont sous utilisés la plupart du temps (Mingui *et al.* (2010)). Par exemple les liens de réseau de communication dorsale ne sont utilisés que de 30 % à 40 % ou moins (Dabaghi *et al.* (2017)). Ainsi, l'utilisation des méthodologies du *Green Networking* permettrait d'économiser une quantité considérable d'énergie en maximisant l'utilisation des ressources réseau (Eyupoglu et Aydin (2015)).

Si l'on adopte un strict point de vue environnemental, l'objectif du *Green Networking* consiste à réduire le volume d'émissions de CO₂ induit par les processus de communication des équipements de l'infrastructure réseau (Khan et Mauri (2013)). L'utilisation de sources d'énergie

renouvelable ou d'électronique à faible consommation constitue des pistes d'amélioration. Ces stratégies peuvent avoir un impact important sur la consommation effective de l'infrastructure, mais n'interviennent que dans une moindre mesure sur les aspects réseau. La délocalisation d'éléments consommateurs, par exemple, impose des contraintes à l'architecture du réseau. Il s'agit essentiellement de planification et d'optimisation.

De nombreuses études ont conclu que même si le matériel technologique devrait permettre l'amélioration de l'efficacité énergétique des réseaux au cours des dix prochaines années, il n'en reste pas moins que cette amélioration sera plus lente par rapport au taux de croissance du trafic (Konstantinos *et al.* (2015)). D'une autre part, la réalisation d'une gestion efficace des ressources en termes d'efficacité énergétique nécessite un contrôle et une gestion plus souple du réseau, étant donné que l'allocation des ressources devient plus dynamique afin d'anticiper une plus grande capacité d'adaptation à de nouveaux services et applications réseau.

Dans cette optique, récemment des initiatives à travers le monde proposent de nouveaux mécanismes permettant l'évolution des réseaux existants vers plus de flexibilité, une meilleure gestion des ressources réseau et une simplification du déploiement des nouveaux services réseau virtualisés (VNF) (Mijumbi *et al.* (2016)). Un de ces mécanismes repose sur une architecture qui sépare le plan de contrôle du plan de transfert des informations. Cette approche est également connue sous le terme (SDN) qui signifie littéralement *Software-Defined Networking* ou réseau défini par logiciel (Hyojoon et Feamster (2013)). Le SDN vise à s'affranchir partiellement de la complexité du réseau physique par le biais d'une couche abstraction qui permet la programmabilité de ses équipements.

En raison de ses avantages, il est intéressant d'étudier la manière dont la technologie SDN peut être utilisée, pour la mise au point d'un mécanisme de conscience énergétique intelligent et centralisé qui offre une gestion efficace de l'énergie consommée par l'infrastructure réseau.

Problématique

La conception des réseaux de communication suivent deux orientations principales. Ils sont généralement surdimensionnés et conçus de manière redondante (Giroire *et al.* (2015)). Le

surdimensionnement est une approche qui permet de prévoir les évolutions du volume de trafic dues à de nouveaux usages et services. Par conséquent, durant les périodes de faible charge, le réseau est actif, mais sous-utilisé et consomme inutilement de l'énergie, même si les profils de trafic sont souvent réguliers et connus. La redondance quant à elle, est nécessaire pour assurer un niveau satisfaisant de fiabilité et de tolérance aux pannes, mais nécessite l'installation d'équipements doublons qui restent en permanence en alerte afin de prendre le relais lorsqu'ils constatent une défaillance.

Ces conditions engendrent une surconsommation énergétique et sont donc opposées aux pratiques environnementales vertes. Leurs répercussions négatives font du *Green Networking* un champ de recherche intéressant et techniquement stimulant.

Dans le contexte du *Green Networking*, il existe de nombreuses pistes d'optimisation de l'énergie au niveau de l'infrastructure réseau ; des applications au routage et au mode de fonctionnement des liaisons de communication, il est possible d'agir à différents niveaux sur une échelle locale ou globale du réseau.

Plusieurs techniques ont proposé le routage écoconscient qui se base sur la technique *Link Sleep* (LS) ou mode inactif du point de vue physique, comme solution écoénergétique pour les réseaux traditionnels. Cependant, leurs performances en matière d'économie d'énergie restent limitées à cause de l'aspect distribué de ces réseaux et l'impact négatif de la technique LS sur la qualité de service (QoS).

La convergence rapide des réseaux a poussé les chercheurs à envisager un contrôle centralisé du réseau en adaptant le paradigme du SDN. De ce fait, le suivi et le réacheminement du trafic peuvent être faits en temps réel. En conséquence, de nouvelles possibilités pour l'amélioration des performances du réseau de manière globale et de l'efficacité énergétique en particulier s'ouvrent.

Afin de proposer un routage écoconscient dans le contexte des réseaux SDN et pour avoir un mécanisme générique d'économie d'énergie agissant en temps réel, ce travail de recherche ré-

pond aux questions suivantes :

Adaptation et Evaluation des approches d'optimisation :

La stratégie du routage écoconscient se base principalement sur des approches d'optimisation. Ces dernières visent à réduire l'énergie du réseau à travers une optimisation de routage des flux actifs dans le réseau. Ainsi, dans l'optique de réduire la consommation énergétique du réseau à travers la stratégie du routage écoconscient, l'adaptation de quelques approches d'optimisation à notre problème s'impose.

Après l'adaptation des approches d'optimisation, une évaluation de ces dernières est nécessaire afin de choisir l'approche qui s'adapte le mieux à notre problème et qui réalise les meilleurs résultats. De ce fait, comment adapter correctement les approches d'optimisation au cas d'étude considéré et quels sont les facteurs à prendre en considération pour évaluer objectivement ces approches ?

Optimisation de le reconfigurations de routage en respectant les contraintes de QoS :

Dans un réseau défini par logiciel (SDN), la configuration de routage (c'est-à-dire les chemins des flux) est appliquée par le contrôleur de réseau. Dans une allocation de flux qui prend en considération l'énergie consommée, la configuration de routage doit être adaptée périodiquement aux volumes de demandes de trafic. Toutefois, l'exécution d'une optimisation de routage en ligne et répétée dans un réseau réel entraîne un grand nombre de reconfigurations de routage dans des intervalles de temps consécutifs. Ces reconfigurations consistent en (i) des transitions d'états des liaisons qui provoquent des pertes de paquets et (ii) le reroutage des flux qui peut affecter l'ordre des arrivées des paquets reçus. De ce fait, les questions suivantes se posent : comment minimiser le nombre de reconfigurations des états des liaisons ainsi que le nombre de reroutages des flux dans notre solution au problème d'économie d'énergie ? Quel est le rôle du SDN dans l'optimisation de routage ?

De plus les réseaux de communication sont généralement surdimensionnés en matière de ressource dans le but de garantir une qualité de service (QoS) qui respecte les termes du contrat (SLA) établi avec le client. De ce fait, comment faire le compromis entre la qualité de service exigée et l'économie d'énergie qui oblige une réduction du dimensionnement des réseaux ?

Complexité du déploiement des applications d'économie d'énergie :

Appliquer une stratégie d'économie d'énergie qui se base sur le reroutage des flux peut affecter les performances du réseau en raison du nombre important de reconfigurations des états d'énergie des liens et de l'oscillation des flux. À quel moment faut-il appliquer les stratégies d'économie d'énergie ? Quels sont les facteurs qui régissent la fréquence d'exécution de ces stratégies ? Sur quelle échelle et à quel type de réseau faut-il appliquer ces stratégies ?

Contributions

Mes contributions dans ce travail sont :

- la modélisation du problème d'économie d'énergie sous les contraintes de qualité de service (QoS), lorsque la technique *Link Rate Switching* est exploitée ;
- le développement et l'adaptation des algorithmes d'optimisations à notre cas d'étude ;
- la simulation, l'analyse des résultats et l'analyse de sensibilité des performances des algorithmes considérés.

Les résultats de ce travail ont fait l'objet d'une publication à soumettre et d'une publication acceptée avec comité de lecture.

Article de conférence accepté :

Samy Zemmouri, Shahin Vakilineia, and Mohamed Cheriet. "Let's adapt to network change : Towards energy saving with rate adaptation in SDN." Network and Service Management (CNSM), 2016 12th International Conference on. IEEE, 2016. (Zemmouri *et al.* (2016))

Article journal à soumettre :

Abdolkhalegh Bayati, Samy Zemmouri and Mohamed Cheriet. "Greening The Network Using Traffic Prediction and Link Rate Adaptation", 2017.

Plan du mémoire

Trois chapitres constituent ce mémoire organisé de la façon suivante :

Le premier chapitre de ce mémoire représente l'introduction de ce mémoire. Il commence par une mise en contexte par rapport au sujet traité et finit par poser les problèmes auxquels va répondre ce mémoire.

Le chapitre 2 présente la revue de littérature. En premier lieu, il introduit les différents axes de recherche du *Green Networking*. Ensuite, il détaille les différentes stratégies qui s'y rattachent et leurs critères de classification dans le cas des réseaux traditionnels et les réseaux SDN. En dernier lieu, il cite les avantages et inconvénients de ces stratégies sont donnés.

Le chapitre 3 se focalise sur la méthode de travail suivie. Il expose dans sa première partie les différentes hypothèses au problème posé. Ces hypothèses aident à mieux cerner ce travail de recherche. Dans sa deuxième partie, il formule et modélise le problème. Finalement, dans la troisième et dernière partie, il résume les différentes approches proposées pour la résolution du problème et qui répondent en partie aux différentes questions et défis présentés dans la partie problématique de l'introduction.

Le chapitre 4 présente l'implémentation de l'environnement expérimental et les résultats relevés pour la comparaison de la performance des algorithmes et à partir des résultats de l'analyse de la performance, il dresse un bilan des performances. Pour finir ce chapitre introduit un cas d'étude qui représente une des implémentations possibles de la solution proposée dans l'environnement SDN.

Le chapitre 5 est une synthèse du mémoire, et une perspective d'un possible travail future.

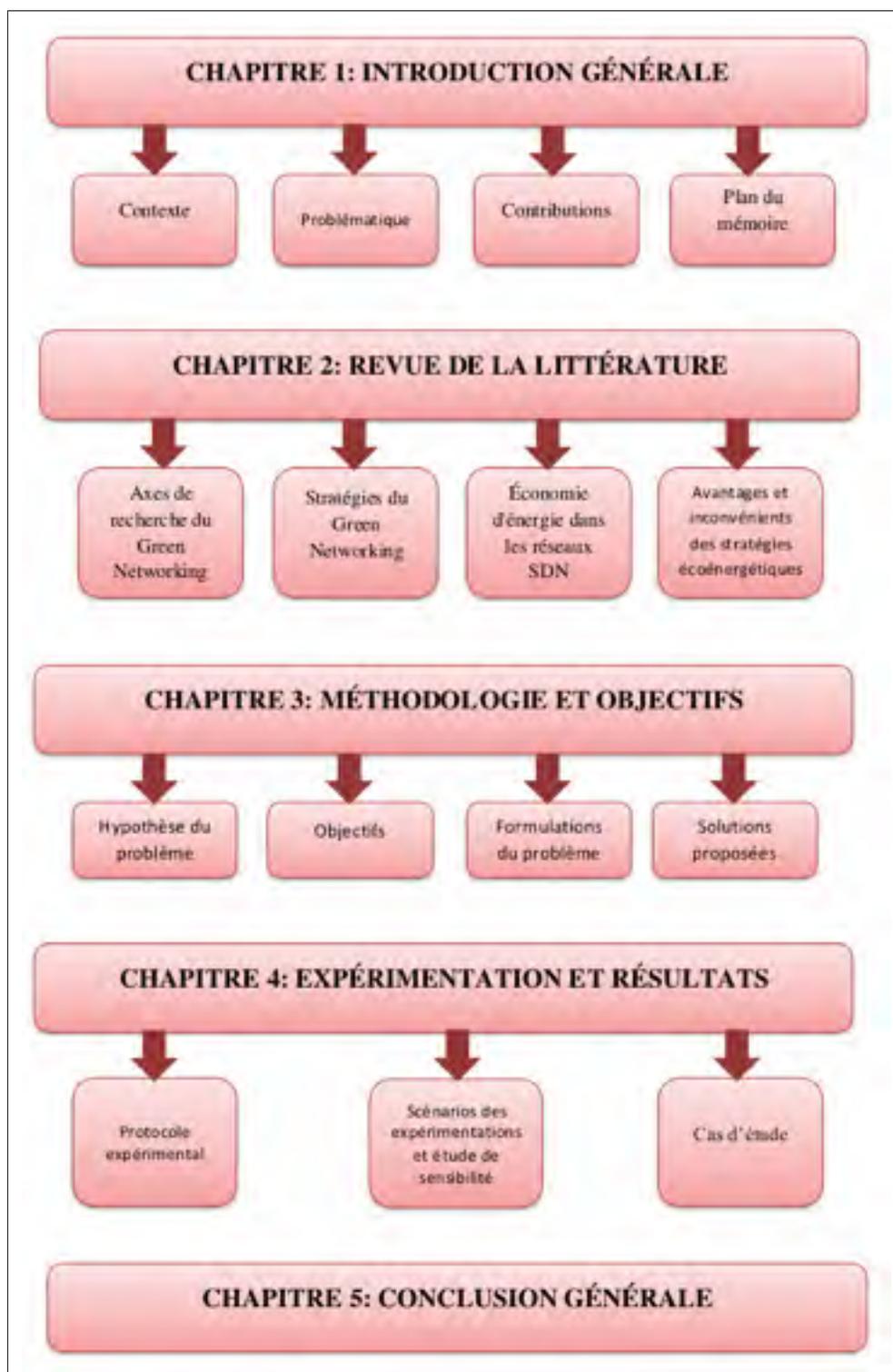


Figure 0.1 Diagramme des chapitres

CHAPITRE 1

REVUE DE LA LITTÉRATURE

1.1 Introduction

Pour rendre le réseau plus écoénergétique, les chercheurs ont mis au point des mécanismes qui permettent d'intégrer l'efficacité énergétique à différents niveaux du réseau tels que les terminaux, les interfaces de communications et les protocoles réseau. Ce chapitre en premier lieu présente les grandes directions de recherche tout en exposant les différents critères de classification établis dans cette optique. Ensuite, il offre une analyse des différentes stratégies et techniques écoénergétiques existantes pour les réseaux traditionnels précédents ainsi que les réseaux de nouvelle génération notamment les réseaux SDN.

1.2 Axes de recherche du Green Networking

Cette section présente axes de recherche du *Green Networking*, c'est-à-dire la consolidation des ressources, la virtualisation des ressources, la connectivité sélective et le traitement proportionnel. Une description plus détaillée est donnée pour chaque axe dans les sections suivantes.

1.2.1 Consolidation des ressources

La consolidation des ressources regroupe toutes les stratégies de dimensionnement qui visent à réduire la consommation énergétique globale en exploitant les dispositifs sous-utilisés à un moment donné. Plus précisément, étant donné que le niveau de trafic suit un comportement quotidien et hebdomadaire bien connu dans un réseau (Vela *et al.* (2016)) (Asfandyar *et al.* (2009)) la figure 1.1 illustre un comportement de trafic bien connu, ce comportement de trafic se traduit par une forte charge de trafic durant la journée et une faible charge durant la soirée. De ce fait, il y a une possibilité d'adapter le niveau du surdimensionnement aux conditions actuelles du réseau. En d'autres termes, le niveau de performance requis sera toujours garanti, mais en utilisant une quantité de ressources qui est dimensionnée pour la demande de trafic

courant du réseau plutôt que pour la demande crête. Le *"traffic grooming"* représente l'une des approches de consolidation des ressources. Cette technique est utilisée pour organiser et simplifier le routage et la commutation dans les réseaux orientés connexion, elle consiste à assembler plusieurs flots de données de bas débit en une seule entité de données qui peut être transportée sur une seule longueur d'onde afin d'améliorer l'utilisation de la bande passante et donc de réduire le coût du réseau. Il est aussi possible d'utiliser traffic grooming de façon à permettre à plusieurs flux des couches supérieures de mutualiser une connexion optique, ce qui permet de réduire le gaspillage des ressources physiques (Yetginer et Rouskas (2009)).

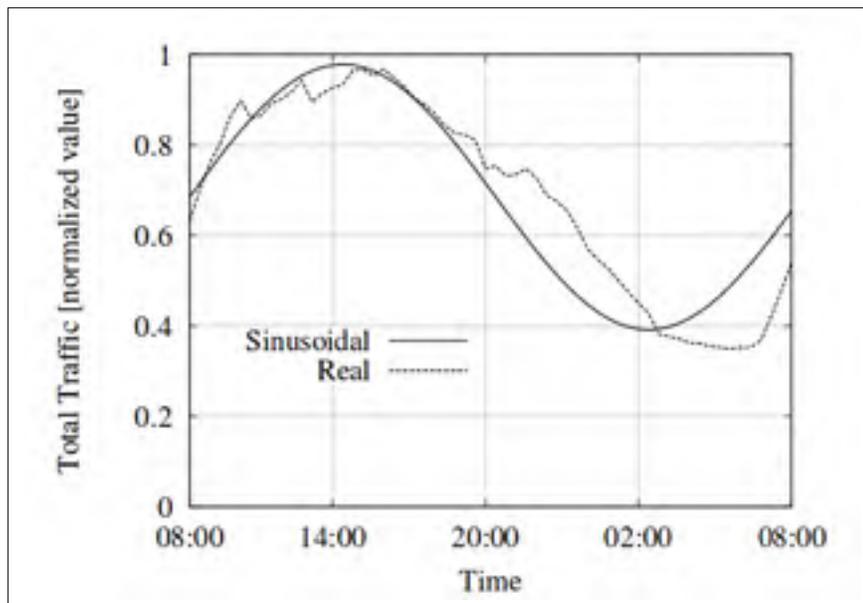


Figure 1.1 Exemple d'un comportement hebdomadaire du trafic
Tirée de Chiaraviglio *et al.* (2009)

1.2.2 Traitement proportionnel à l'énergie

Le traitement proportionnel a été introduit dans (André et Hölzle (2007)). Il peut être appliqué à un système dans son ensemble, aux protocoles réseau, ainsi qu'aux dispositifs et sous-composants individuels d'un dispositif tel que le processeur, la mémoire, les ports des cartes réseau, etc. Pour illustrer ce principe, la figure 1.2 représente différents profils de consommation d'énergie (ou coût) qu'un dispositif peut présenter en fonction de son niveau d'utilisation

en termes de débit de transfert des données, nombre d'opérations à réaliser, quantité de données à stocker, etc. Ces différents profils offrent différentes possibilités d'optimisation. Les dispositifs agnostiques à l'énergie consommée, dont la consommation d'énergie est constante, indépendamment de leur utilisation, représentent le pire des cas : ces dispositifs sont soit allumés et consomment une quantité maximale d'énergie, soit désactivés et consomment peu ou pas d'énergie du tout. Il existe un autre type de profil de consommation énergétique appelé profil proportionnel pour ce type de profil l'énergie consommée est totalement proportionnelle au niveau d'utilisation (Bianzino *et al.* (2012)). Entre ces deux profils extrêmes, il existe un nombre fini de profils intermédiaires qui essaient de se rapprocher le plus possible du profil idéal.

Les techniques nommées *Dynamic Voltage and Frequency Scaling* (Mark *et al.* (1994)) et *Adaptive Link Rate* (Bilal *et al.* (2013)) ou l'adaptation de débit des liaisons sont des exemples typiques de traitement proportionnel. La première réduit l'état d'énergie du CPU en fonction de la charge du système, tandis que la deuxième applique un concept similaire aux interfaces réseau en réduisant leur débit de transmission des données et donc leur consommation, en fonction de la charge de trafic de la liaison.

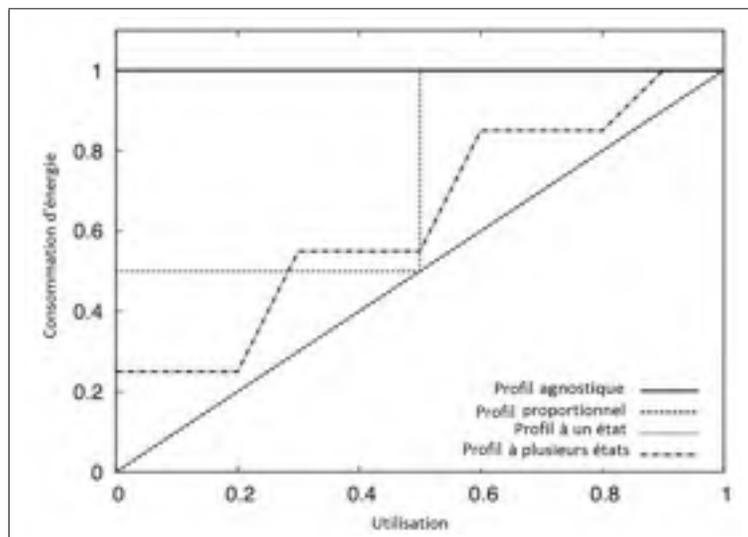


Figure 1.2 Différents profils de consommation d'énergie
Tirée de Bianzino *et al.* (2012)

1.2.3 Virtualisation des ressources

La virtualisation représente un excellent outil pour répondre à la problématique de l'efficacité énergétique. Elle regroupe un ensemble de mécanismes permettant à plusieurs services d'opérer sur la même plateforme matérielle, optimisant ainsi l'utilisation du matériel. La virtualisation peut être appliquée à différentes sortes de ressources, y compris les liaisons d'interconnexion du réseau, les dispositifs de stockage, les ressources logicielles, etc.

Comme ce mémoire cadre avec le projet V-WAN (Virtual Wide Area Network) de l'équipementier de télécommunications Ciena (Ciena, 2014), nous nous sommes intéressés à la virtualisation et au monitoring des réseaux virtuels.

Un exemple de virtualisation est la virtualisation des commutateurs dans les réseaux d'interconnexion des centres de données. Dans ce cas, elle représente la réutilisation d'une seule ressource physique par plusieurs autres commutateurs, dits virtuels ; ou bien, la consolidation de plusieurs de ces ressources physiques pour obtenir un commutateur virtuel plus puissant avec plus de fonctionnalités (Zhao *et al.* (2017)).

Un autre exemple typique de virtualisation consiste à faire fonctionner simultanément plusieurs systèmes d'exploitation (par exemple Linux, Windows, Mac...) qui partagent les ressources d'une seule machine physique. Cela engendre à la fois la réduction des coûts liés au matériel, l'amélioration de l'efficacité énergétique et enfin la réduction de l'empreinte carbone des centres de données d'une manière plus générale. Dans ce contexte, la virtualisation par le passé a déjà été déployée à grande échelle avec succès : par exemple, le Service postal américain a virtualisé 791 de ses 895 serveurs physiques (Richard (2008)).

1.2.4 Connectivité sélective

La connectivité sélective des dispositifs, telle que décrite dans (Christensen *et al.* (2004), Mark *et al.* (2007)), consiste en des mécanismes distribués permettant à chaque équipement du réseau d'être inactif / veille pendant un certain temps, de manière aussi transparente que possible pour

le reste des équipements de l'infrastructure réseau. Si le principe de consolidation s'applique aux ressources qui sont partagées au sein de l'infrastructure réseau, la connectivité sélective quant à elle permet de désactiver des ressources non utilisées. Par exemple, les nœuds d'extrémité du réseau peuvent basculer au mode inactif/veille afin d'éviter les tâches de support à la connectivité, telle que le trafic de diffusion inutile, etc. Ces tâches peuvent être prises en charge par d'autres nœuds, tels que les proxies de sorte qu'aucun changement fondamental n'est nécessaire dans les protocoles réseaux.

1.3 Stratégies du Green Networking

Cette section commence par présenter les critères utilisés pour la classification des différentes stratégies écoénergétiques. Ensuite, en se basant sur ces critères de classification, plusieurs stratégies sont abordées.

1.3.1 Critères de classification

Bien que de nos jours, les équipements et les protocoles de communications sont pour la plupart inconscients de l'énergie qu'ils consomment, un certain nombre de travaux de recherche ont exploré et proposé des techniques qui permettent d'incorporer la conscience énergétique dans les réseaux de communication filaires. Dans cette section, nous présentons différents critères généraux, qui sont résumés dans le tableau 1.1 et qui peuvent nous aider à classer les différentes techniques vertes (écoénergétiques) étudiées jusqu'à présent dans la littérature.

Tableau 1.1 Critères de classification des stratégies
Tirée de Bianzino *et al.* (2012)

Critères	Valeurs	Description
Délais de prise de décisions	En ligne / Hors ligne	Définit la fréquence de mise à jour de la stratégie
Champs d'applications	Locale ou Globale	Influence le volume de communications nécessaires pour atteindre l'objectif
Couche OSI	Liaison de données, réseau, transport, application, ou interaction entre plusieurs couches réseau	Indique quelles entités doivent collaborer
Type de données en entrée	Temps réel, observations historiques, prévisions	Définis la capacité d'apprentissage et d'adaptation de l'algorithme

1.3.1.1 Délai de prise des décisions

Un premier critère important concerne les délais de prise des décisions par la stratégie verte. Comme souligné dans (Christensen *et al.* (2004)), les délais définissent directement le niveau architectural où les actions peuvent être prises : plus le délai est petit et plus la couche est basse, moins l'interaction entre les différents composants du système est possible. Par exemple, au niveau CPU, des échelles temporelles de l'ordre de nanosecondes ou microsecondes peuvent résulter. Ceci est pertinent dans les niveaux d'architecture informatique et logicielle. D'autre part, des échéances de l'ordre de micro aux millisecondes sont plutôt pertinentes pour la couche système. Entre ces échelles de temps, des décisions peuvent être prises entre les paquets consécutifs du même flux (interpaquets, intraflux).

Le critère du délai peut aussi être utilisé pour classer les stratégies d'économie d'énergie, selon qu'elles sont des stratégies qui agissent en ligne ou hors-ligne. La première catégorie se réfère aux stratégies qui agissent au moment de l'exécution (avec des périodes plus courtes qu'une seconde). Les stratégies agissant hors-ligne se réfèrent plutôt à des stratégies dont l'exécu-

tion vient en posteriori aux reconfigurations du réseau qui viennent appliquer les résultats de l'exécution de la stratégie (avec des échelles de temps au-dessus de la seconde).

1.3.1.2 Champs d'application

Selon le type et le nombre d'entités du système impliquées, les solutions en ligne peuvent être classées comme étant soit locales soit globales, dépendamment de la quantité et du type de données nécessaires à la prise de décision. Les stratégies locales s'appliquent à un nœud ou un lien unique à la fois, indépendamment des autres composantes du réseau. D'un autre côté, les stratégies globales s'appliquent à une plus grande échelle et utilisent pour la prise de décision des informations qui se rapportent à un ensemble de nœuds et de liens du réseau.

1.3.1.3 Couche réseau

Comme les réseaux IP sont implémentés suivant un principe de superposition de couches, on peut aussi classer les solutions en fonction de la couche à laquelle elles se rapportent. Compte tenu de la pile de protocoles TCP / IP, chaque solution peut soit impliquer une seule couche parmi les couches ; liaison de données, réseau, transport et application, ou peut nécessiter une interaction entre plusieurs couches réseau.

1.3.1.4 Données en entrée

Un autre critère de classification provient de l'analyse des données en entrée et qui conduisent à la prise de décision. La décision peut être prise sur la base des données représentant la situation en temps réel, sur la base d'une observation historique, ou sur la base d'une prévision (en fonction à la fois des données actuelle et d'observations historiques). Dans le cas des solutions en ligne, les trois types de décisions sont possibles. Dans le cas des solutions hors ligne, les décisions basées sur des données temps réel sont exclues.

Le tableau 1.1 résume les critères précédemment cités et dont nous avons choisi de tirer notre classification des stratégies d'économie d'énergie. Il existe d'autres critères de classification,

notamment le type de service visé. Cependant ils ne sont pas très pertinents dans le contexte de notre recherche, car les techniques écoénergétiques sont généralement applicables dans une mesure assez générale. De ce fait, dans ce qui suit, nous adopterons un point de vue agnostique au type de service.

1.3.2 Stratégies écoénergétiques : locales

Cette section présente les principales stratégies du *Green Networking* exploité d'une manière locale. Autrement dit, qui affecte un équipement ou un sous-composant d'un équipement. Les stratégies locales ne nécessitent pas la connaissance de l'état général du réseau à un moment donné.

1.3.2.1 Adaptation dynamique à la charge de trafic

Jusqu'à présent, la plupart des efforts dans le cadre du *Green networking* ont été consacré à cette stratégie. Les principales techniques qui implémentent cette stratégie suivent en général le paradigme du traitement proportionnel et la consolidation des ressources. Ces techniques sont conçues pour réduire la consommation énergétique en réponse à une faible utilisation ou un faible volume de trafic. Elles peuvent agir localement sur un lien ou globalement sur tout le réseau en même temps dépendamment de la couche du réseau, la dimension du réseau concerné ainsi que de l'existence ou l'inexistence d'interactions entre les éléments du réseau.

Cette stratégie peut agir sur différents composants de l'infrastructure réseau tels que les routeurs, les commutateurs, les liens qui transportent les données, etc. La technique nommée *Adaptive Link Rate (ALR)* est utilisée dans le cas où l'objectif est de réduire la consommation des liaisons. D'autre part, pour les autres composants de l'infrastructure réseau la technique *Dynamic voltage and frequency scaling (DVFS)* est utilisée. La technique DVFS consiste à réduire la fréquence du processeur et donc à réduire la consommation énergétique car la fréquence a un impact quadratique sur l'énergie. Ces actions sur le processeur consiste à utiliser différents couples voltage-fréquence variant selon les processeurs (appelés P-State pour performance state).

Des mesures empiriques ont montré que la consommation d'énergie sur une liaison Ethernet est largement indépendante de son utilisation (Joseph *et al.* (2008b), Helmut *et al.* (2009), Priya *et al.* (2009)). En pratique, même pendant les intervalles de temps où aucune trame n'est transmise, les liaisons sont utilisées pour envoyer continuellement le trafic de contrôle afin de préserver la synchronisation. Par conséquent, la consommation d'énergie d'un lien dépend largement de la vitesse de transfert négociée plutôt que la charge de trafic courante sur la liaison.

Un certain nombre de travaux ont donc proposé d'adapter le débit des liaisons en utilisant la technique ALR. Cette technique agit soit (i) en mettant les liens dans un mode inactif/veille pendant la période de non-transmission, cette technique est connue sous le terme *Link Sleep* ou (ii) en réduisant le débit des liaisons pendant la période de faible transmission comme illustrée dans la figure 1.3. Dans les deux cas, le débit des liaisons est adapté pour correspondre à la charge de trafic réel des liaisons, cette reconfiguration locale des débits permet de rendre la consommation d'énergie pseudo proportionnelle à l'utilisation des liens (Priya *et al.* (2010)). Un sommaire des différents mécanismes et politiques de la technique ALR est illustré dans la figure 1.4.

Link Sleep (LS)

Les travaux tels que (Maruti et Singh (2003), Maruti *et al.* (2004), Helmut *et al.* (2007)), considèrent deux états de fonctionnement : un mode inactif, et un mode entièrement fonctionnel (ou actif). La difficulté dans ce cas consiste à trouver le compromis entre la réactivité du système et l'économie d'énergie. Dans (Maruti et Singh (2003)) qui est considéré comme étant l'un des travaux pionniers ayant utilisé cette technique, les auteurs ont laissé les nœuds réseau décider du statut actif ou inactif de leurs interfaces, en mesurant le temps entre les arrivées des paquets. Il faut noter qu'une vérification est faite pour vérifier si cet intervalle est suffisamment long pour justifier une économie d'énergie entre deux trames consécutives. Comme l'efficacité d'une telle stratégie est directement liée à la distribution temporelle de l'arrivée des paquets, les auteurs ont commencé par analyser une trace du trafic afin de déterminer si une telle approche

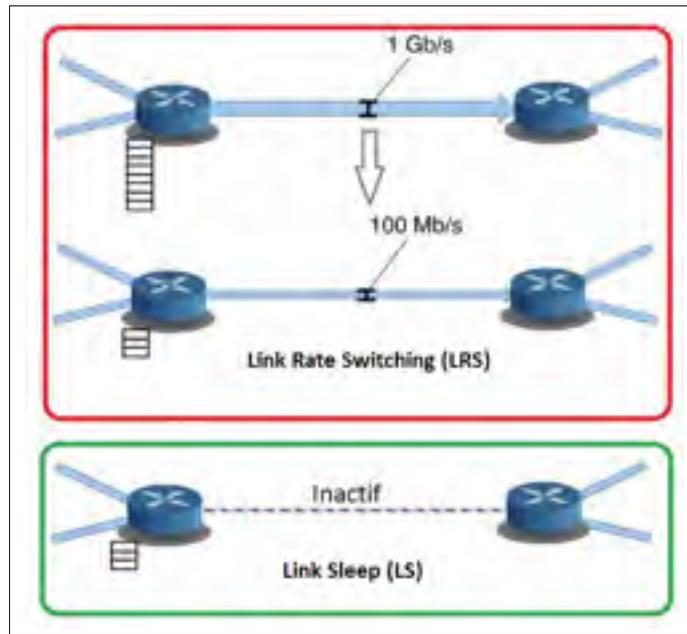


Figure 1.3 Link Sleep vs Link Rate Switching

serait efficace ou non dans la pratique. Mis à part les résultats, ce premier travail soulève un certain nombre de questions, abordées ultérieurement par des chercheurs. Tout d'abord, différents états sont proposés pour une interface. L'interface peut-être dans un état d'inactivité complète et : (i) rejette les paquets qui arrivent au cours de cette période (Van Heddeghem *et al.* (2016)), (ii) basculer au mode actif pour chaque réception de paquet (iii) utilise un tampon pour stocker les paquets reçus au cours des intervalles d'inactivité, et les traiter dans la période active, ou sinon (iv) utiliser un port tiers qui prend en charge la réception des paquets à la place d'un groupe de ports inactifs (Bolla *et al.* (2016)). Cependant, même cet état partiellement inactif a un prix en matière d'énergie. Premièrement, tout état inactif dans lequel les paquets peuvent être détectés a besoin d'un peu d'électronique, et consomme donc une petite quantité d'énergie. Par conséquent, revenir au mode actif à chaque arrivée de paquet réduit la latence et la perte de paquets, mais d'un autre côté ceci réduit également l'économie d'énergie.

Dans (Maruti *et al.* (2004)), les auteurs proposent un modèle à deux états. Le premier état correspond au mode de fonctionnement normal (mode actif), et le second au mode d'économie d'énergie (mode inactif). La première transition, à partir du mode d'économie d'énergie au

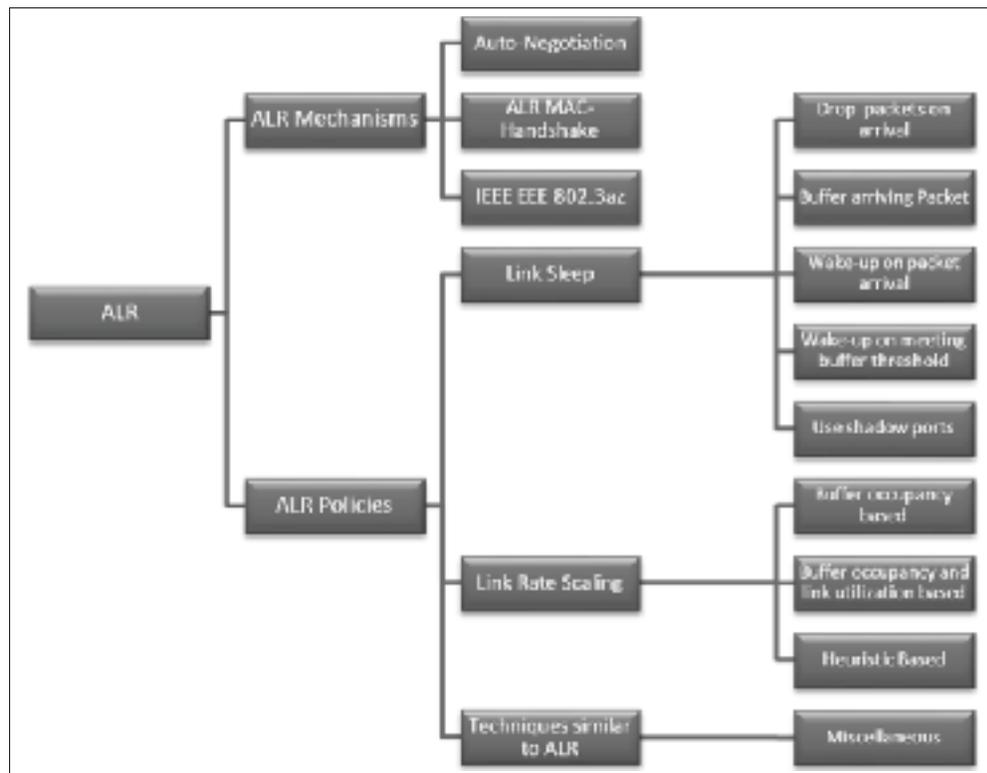


Figure 1.4 Mécanismes et politiques de la technique ALR
Tirée de Bilal *et al.* (2013)

mode actif, prend un délai d'environ 0,1 ms et génère un pic de consommation d'énergie. La deuxième transition, à partir du mode actif vers le mode inactif est censée être instantanée et consomme moins d'énergie. Ainsi, ce comportement peut être étendu pour inclure plus de deux états, ce qui rend possible une modélisation de la stratégie d'adaptation de débit des liaisons.

La norme IEEE 802.3 az (IEEE (2010)) représente une des implémentations de la technique *Link Sleep*. Elle standardise le réseau Ethernet pour qu'il soit plus efficace énergétiquement. Il s'agit d'une standardisation des couches physiques du protocole Ethernet. Le standard implémente des modes de veille (Low Power Idle : LPI) pour différents matériels réseau. Deux types de fonctionnements sont possibles : Ethernet faible vitesse (10Mbps and 100Mbps) et haute vitesse (Low1000BASE-T and 10GBASE-T).

Link Rate Switching (LRS)

Outre le choix entre deux modes extrêmes, mode inactif et mode actif, la plupart des technologies actuelles proposent plusieurs états intermédiaires à travers l'utilisation de plusieurs vitesses de transmission (débits)(Bilal *et al.* (2013), Bolla *et al.* (2016)). À chaque débit correspond un niveau différent de consommation énergétique. La technique Ethernet à l'heure actuelle par exemple, définit plusieurs vitesses de transmission (de 10 Mb/s à 10 Gb /s), cette limite est appelée à augmenter avec l'augmentation de la capacité des liens Ethernet.

Dans (Chamara *et al.* (2008)), les auteurs montrent qu'il existe une différence non négligeable dans l'énergie consommée entre les différents niveaux de vitesses de transmissions établies. Par exemple, pour une carte réseau (NIC) reliée à un PC une augmentation des débits de transfert des données de 10 Mb/s à 1 Gb/s résulte par une augmentation de l'énergie consommée à 3 W, ce qui représente environ 5 % de la consommation globale de la carte réseau (NIC). Les auteurs ont aussi proposé un allègement des politiques de contrôle des débits, pour se baser uniquement sur l'état actuel du système, ou sur une analyse historique. D'un autre point de vue, la sélection des vitesses de transmission appropriées parmi un ensemble limité de possibilités peut être traduite en un problème d'optimisation de flux multicommodités dont l'objectif est de minimiser la consommation globale d'énergie du réseau tout en respectant les contraintes établies telles que la capacité des liens. Ce problème est connu pour être NP-difficile (Bianzino *et al.* (2012)).

1.3.2.2 Proxy d'interface

En ce qui concerne notre classification, l'interface proxy suit le principe de la connectivité sélective et fait partie de l'ensemble des solutions locales / en ligne.

Contrairement au mode inactif, où les fonctionnalités peuvent être simplement désactivées (aucune transmission lorsque le lien est inactif). Dans le cas des dispositifs terminaux, ceci n'est toutefois pas possible, car certaines fonctionnalités doivent être déléguées (le traitement du trafic est transmis à des entités plus écoénergétiques). En effet, même si les utilisateurs sont

inactifs, le trafic réseau en arrière-plan est néanmoins reçu et doit être traité, ce qui empêche les ordinateurs (PCs) de passer au mode inactif. Les auteurs de (Chamara *et al.* (2005), Sergiu *et al.* (2009)) soulignent que la partie majeure du trafic entrant reçu par une interface réseau pendant les périodes d'inactivité peut simplement être ignorée ou ne nécessite qu'une réponse minimale. Par exemple, la plupart des trames de diffusion et le trafic lié au balayage de ports peuvent simplement être ignorés. Les échanges habituels, tels que le traitement ARP, la réponse à l'écho ICMP, sont des tâches simples qui peuvent être facilement exécutées directement par l'interface réseau. Ainsi l'idée derrière le proxy d'interface consiste à déléguer le traitement de ce type de trafic habituellement pris en charge par les CPUs des cartes mères à un certain nombre d'entités différentes à faible énergie comme les processeurs des cartes réseau (NIC) du même dispositif. Dans ce Cas, l'interface réseau (NIC) va gérer le trafic qui nécessite un calcul minimal, tandis que le système complet sera actif uniquement lorsque des paquets non triviaux nécessitant un traitement ultérieur sont reçus. Cela permet l'économie d'énergie par la mise hors tension des systèmes terminaux, sans perturber leur connectivité réseau.

1.3.3 Stratégies écoénergétiques : globales

Les stratégies envisagées jusqu'ici ne concernent que les décisions locales, au niveau d'un seul dispositif ou d'un ensemble très restreint de dispositifs réseau collaboratifs. Bien que ces stratégies offrent des économies d'énergie non négligeables, on peut s'attendre à une amélioration supplémentaire par la collaboration entre les différents dispositifs réseau, qui partagent une connaissance plus large sur l'état du système ou du réseau global. En général, une stratégie globale exploite une ou plusieurs stratégies locales sur l'échelle du réseau. Ce qui permet de prendre en compte les contraintes liées à chaque dispositif.

Avant l'émergence du concept SDN et de son infrastructure, il existait peu de travaux ayant abordé la conscience énergétique d'un point de vue architecture globale, car les réseaux IP traditionnels sont intégrés verticalement, ce qui signifie que les plans de contrôle et de données sont regroupés. Il est donc difficile de configurer le réseau, car tous les commutateurs et routeurs doivent fonctionner selon les mêmes protocoles de contrôle et de réseau de transport. La

modification de certains comportements signifie que tous les périphériques réseau doivent être mis à jour par un administrateur réseau qui prend beaucoup de temps. Il est donc difficile de reconfigurer le réseau pour répondre aux changements qui sont courants, notamment dans les centres de données. Dans ce qui suit, on présente les principales stratégies globales du *Green Networking*.

1.3.3.1 Routage écoconscient

Suivant le principe de consolidation des ressources et dépendamment des capacités physiques des éléments réseaux a adapté la consommation d'énergétique à la charge de trafic transportée, le routage écoconscient vise généralement soit à agréger les flux de trafics sur un sous-ensemble d'équipements et liaisons réseau, ce qui permet à d'autres liaisons et dispositifs d'interconnexion de basculer vers un mode inactif, soit à rerouter une partie du trafic de manière à optimiser l'utilisation des liaisons et des équipements et permettre à ces dernières de transiter vers un état moins énergivore. Ces solutions devraient préserver la connectivité et la QoS, par exemple en limitant l'utilisation maximale sur n'importe quel lien ou en assurant un niveau minimum de diversité de chemin. D'une manière formelle, le routage écoconscient est un cas particulier du problème général de flux multicommodités (Luca *et al.* (2009)), et se classe ainsi dans l'ensemble des solutions hors ligne.

Dans (Luca *et al.* (2009)), le problème est formulé à l'aide de la programmation linéaire en nombres entiers (ILP), et certaines heuristiques qui visent à éteindre progressivement les nœuds et les liaisons réseau. Les auteurs étudient les effets de différentes stratégies de sélection sur l'ensemble des nœuds et des liens. Le problème est géré au niveau des réseaux ISP, avec des nœuds d'extrémité multi-homed. D'autres travaux, tels que (Will *et al.* (2010), Prem *et al.* (2010)), résolvent le problème ILP numériquement, en considérant que seuls les liens peuvent être désactivés. Contrairement à (Prem *et al.* (2010)) où les auteurs considèrent que les liens et nœuds tous les deux peuvent être désactivés.

Dans (Lin *et al.* (2013)) les auteurs ont proposé d'intégrer la technique LRS aux réseaux de centres de données. Avec des dispositifs réseau qui supportent la technique LRS, les auteurs

ont tenté d'approcher la proportionnalité de l'énergie du réseau par rapport au trafic transporté à travers l'optimisation de routage. Afin d'éviter le réordonnement des paquets, les auteurs supposent que toutes les demandes sont routées d'une manière non fragmentée. Ceux qui veut dire que flux suit un seul chemin. Ils ont commencé par la formalisation du problème, ensuite ils ont proposé un algorithme d'approximation efficace (TSRR). Les simulations faites sur un ensemble de topologies des centres de données (FatTree, BCube, DCell) montrent qu'en intégrant l'adaptation des taux aux réseaux de centres de données, l'optimisation globale du routage du réseau peut permettre d'économiser jusqu'à 40% d'énergie, même sans éteindre les périphériques réseau complètement.

De la même manière que le travail précédent, les auteurs dans (Tang Jian (2012)) exploitent la technique LRS dans les réseaux filaires en étudiant le problème d'allocation de flux : Étant donné un ensemble de chemins candidats pour chaque demande de communication de bout en bout, ils tentent de déterminer quel chemin peut être alloué pour transporter le flux afin de minimiser la consommation d'énergie, sous réserve de satisfaire la demande de trafic de chaque flux. Les auteurs considèrent une fonction ascendante discrète pour la consommation d'énergie des liaisons. Ils abordent le cas des sessions de communication unique et multiple et les formulent comme deux problèmes d'optimisation. À savoir l'allocation de flux pour les sessions uniques (SF-RAP), et l'allocation de flux multisession (MF-RAP). Les auteurs démontrent d'abord que les deux problèmes sont NP-complet et présentent une formulation en Programmation linéaire multicommodites pour le cas multisessions (MF-RAP) pour fournir des solutions optimales. Ils présentent ensuite un algorithme d'approximations pour le cas (SF-RAP) et une heuristique ainsi qu'un framework pour le cas (MF-RAP). Des simulations sur le réseau de recherche Abilene et le réseau NSF ont été effectuées. Les résultats de la modélisation montrent que l'algorithme proposé pour le cas (SF-RAP) surpasse la solution de base basée sur le chemin le plus court et que l'algorithme proposé pour le cas (MF-RAP) fournit une solution proche de l'optimale.

1.3.3.2 Applications écoénergétiques

Les applications écoénergétiques représentent toutes les stratégies globales qui exécutent des applications afin de réduire l'énergie consommée par des éléments de l'infrastructure réseau et qui n'affectent pas le routage. Dans (Sergiu *et al.* (2008)) les auteurs proposent une solution globale où le trafic est remodelé en bloc au niveau du routeur d'entrée du réseau, en disposant tous les paquets destinés au même routeur de sortie de manière contiguë dans les blocs Figure (1.5). Cette approche ajoute un délai à l'entrée du réseau. Ils considèrent l'adaptation automatique des vitesses de transmission des liaisons d'un point de vue global d'un réseau dorsal. En conséquence, les périodes d'activité et d'inactivité s'alternent moins souvent par rapport à une simple stratégie de *Link Sleep*. Ce travail étudie aussi l'incidence sur le temps d'inactivité pour différents paramètres : l'utilisation moyenne du réseau, la taille des blocs et le temps de transition entre les états. Les auteurs soutiennent que cette stratégie n'ajoute pas de complexité significative au réseau, bien qu'ils ne proposent pas de stratégies pour déterminer quand et pendant combien de temps les nœuds devraient idéalement être inactifs. De plus, les effets de la remodelisation du trafic sur la gigue ne sont pas analysés, même s'ils devraient être importants.

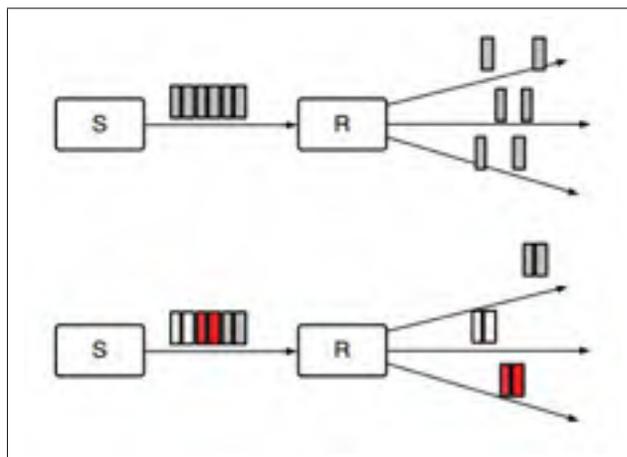


Figure 1.5 Les paquets dans un bloc sont organisés par destination
Tirée de Sergiu *et al.* (2008)

D'autre part, l'introduction de la conscience énergétique dans le processus de conception de l'architecture réseau est étudiée dans (Joseph *et al.* (2008a), Brunilde et Mellah (2009)). Plus en détail (Joseph *et al.* (2008a)) introduit le coût de la consommation d'énergie dans la formulation multicommodité du problème de conception, ainsi que les contraintes de performance et de robustesse. Une approche similaire est adoptée dans (Brunilde et Mellah (2009)), qui évalue également le compromis entre la consommation d'énergie et la performance du réseau, mettant en évidence les caractéristiques de tolérance aux pannes des différents scénarios possibles.

1.4 Software defined networks (SDN)

Du point de vue de l'économie d'énergie, les sections précédentes ont abordé les principales stratégies du *Green Networking* ainsi que les techniques qui s'y rattachent pour le cas des réseaux traditionnels. En revanche cette section abordera le problème d'économie d'énergie pour le cas des réseaux de nouvelle génération, notamment les réseaux SDN. Pour ce faire une brève introduction aux réseaux SDN et leurs caractéristiques sont données. Ensuite on exposera un sommaire des différentes stratégies d'économie d'énergie induites par ses caractéristiques.

1.4.1 Introduction

Le principe du SDN est la séparation du plan de contrôle des éléments du réseau. Ce concept présente plusieurs avantages. Le plan de contrôle peut être un programme logiciel exécuté sur le matériel de base. Ceci sépare le plan de contrôle des dispositifs monolithiques et permet le développement et l'adaptation du plan de contrôle sans avoir besoin de nouveau matériel. D'autre part, le plan de données est maintenant interchangeable et peut-être constitué de matériel construit à partir de composants standard.

Ceci est possible uniquement grâce à l'existence des interfaces de programmation applicatives ouvertes entre le contrôle et le plan de données. Cette interface est souvent appelée «South-bound API». L'implémentation la plus répandue de cette interface est le protocole OpenFlow (McKeown et Turner (2008)). Il fournit un ensemble de messages standard qui permettent à un plan de commande externe de faire fonctionner ses éléments de réseau connectés via un ges-

tionnaire de réseau. En plus du contrôleur, les fonctionnalités du réseau sont exécutées en tant que modules. Ces modules sont programmables librement et peuvent être modifiés et combinés selon les exigences du réseau sur lequel ils sont exécutés.

De plus, ces modules de contrôle peuvent communiquer avec des applications exécutées au-dessus ou en conjonction avec le contrôleur du réseau, par exemple un logiciel d'orchestration dans le nuage, pour optimiser le réseau en fonction des besoins des applications. Cette interface est appelée «*Northbound-API*». La figure 1.6 illustre les composants de l'architecture SDN.

D'après l'ONF (SDN (2013b)) (Open Networking Fondation), d'un point de vue technique, le SDN agit au niveau d'un plan de contrôle agrégé et centralisé qui pourrait être une solution prometteuse pour les problèmes de gestion et de contrôle réseau. De plus le protocole openFlow constitue un élément fondamental, indispensable à la conception de solutions SDN.»

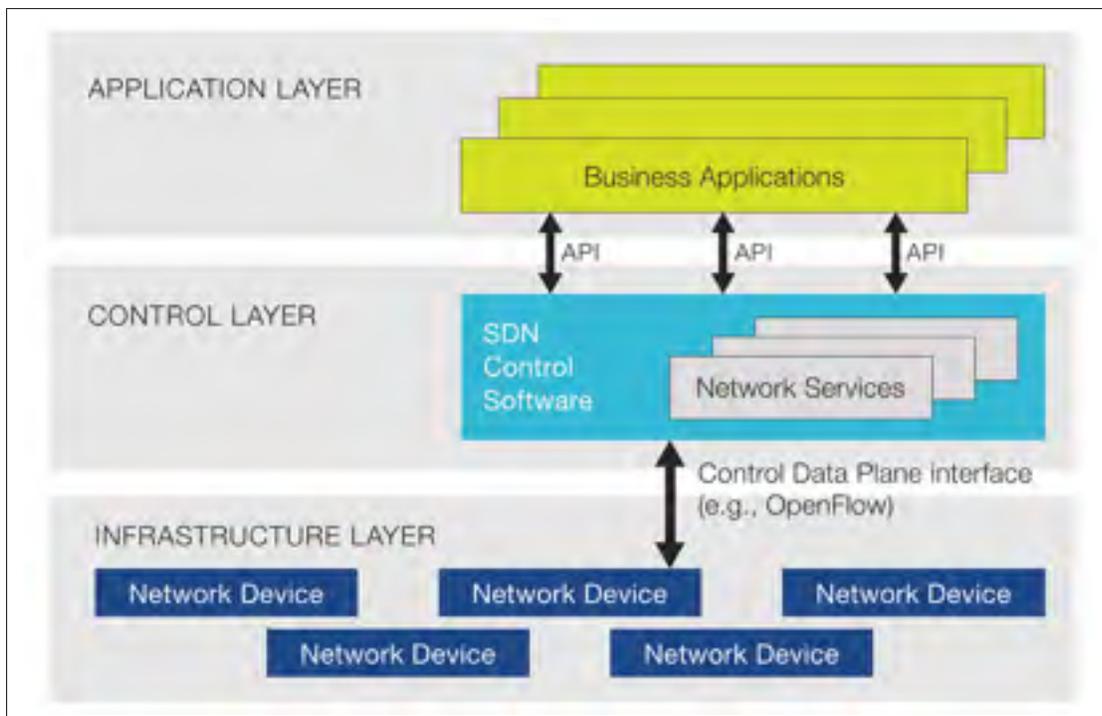


Figure 1.6 Architecture SDN
Tirée de ONF (2014)

1.4.2 Architecture SDN

Dans l'architecture SDN, le fractionnement des fonctions de contrôle et de transfert de données est appelé «désagrégation», car ces pièces peuvent être achetées séparément, plutôt que déployées comme un seul système intégré. Cette architecture donne aux applications plus d'informations sur l'état de l'ensemble du réseau à partir du contrôleur, par opposition aux réseaux traditionnels où le réseau est sensible aux applications.

Les architectures SDN ont généralement trois composantes ou groupes de fonctionnalités :

Applications SDN : Les applications SDN sont des programmes qui communiquent les comportements et les ressources nécessaires avec le contrôleur SDN via l'interface de programmation d'application API. En outre, les applications peuvent construire une vue abstraite du réseau en collectant des informations à partir du contrôleur à des fins de prise de décision. Ces applications pourraient inclure la gestion de réseau, l'analyse, ou des applications d'affaires utilisées pour exécuter de grands centres de données. Par exemple, une application d'analyse peut être créée pour reconnaître l'activité réseau suspecte pour des raisons de sécurité.

Contrôleur SDN : Le contrôleur est le cœur d'un réseau SDN. Il se trouve entre les équipements réseau qui représentent la couche physique et les applications réseaux qui représentent la couche applicative c'est une entité logique qui reçoit des instructions ou des exigences de la couche application SDN et les relaie aux composants réseau. Le contrôleur extrait également des informations sur le réseau à partir des équipements et les communique aux applications SDN via une vue abstraite du réseau. Ainsi il facilite la gestion automatisée du réseau et facilite l'intégration et l'administration des applications.

Dispositifs de réseau SDN : Les dispositifs de réseau SDN contrôlent les capacités de transmission et de traitement de données pour le réseau. Cela inclut l'envoi et le traitement du chemin de données.

1.4.3 Économie d'énergie dans les réseaux SDN

La réalisation des opérations réseau économes en énergie nécessite un contrôle et une gestion de réseau plus souples, puisque l'allocation des ressources devient plus dynamique en anticipant un degré plus élevé d'adaptabilité pour les nouvelles applications et les services réseau. Dans ce contexte, la mise en réseau logiciel (SDN) (SDN (2013a)) et la virtualisation des fonctions réseau (NFV) (NFV (2016)) sont des solutions viables pour renforcer rapidement et souplement la capacité du réseau, assurant une connectivité entre les réseaux et les nuages informatiques. Non seulement cela, mais comme les réseaux sont utilisés pour un nombre croissant de tâches de complexité variable, tout cela doit être fait de manière plus efficace et plus rentable. D'autre part, le développement d'applications SDN écoénergétiques exige l'existence de fonctionnalités spécifiques à l'intérieur des dispositifs du réseau.

Les caractéristiques de l'architecture SDN citées dans la section précédente permettent au SDN d'être déployées dans un ensemble diversifié de plates-formes, allant des réseaux domestiques aux réseaux de centres de données, offrant ainsi plusieurs avantages tels que la flexibilité, une efficacité élevée grâce à un routage optimisé, une facilité de mise en œuvre et d'administration et une réduction des coûts Opex et Capex.

L'optimisation énergétique peut être appliquée sur différents composants de l'architecture SDN et peut-être adressée par application ou par des améliorations matérielles. Les solutions basées sur le matériel sont appliquées au niveau des commutateurs. Les solutions logicielles sont appliquées au niveau du contrôleur. D'après la littérature la plupart des stratégies écoénergétiques appliquées aux réseaux traditionnels sont applicables pour les réseaux SDN. D'un autre côté de nouvelles stratégies relatives à la technique SDN et ces avantages ont émergé. Nous classons les stratégies d'efficacité énergétique les plus récentes et les plus déployés pour les réseaux SDN en deux grandes catégories, à savoir la détection du trafic, le compactage du TCAM, l'optimisation de placement des règles de flux et la connaissance de l'hôte final.

1.4.3.1 Applications écoénergétiques

L'optimisation du placement des règles de flux :

Une des applications écoénergétiques les plus exploitées dans les réseaux SDN est l'optimisation du placement des règles de flux qui se concentre sur comment et où placer les règles de flux dans la mémoire TCAM. Cette dernière est coûteuse, très gourmande en consommation électrique et limitée en matière de capacité de stockage. Ainsi l'optimisation des règles de flux minimise l'espace utilisé sur la mémoire TCAM et minimise par ce fait sa consommation en matière de consommation d'énergie. Cette technique en général ne nécessite pas la mise en place d'une nouvelle infrastructure. Les techniques d'efficacité énergétique qui procèdent par l'optimisation du placement des règles de flux commencent par la formalisation du modèle de coût énergétique et les contraintes associées. Puisqu'en général l'optimisation du placement des règles est un problème NP-complet, une technique heuristique est appliquée pour trouver une stratégie optimale d'économie d'énergie, bien que les approches heuristiques ne garantissent pas des solutions optimales, elles offrent généralement des résultats proches des résultats optimaux en fonction des contraintes (Yossi *et al.* (2013)), (Nanxi *et al.* (2013)), (Frédéric *et al.* (2014)). Des règles de transfert de flux sont générées ensuite et envoyées aux commutateurs à travers le contrôleur.

- **Palette distribution**

C'est une approche distribuée appliquée aux tables des commutateurs (Yossi *et al.* (2013)). Étant donné que la table des contrôleurs SDN ne peut traiter que des centaines d'entrées et que la mémoire est coûteuse et énergivore. Le framework Palette proposé, permet d'équilibrer la taille des tables sur le réseau, ainsi que de réduire le nombre total d'entrées en partageant les ressources entre les différentes connexions. Il gère deux problèmes d'optimisation NP-complet : décomposer une grande table en sous-tables équivalentes et distribuer les sous-tables. La mise en œuvre de palette est basée sur des algorithmes de formulation de théorie de graphe et des heuristiques et les résultats obtenus sont proches de l'optimale en pratique.

La compression de la mémoire TCAM :

Cette solution tente de minimiser le besoin de mémoire des informations stockées dans les commutateurs. Dans l'architecture SDN, les commutateurs utilisent le type de mémoire TCAM, qui est un type spécialisé de mémoire à grande vitesse. Cette dernière effectue une recherche sur toute la mémoire dans un seul cycle d'horloge. Cependant, la mémoire TCAM est très coûteuse et énergivore. Une stratégie optimale en coût et en énergie appliquée à la mémoire TCAM peut être obtenue en compactant la TCAM elle-même où en compressant les informations qui y sont stockées (Kannan et Subhasis (2013)).

Les types de compression qui peuvent être appliqués dans la TCAM sont la compression des règles et la compression du contenu. Dans une liste de contrôle d'accès traditionnel, une règle comporte cinq composantes : plage source, plage de destination, protocole, ports et action. Dans l'architecture SDN, la décision d'acheminement d'un commutateur est basée sur des tables de flux implémentées dans la TCAM.

- **Rectilinear**

«Rectilinear» (Luo *et al.* (2014)) est une approche qui exploite les fonctionnalités de l'architecture SDN telle que l'interface de programmation ouverte aux commutateurs et la détermination dynamique des actions pour chaque flux au niveau des commutateurs. La compression réduit la taille des bits pour stocker les informations qui sont essentielles pour classer les paquets en flux. Un identifiant est donné à chaque flux pour identifier de façon unique les paquets dans le flux correspondant. Les en-têtes de paquets sont modifiés au niveau des commutateurs d'acheminement pour transporter l'identifiant de flux qui peut être utilisé par d'autres commutateurs sur le chemin pour classer les paquets. Les auteurs ont démontré que la représentation compressée des flux peut réduire la consommation d'énergie de la TCAM de 80% en moyenne.

- **Bit Weaving**

La technique de «bit Weaving» utilise un schéma de compression sans préfixe (Meiners *et al.* (2012)) et est basée sur l'observation des entrées TCAM qui ont la même décision, mais dont les prédicats diffèrent d'un seul bit peuvent être fusionnés en une seule entrée en remplaçant le bit en question par *. Le «bit Weaving» utilise deux nouvelles approches, l'échange de bits et la fusion de bits. Les avantages clés de cette technique sont la rapidité d'exécution, l'efficacité et peut-être complémentaire à d'autres méthodes d'optimisation TCAM en tant que routine de pré / post-traitement.

La virtualisation :

Les solutions d'économie d'énergie utilisent la pratique consistant à désactiver les serveurs physiques sous-utilisés et à exécuter leurs tâches sur un nombre restreint de serveurs dans les centres de données SDN (Shao-Heng *et al.* (2014)). Spécifiquement dans les centres de données, le modèle SDN est utilisé pour former une superposition reliant des machines virtuelles. La virtualisation de serveurs aide les systèmes à exécuter plusieurs systèmes d'exploitation et services sur une seule machine physique, ce qui diminue la consommation d'électricité par rapport à l'exécution de serveurs physiques sous-utilisés. Ainsi, au lieu d'exploiter de nombreux serveurs à faible utilisation, la technique de virtualisation combine la puissance de traitement sur un nombre inférieur de serveurs qui auront une utilisation totale plus élevée.

- **Honeyguide**

«Honeyguide» est une topologie de migration de machines virtuelles pour l'efficacité énergétique dans les réseaux de centres de données (Hiroki *et al.* (2013)) ou deux techniques sont combinées : optimisation du placement des machines virtuelles (VM) et consolidation de trafic.

- **EQVMP**

La solution EQVMP propose un placement de machines virtuelles économes en énergie et qui répond aux exigences de la QoS pour les centres de données SDN (Shao-Heng *et al.* (2014)) Figure 1.7. Contrairement à ElasticTree, la mise en activité et le basculement au mode inac-

tif sont appliqués aux serveurs et non pas à l'infrastructure réseau. L'EQVMP combine trois techniques : réduction des sauts, économie d'énergie et équilibrage de charge. La réduction du saut divise les VM en groupes et réduit la charge de trafic entre les groupes par partitionnement graphique. Les économies d'énergie sont principalement réalisées par le placement de VM. Fat-tree est utilisé pour représenter la machine virtuelle et les serveurs dans le centre de données. Le SDN est utilisée pour équilibrer la charge dans le réseau. L'équilibrage de charge réalise la transmission de flux dans les réseaux sans congestion.

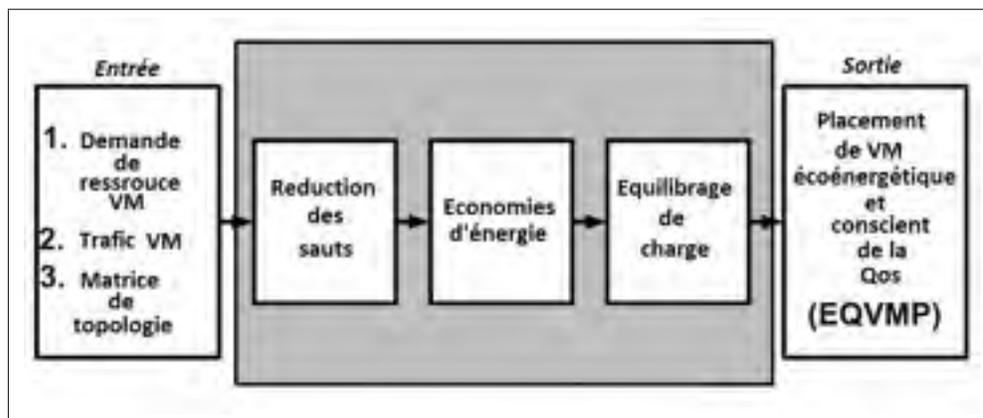


Figure 1.7 EQVMP : Energy-efficient with QoS-aware VM Placement algorithm
Tirée de Shao-Heng *et al.* (2014)

1.4.3.2 Routage écoconscient

Comme vu précédemment, la plupart des approches qui exploitent le routage à des fins d'économie d'énergie sont inspirées par le fait que les composants du réseau sont souvent sous-utilisés. Le principal défi consiste à déterminer lesquels des composants (ports, CPU, commutateurs) qui doivent être désactivés et réactiver sans compromettre la qualité de service requise (QoS). L'utilisation de ces techniques n'est possible qu'avec la connaissance du trafic à travers un échantillonnage à des intervalles de temps bien définis.

- **Elastic Tree**

Dans (Brandon *et al.* (2010)) les auteurs proposent le framework «Elastic Tree» pour la gestion de l'énergie des réseaux de centres de données illustrées dans la Figure 1.8. L'idée est de rendre la consommation d'énergie du réseau quasi proportionnel au trafic qui change d'une manière dynamique. Pour ce faire les auteurs commencent par comparer trois stratégies d'optimisation basée sur les algorithmes suivants : la résolution par Cplex du modèle formel, l'algorithme glouton et enfin une heuristique. Chaque algorithme d'optimisation prend en entrée la topologie du réseau (un graphe), les contraintes de routage, un modèle d'énergie (agnostique au volume de trafic), la matrice de trafic, et génère un sous-ensemble du réseau (liens, commutateurs). La fonction objective minimise la somme totale de commutateurs et liaison activés. L'avantage du modèle est qu'il garantit une solution dans un optimum facilement configurable n'affectant pas beaucoup les performances du réseau grâce au nombre de transitions (actifs / inactifs) réduites des éléments du réseau, cependant, il ne s'échelonne que jusqu'à 1000 hôtes.

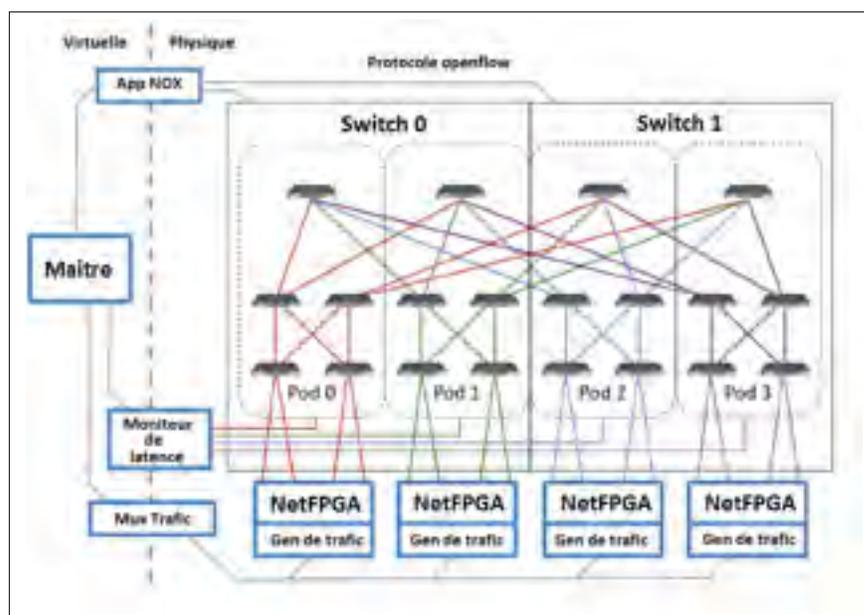


Figure 1.8 Banc d'essai : Elastic Tree
Tirée de Brandon *et al.* (2010)

L'algorithme glouton quant à lui évalue les chemins possibles pour chaque flux et assigne ce dernier au chemin le moins chargé. L'algorithme améliore l'évolutivité du modèle formel. Cette approche souffre du même problème que n'importe quelle technique gloutonne. Cependant, les solutions basées sur les algorithmes gloutons peuvent être calculées de façon incrémentielle et en raison de leur rapide convergence ils sont considérés comme faisant partie des solutions qui agissent en ligne.

L'algorithme heuristique d'autre part, divise les flux tout en cherchant un sous-ensemble de liens minimums pour acheminer tous les flux actifs. Cette heuristique est aussi plus efficace du point de vue du temps de calcul. Mais présente un inconvénient relatif à la dégradation des performances en raison de la mise en activité et la désactivation fréquentes des éléments du réseau.

Les auteurs examinent les compromis entre l'efficacité énergétique, la performance et la robustesse pour chaque stratégie, avec des traces de trafic réelles extraites d'un site Web de production de commerce électronique. L'heuristique proposée pour le calcul des sous-ensembles de réseau permet au framework de s'échelonner sur des centres de données contenant des milliers de nœuds. En matière d'économie d'énergie, les résultats démontrent que, pour les charges de travail des centres de données, le framework ElasticTree peut économiser jusqu'à 50% de l'énergie du réseau.

- **CARPO**

Dans (Xiaodong *et al.* (2012)) les auteurs optent pour une autre approche. En effet dans ce travail, les auteurs proposent un algorithme d'optimisation basé sur la corrélation des flux. Cette solution est appliquée aux réseaux de centre de données et illustrée dans la Figure 1.9. Le but étant de regrouper dynamiquement les flux de trafic sur un petit ensemble de liaisons et de commutateurs, puis désactiver les éléments réseau non utilisés pour économiser l'énergie. Contrairement aux travaux existants, CARPO est conçue sur la base de l'analyse des traces de trafic réelles des centres de données. De cette analyse les auteurs ont observé que les flux n'ont pas tous une charge crête exactement au même moment. Par conséquent, ils ont pris en compte

les corrélations entre les différents flux durant la consolidation du trafic et l'intègrent avec la technique d'adaptation des débits des liaisons, pour économiser plus d'énergie. Les résultats obtenus démontrent que l'algorithme CARPO peut économiser jusqu'à 46% de l'énergie du réseau pour un centre de données.

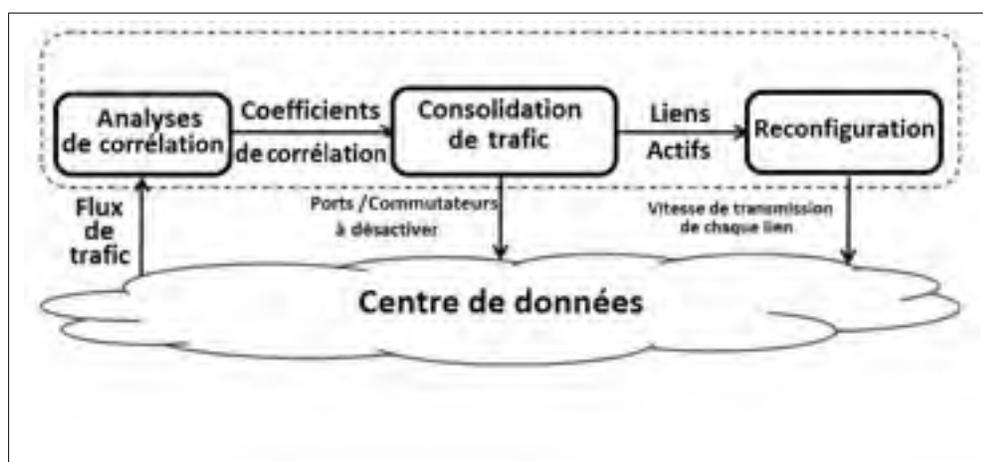


Figure 1.9 CARPO : correlation-aware power optimization
Adaptée de Xiaodong *et al.* (2012)

- **REsPoNse**

C'est un framework proposé dans (Nedeljko *et al.* (2011)), il permet aux opérateurs de réseau d'identifier automatiquement les chemins critiques en termes d'énergie. Il étudie la possibilité de précalculer quelques chemins énergivores qui, lorsqu'ils sont utilisés de manière efficace, peuvent produire en permanence des économies d'énergie proches de l'optimale sur de longues périodes de temps. REsPoNse procède par l'analysant des matrices de trafic, les installe dans un petit nombre de tables de routage et utilise un mécanisme d'ingénierie de trafic simple et évolutif pour désactiver et activer des éléments de réseau sur demande. Les résultats obtenus à partir du framework REsPoNse sont issus de matrices de trafic réelles. Ces derniers démontrent que le framework réalise pratiquement les mêmes performances en termes d'économies d'énergie que les approches existantes, avec un impact marginal sur l'évolutivité du réseau et les performances des applications.

1.5 Avantages et inconvénients des stratégies d'économie d'énergie

Les techniques écoénergétiques locales c'est-à-dire impliquant des dispositifs seuls ou des très petits jeux de dispositifs collaboratifs, offrent une économie d'énergie non négligeable. Cependant, on peut s'attendre à une plus grande économie d'énergie par l'utilisation des techniques globales qui utilisent une connaissance plus large de l'état du réseau.

Les commutateurs utilisent la mémoire TCAM. Cette dernière est coûteuse et très énergivore. Certaines techniques ont tenté de compresser les règles de flux stockées dans la TCAM. Un des plus importants inconvénients de telles solutions est qu'elles sont limitées par un seuil qui une fois atteint limite la compression des informations stockées dans la TCAM.

Le placement des règles affecte directement les performances du réseau notamment le routage. Compte tenu d'une politique de routage, il est nécessaire de disposer d'un moyen formel et plus efficace de représenter les règles des flux.

Parmi les techniques écoénergétiques prenant en charge le trafic, le routage écoconscient utilise la technique LS ou la technique LRS en plus de la distribution des flux à travers le réseau afin de déterminer le sous-ensemble de composants réseaux à activer ou désactiver ou à reconfigurer. Toutefois, la détermination dynamique des sous-ensembles de composants réseau à activer ou désactiver sans affecter la QoS et la performance est en général un problème NP-complet. Une solution efficace dans ce domaine devrait prendre en compte le compromis entre les économies d'énergie et la performance du réseau ainsi que l'évolutivité et la flexibilité du système à adapter dynamiquement les différentes configurations du réseau.

Plusieurs problèmes peuvent se poser en cas de désactivation complète des commutateurs ou des liaisons. En effet d'après une étude faite par les auteurs de (Brandon *et al.* (2010)), où ils ont mesuré le temps nécessaire pour allumer/éteindre les liens et les commutateurs des équipements réels. Ils ont trouvé que le temps pour le démarrage d'un commutateur varie de 30 secondes pour le commutateur Quanta à environ 3 minutes pour le commutateur HP. L'activation et la désactivation des ports prennent environ 1 à 3 secondes et remplir la table de

flux d'un coup prend moins de 5 secondes. Parmi les effets indésirables engendrés par ces délais on retrouve l'oscillation de la route et la non convergence du réseau. Pendant la phase de convergence, le réseau peut subir une dégradation de la performance due à des chemins temporaires inviabilisés. De plus, le délai nécessaire pour activer / désactiver le composant réseau peut augmenter le temps d'attente pour l'optimisation de tous les flux du réseau et peut également réduire la réactivité des nouveaux flux et par conséquent engendrer de la latence.

La technique LRS permet de s'affranchir des dégradations que peut causer la technique LS tout en ayant les mêmes avantages. En effet, la technique LRS ne nécessite pas la désactivation complète du port, ce qui permet au flux de contrôle de transiter normalement dans le réseau et d'éviter par conséquent les problèmes de connectivité. De plus, la technique LRS peut être appliquée plus souvent et d'une manière continue sur le réseau. En effet, étant donné que le délai de transition entre les états des liaisons est moins important que le délai engendré par l'activation / désactivation des liaisons (Sergiu *et al.* (2008)), ceci rend l'utilisation de la technique LRS moins critique sur les performances du réseau. Enfin, la technique LRS n'utilise pas ou peu les mémoires tampons contrairement à la technique LS (Reviriego *et al.* (2010)), ce qui veut dire une meilleure optimisation des ressources.

Malgré les avantages de la technique LRS par rapport à la technique LS, néanmoins, elle présente une faiblesse causée par l'absence de mécanismes efficaces pour la commutation rapide des débits des liaisons. En effet, la reconfiguration des débits engendre des délais qui provoquent la perte de paquets à cause de chemins temporairement inviabilisés.

1.6 Conclusion

les réseaux IP traditionnels sont complexes et très difficiles à gérer. D'un autre côté, Le nouveau paradigme SDN permet la programmation des équipements réseau ce qui permet de faire une abstraction entre l'infrastructure et les services réseau. Cette nouvelle fonctionnalité apporte plus de flexibilité, une meilleure gestion des ressources réseau et représente une grande opportunité pour améliorer les performances du réseau en général et l'efficacité énergétique

en particulier en profitant de la vue globale fournie du réseau pour développer un mécanisme centralisé écoénergétique.

CHAPITRE 2

MÉTHODOLOGIE ET OBJECTIFS

2.1 Introduction

Au cours de ce chapitre, nous allons essayer de répondre aux problématiques établies dans la section 1.3. Après avoir présenté, classer, analyser et comparer les stratégies d'économie d'énergies existantes dans le contexte du *Green Networking* pour les réseaux traditionnels et les réseaux de nouvelle génération basés sur le paradigme du SDN. Nous avons pu dans la conclusion du chapitre 1 déterminer les points forts et les faiblesses de chacune des stratégies. Partant de ce point, dans le présent chapitre, nous nous expliquerons sous quel angle nous avons choisi d'attaquer le problème d'économie d'énergie. Nous établirons des hypothèses. Ensuite, nous présenterons en détail les approches proposées pour la résolution de ce problème en exploitant les points forts des stratégies existantes.

2.2 Hypothèses du problème

Un rôle clé est joué par la technologie SDN dans le contrôle et la coordination de centaines de nœuds du réseau. Ces nœuds doivent être reconfigurés à la volée afin d'optimiser l'utilisation et la qualité de service (QoS), compte tenu de l'évolution rapide des flux de trafic. Parmi les actions coordonnées qui peuvent être prises, on peut citer l'activation, la désactivation et la reconfiguration des vitesses de transmission des composants du réseau peu utilisés, y compris les liaisons et les commutateurs. Ces actions engendrent toujours un reroutage des flux existants afin de répondre conjointement aux exigences d'économie d'énergie et de qualité de service (ex : reroutage de flux dans le but d'éviter la congestion et réduire la latence). De ce fait, il existe une relation directe entre le choix de placement de flux et la quantité d'énergie consommée. De ce point de vue, le problème d'optimisation de placements des flux peut être étendu pour contenir le problème d'économie d'énergie.

La solution au problème d'optimisation de placement des flux est une nouvelle distribution qui satisfait la fonction objectif fixée. Une fois la nouvelle distribution optimisée de flux trouvée, un reroutage des flux est nécessaire pour concrétiser cette distribution. De ce fait, dans notre travail le routage écoconscient est choisi comme stratégie d'économie d'énergie. Avant de présenter nos algorithmes d'optimisation de placement de flux, on pose les hypothèses suivantes :

- en raison du délai important nécessaire pour activer / désactiver les nœuds et qui peut varier d'un équipement à un autre. Ce travail se focalisera uniquement sur l'économie d'énergie au niveau des liaisons ;
- pour limiter les effets négatifs de la technique LS qui affectant la performance du réseau et permettre à notre approche d'être exécutée sur n'importe quel type de réseau. Notre approche sera basée sur la stratégie LRS au lieu de la stratégie LS ;
- en l'absence d'équipements réseau ayant la capacité matérielle de prendre en charge la gestion de l'énergie des liaisons, nous posons l'hypothèse que la transition entre les états d'énergie des liaisons (reconfiguration des débits) est faite automatiquement par l'équipement ;
- le trafic est réparti entre les chemins disponibles à la granularité d'un flux, ce qui veut dire que les paquets de chaque flux suivent le même chemin. En effet, bien que splitter les flux puisse se révéler avantageux, cette technique est typiquement indésirable en raison de l'effet de réordonnement de paquets à la réception au niveau de la couche de transport ;
- nous considérons l'existence de chemins précalculés entre chaque nœud source et destination (figure 2.1). Plus précisément, pour chaque session de communication de bout en bout, un protocole de routage standard peut être utilisé pour trouver un ensemble de chemins candidats et ainsi les algorithmes de placement de flux proposés peuvent être exécutés pour répartir la charge de trafic sur ces chemins de telle sorte que l'énergie soit minimisée.

2.3 Objectifs

Le principal objectif du mémoire consistera à proposer un algorithme de placement dynamique des flux afin de minimiser l'utilisation des liaisons actives (trafic total sur un lien actif divisé

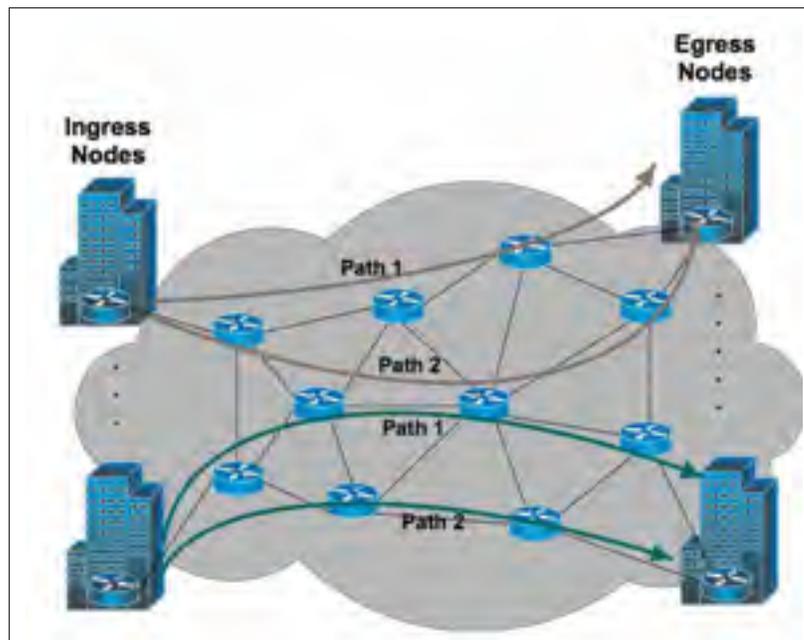


Figure 2.1 Exemple de chemins précalculés
Tirée de Athanasiou (2013)

par la capacité de la liaison) dans le but de réduire la consommation d'énergie dans un réseau SDN.

De manière plus générale et en se basant sur les critères de classification des stratégies d'économie d'énergie présentées dans le tableau 2.1, notre objectif sera de proposer une technique d'économie d'énergie qui agira en ligne, elle aura un champ d'application globale sur le réseau, agira principalement à la couche liaison/réseau et recevra en entrée des données en temps réel.

Cet objective se traduit en termes des sous-objectifs suivants :

- étudier le problème de la distribution des flux dans les réseaux denses à large échelle ;
- modéliser la consommation énergétique du lien en fonction de sa charge de trafic ;
- prendre en compte la contrainte du nombre de reconfigurations des états des liaisons ;
- prendre en compte la contrainte du nombre de reroutage des flux ;
- proposer une modélisation (ILP) au problème d'économie d'énergie qui prend en charge les contraintes établies ;

- résoudre les problèmes avec l’outil Cplex pour montrer l’aspect NP-difficile du problème ;
- proposer, tester et implémenter un algorithme qui redistribue les flux dans le réseau en vue de réduire la consommation d’énergie en un minimum de temps ;
- proposer un cas d’étude dans l’environnement SDN de la solution proposée.

2.4 Formulation du problème

Dans notre formulation, le réseau est décrit comme un graphe $G(V, E)$ où V est l’ensemble des nœuds et E est l’ensemble des liens ou chaque l est utilisé pour la communication dans les deux sens (duplex). Le tableau 2.1 montre les notations qui ont été utilisées pour la formulation de notre problème d’optimisation. Étant donné qu’on se base sur la technique LRS, chaque lien peut fonctionner sur différents états. C’est-à-dire que le lien l fonctionne dans l’un des S_l états et la consommation d’énergie du lien dans l’état s sera la constante e_s . En général, les liens consomment moins d’énergie lorsqu’ils fonctionnent à des vitesses de transmission basses (c.-à-d. $e_{s-1} < e_s \Leftrightarrow c_{s-1} < c_s$ où c_s et e_s sont respectivement la capacité et la consommation d’énergie du lien à l’état s).

Tableau 2.1 Table des notations

E	Ensemble des liens dans le réseau
V	Ensemble des nœuds dans le réseau
K	Ensemble des commodités dans le réseau
N_k	Ensemble des flux de la commodité k
S_l	Ensemble des états des liaisons
e_s	Consommation d'énergie d'un lien l à l'état s
c_s	Capacité d'un lien l à l'état s
P_k	Ensemble de chemins précalculés pour la commodité k
p	Un chemin précalculé parmi la liste des chemins P_k
d_k	Montant de la demande de trafic pour la commodité k
$d_{k,n}$	Montant de la demande du flux n de la commodité k
f_l	Charge de trafic d'un lien l
$f_{k,p,n}$	Charge de trafic du flux n de la demande k routée à travers le chemin p_k ,
$E(f_l)$	Consommation d'énergie d'un lien l en fonction de sa charge de trafic f_l
<i>Commodite</i>	Un nœud origine (O) et un nœud destination (D); eg $:(O_1,D_2) \Leftrightarrow (D_2,O_1)$

La fonction croissante et discrète $E(f_l)$ (2.1) représente la consommation d'énergie du lien l en fonction de sa charge de trafic f_l .

$$E(f_l) = \begin{cases} e_1 & \text{if } 0 \leq f_l < c_1 \\ e_2 & c_1 < f_l < c_2 \\ \dots & \\ e_s & c_{s-1} < f_l < c_s \\ \dots & \\ e_{s_\ell} & c_{s_\ell-1} < f_l \leq c_{s_\ell} \end{cases} \quad (2.1)$$

L'objectif est de minimiser la consommation d'énergie globale du réseau tout en préservant la qualité de service (QoS). Le contrôleur de réseau SDN alloue les flux aux chemins disponibles dans une commodité k . P_k représente l'ensemble des chemins précalculés pour la commodité k . Le montant de la demande de trafic pour la commodité k est représenté par d_k , p est sélectionné pour supporter la demande de trafic des flux n de la commodité k ou chaque demande de trafic d'un flux est représenté par $d_{k,n}$.

Le processus d'allocation de flux exécute un algorithme d'optimisation formulé comme un problème ILP pour minimiser la consommation d'énergie dans l'ensemble du réseau. Les entrées de ce processus d'optimisation comprennent les listes des chemins précalculés P_k pour chaque commodité k , les capacités maximales c_{s_l} et la liste des vitesses de transmission établies pour les liaisons, et enfin la matrice de trafic. La sortie consiste en une nouvelle distribution des flux sur les chemins p_k ainsi que les débits des liaisons qui y correspondent. L'optimisation ILP est formulée dans ce qui suit :

2.4.1 Modélisation de la consommation énergétique du réseau :

La consommation d'énergie totale du réseau est égale à : $\sum_{\forall l \in E} \sum_{\forall s \in S_l} e_s y_l^s$.

Dans l'optimisation ci-dessous, la fonction objective (2.2) représente la somme de la consommation d'énergie de tous les liens du réseau.

$$\min \sum_{\forall l \in E} \sum_{\forall s \in S_l} e_s y_l^s \quad (2.2)$$

L'équation (2.1) n'est pas une conversion linéaire. Ainsi, il est nécessaire d'ajouter des variables binaires supplémentaires complémentaires, à savoir y_l^s telles que :

$$y_l^s = \begin{cases} 1 & \text{Si le lien } l \text{ fonctionne dans l'état } s \\ 0 & \text{sinon} \end{cases} \quad (2.3)$$

la variable de contrôle $x_{k,p,n}$ définit si un flux donné n est bien transporté par un chemin p .

$$x_{k,p,n} = \begin{cases} 1 & \text{Si le chemin } p \in P_k \text{ transporte le flux } n \in N_k \\ 0 & \text{sinon} \end{cases} \quad (2.4)$$

La contrainte (3.5) indique qu'un seul chemin doit être sélectionné pour chaque flux $n \in N_k$.

$$\sum_{\forall p \in P_k} x_{k,p,n} = 1 ; \quad x_{k,p,n} \in \{0,1\}, \forall k \in K, \forall n \in N_k, \quad (2.5)$$

Les contraintes (3.6) et (3.7) déterminent les états appropriés des liens.

$$\sum_{s=1}^{S_l} y_l^s = 1 ; \quad \forall l \in E \quad (2.6)$$

$$y_l^s \leq 1 + \frac{c_s - f_l}{c_{S_l}} ; \quad y_l^s \in \{0,1\}, \forall s \in S_l, \forall l \in E \quad (2.7)$$

La contrainte (3.8) s'assure que toutes les demandes de trafic des N_k flux de la commodité k sont satisfaits.

$$f_{k,p,n} = x_{k,p,n} d_{k,n} ; \quad \forall p \in P_k, \forall k \in K, \forall n \in N_k \quad (2.8)$$

La variable f_l (2.9) représente la quantité totale de trafic transportée par la liaison l , qui ne peut dépasser la capacité de liaison maximale c_{S_l} (la capacité de la liaison l travaillant à son état le plus élevé). Cette dernière est assurée par la contrainte (2.10). La valeur constante doit être correctement sélectionnée pour éviter la congestion. Afin de limiter la congestion dans le réseau la capacité des liaisons est limitée à 90 % de la capacité maximale ($u_l = 0.9$).

$$f_l = \sum_{\forall k \in K} \sum_{\forall p \in P_k} \sum_{\forall n \in N_k} f_{k,p,n} ; \quad \forall l \in E \quad (2.9)$$

$$f_l \leq u_l c_{S_l} ; \quad \forall l \in E \quad (2.10)$$

2.4.2 Modélisation des reconfigurations de routage :

Dans un réseau défini par logiciel (SDN), la configuration de routage (c'est-à-dire les chemins des flux et états de la liaison) est appliquée par le contrôleur de réseau. Dans une allocation de flux prenant en considération l'énergie, la configuration de routage doit être adaptée périodiquement aux volumes de demandes de trafic. Toutefois, l'exécution d'une optimisation de routage répétée dans un réseau réel entraîne un grand nombre de reconfigurations dans le réseau dans des intervalles de temps consécutifs. Ces reconfigurations consistent en (i) des transitions d'états d'énergie des liaisons qui provoquent des pertes de paquets, et (ii) un reroutage des flux qui peut affecter l'ordonnement des paquets et provoquer du délai à la réception. Afin de pouvoir contrôler l'impact de l'optimisation de routage répété sur la performance du réseau. Nous avons modifié notre fonction objective qui devient :

$$\min \| E_s Y + R + Q \|_2 \quad (2.11)$$

Dans cette formulation, $E_s = \{e_s^t\}$, $Y = \{y_l^{s,t}\}$, $R = \{R_l^{s,t}\}$, et $Q = \{Q_l^{s,t}\}$ sont quatre vecteurs représentant les états énergétiques des liaisons sélectionnés par Y , les transitions d'états des liaisons et les reroutages des flux respectivement. L'indice $t - 1$ représente l'état précédent. Il est utilisé par les variables se référant aux données en entrée. D'autre part, l'indice t représente l'état présent.

Dans la nouvelle formulation, les contraintes sont :

Les variables de contrôle binaires (3.11), (3.12) s'assurent respectivement qu'à l'instant présent t , un flux donné n est bien transporté par un chemin p de la commodité k et qu'un lien l fonctionne bien dans un état donné s .

$$x_{k,p,n}^{s,t} = \begin{cases} 1 & \text{Si le chemin } p \in P_k \text{ transporte le flux } n \in N_k \text{ au moment } t \\ 0 & \text{sinon} \end{cases} \quad (2.12)$$

$$y_l^{s,t} = \begin{cases} 1 & \text{Si le lien } l \text{ fonctionne dans l'état } s \text{ au moment } t \\ 0 & \text{sinon} \end{cases} \quad (2.13)$$

La contrainte (3.13) s'assure que toutes les demandes de trafic des N_k flux de la commodité k sont satisfaits à l'instant t .

$$f_{k,p,n}^t = x_{k,p,n}^t d_{k,n}^t ; \quad \forall p \in P_k, \forall k \in \mathcal{K}, \forall n \in N_k \quad (2.14)$$

La variable f_l définit dans la formulation (3.14) représente la quantité totale de trafic transportée par la liaison l , qui ne peut dépasser la capacité de liaison maximale définit par la variable c_{S_l} dans (3.15).

$$f_l^t = \sum_{\forall k \in \mathcal{K}} \sum_{\forall p: l \in p} \sum_{\forall n \in N_k} f_{k,p,n}^t ; \quad \forall l \in E \quad (2.15)$$

$$f_l^t \leq u_l c_{S_l} ; \quad \forall l \in E \quad (2.16)$$

Les contraintes (3.16), (3.17) déterminent les états d'énergie des liaisons à l'instant t .

$$y_l^{s,t} \leq 1 + \frac{c_s - f_l^t}{c_{S_l}} ; \quad \forall s \in S_l, \forall l \in E \quad (2.17)$$

$$\sum_{s=1}^{S_l} y_l^{s,t} = 1 ; \quad \forall l \in E \quad (2.18)$$

La variable R_l^s représente le changement d'état énergétique des liaisons. À partir de cette variable on peut calculer le vecteur R qui représente le nombre de liaisons ayant changé d'états.

$$R_l^s = \begin{cases} 1 & \text{si } y_l^{s,t} \neq y_l^{s,t-1} \\ 0 & \text{sinon} \end{cases} ; \quad \forall l \in E, \forall s \in S_l \quad (2.19)$$

La variable Q_l^s représente le reroutage d'un flux n sur un autre chemin de la commodité k . À partir de cette variable on peut calculer le vecteur Q qui représente le nombre de flux rerouté.

$$Q_l^s = \begin{cases} 1 & \text{si } x_{k,p,n}^{s,t} \neq x_{k,p,n}^{s,t-1} \\ 0 & \text{sinon} \end{cases} ; \quad \forall p \in P_k, \forall k \in \mathcal{K} \quad (2.20)$$

2.5 Solutions proposées

Comme cela est présenté dans la section précédente, le problème de placement des flux prenant en considération la minimisation de la consommation d'énergie et le nombre de reconfigurations dans le réseau peut être formulé comme un problème d'optimisation (ILP) ayant des variables entières. Malheureusement, la résolution de l'ILP n'est pas viable, puisqu'elle tombe dans la classe des problèmes connus sous l'appellation *multi-commodity flow*. Ces problèmes sont connus pour être NP-difficile (Chiaraviglio *et al.* (2009)). Des solutions exactes ne peuvent être trouvées que pour les petits réseaux et pour certains cas triviaux, car le temps de traitement requis pour le calcul des solutions augmente rapidement avec l'augmentation de la taille des topologies. Étant donné que la solution proposée est une solution qui agit en temps réel (en ligne), le temps de traitement devient un facteur aussi important que le taux d'économie d'énergie et le nombre de reconfigurations dans le choix de l'algorithme d'optimisation à adopter. Pour surmonter la complexité de la recherche d'une solution optimale, certains algorithmes d'optimisations ont été étudiés et évalués. La figure 2.2 illustre quelques algorithmes utilisés dans le cadre de l'optimisation. Les approches encadrées en rouge représentent les approches utilisées dans ce travail.

2.5.1 Approche formelle

Le problème d'optimisation décrit dans la section (2.4) peut être résolu par un logiciel d'optimisation tel que CPLEX (IBM (2009)). Cependant, le temps de calcul tend à augmenter de façon exponentielle avec la taille du réseau. Dans un réseau à grande échelle, la résolution du problème d'optimisation n'est pas applicable. Le contrôleur SDN doit être capable de réagir

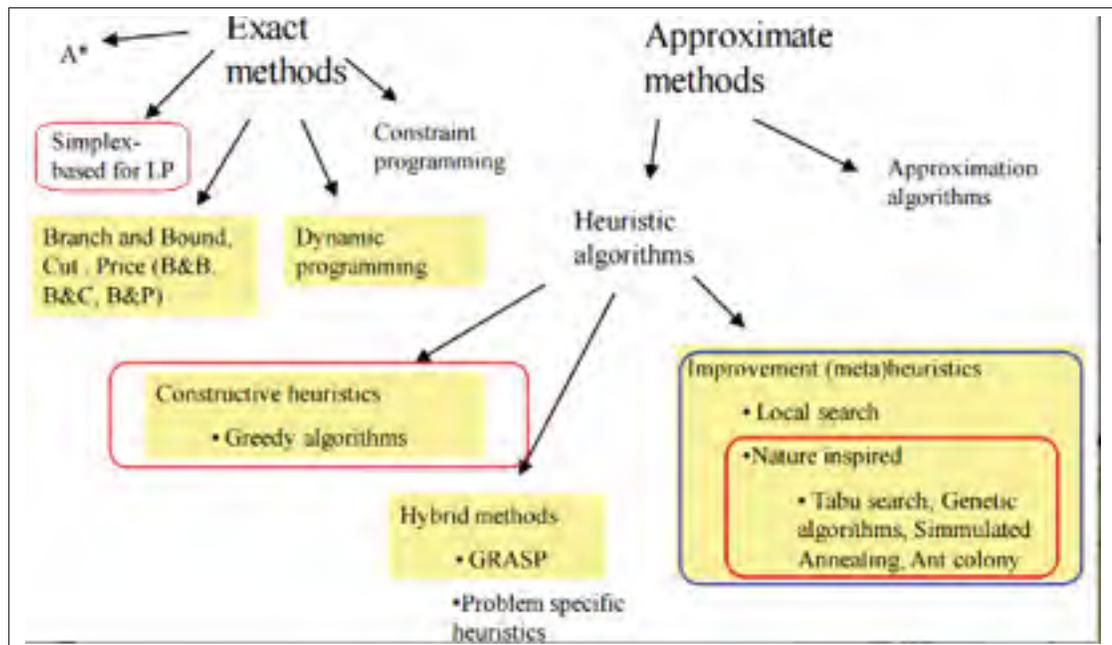


Figure 2.2 Schéma représentant les différentes méthodes d'optimisation

instantanément aux changements de trafic et de rediriger rapidement les flux en conséquence pour éviter la congestion du réseau. Cela étant dit, pour démontrer l'efficacité de notre approche en termes d'économie d'énergie et temps d'exécution par rapport à l'approche formel, on a soumis nos modèles à une résolution par l'outil CPLEX.

2.5.2 Approche heuristique

Afin de résoudre le problème d'économie d'énergie, l'optimisation de placement des flux est choisie comme une stratégie globale d'économie d'énergie en exploitant la technique LRS. Pour résoudre ce problème d'optimisation, un ensemble d'heuristiques ont été utilisés en premier lieu. Le but de ces heuristiques est de trouver une meilleure solution locale. Cependant, ils ne sont pas capables de trouver nécessairement une solution globale optimale. Par ailleurs, ces algorithmes ont une faible complexité d'implémentation et ont un temps d'exécution polynomial. Ce qui représente un grand atout pour le cas des stratégies qui s'exécutent en temps réel (en ligne). La partie commune entre les 3 algorithmes de l'approche heuristique est présentée dans l'organigramme 2.3.

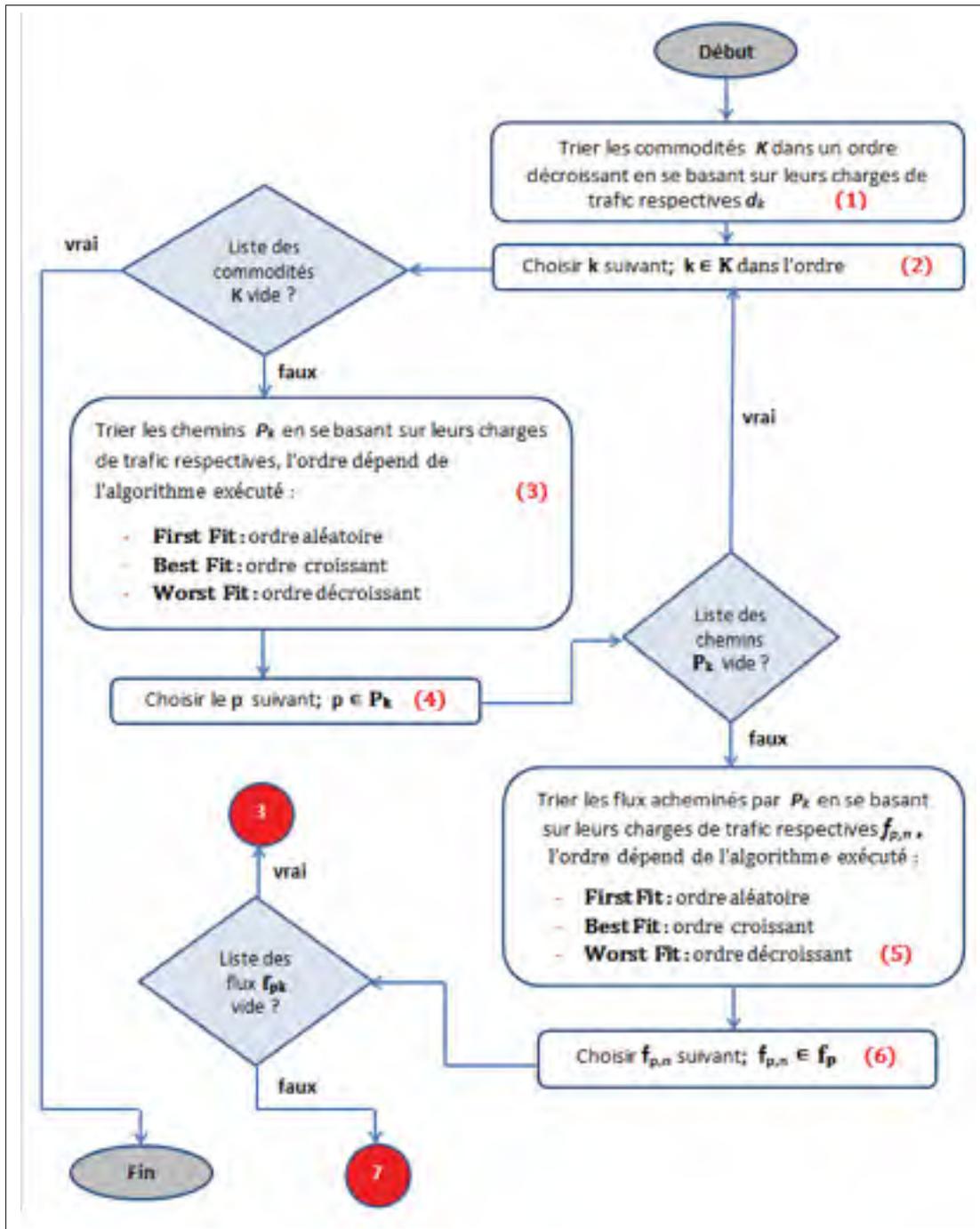


Figure 2.3 Organigramme représentant le trie qui est fait au début des trois algorithmes de l'approche heuristique

Étape 1 : une fois la demande de trafic d_k de chaque commodité k connu, un tri décroissant des commodités est effectué en se basant sur leurs demandes de trafic respectives.

Étape 2 : suivant l'ordre établi par le tri, les commodités sont choisies une par une. Pour chaque commodité choisie, on lui applique l'un de nos algorithmes d'optimisation (First Fit, Best Fit, Worst Fit).

Étape 3 : un tri est appliqué aux chemins correspondants à la commodité choisie. Le tri est basé sur la charge de trafic des chemins. La charge de trafic d'un chemin correspond à la charge du lien le plus élevé. La charge d'une liaison correspond à la somme des charges de flux qui transitent sur la liaison. Le type de tri varie dépendamment de l'algorithme choisi comme illustré dans l'organigramme.

Étape 4 : une fois le tri terminé, les chemins sont choisis un à un dans l'ordre effectué.

Étape 5 : pour chaque chemin choisi, un tri est appliqué sur les flux qu'il transporte. Le type de tri varie dépendamment de l'algorithme choisi comme illustré dans l'organigramme.

Étape 6 : une fois les flux ordonnés, les flux sont choisis un à un pour être rerouté sur les autres chemins restants. Après avoir parcouru toute la liste des flux d'un chemin, un tri est effectué sur les chemins restants (similaire à celui de l'étape (3)). Ce tri ne touchera que les chemins dont les flux n'ont pas été parcouru encore parcourus.

2.5.2.1 Algorithme First Fit (FF)

Cet algorithme se caractérise par sa rapidité d'exécution étant donné qu'une itération de ce dernier s'arrête dès qu'une solution possible est trouvée dans l'espace des solutions possibles. Par rapport à notre problème, chaque itération consiste à chercher un chemin dont la bande passante non allouée est assez large pour transporter le flux à rerouter et qui satisfait les contraintes prédéfinies. L'organigramme de cet algorithme est illustré dans la figure 2.4.

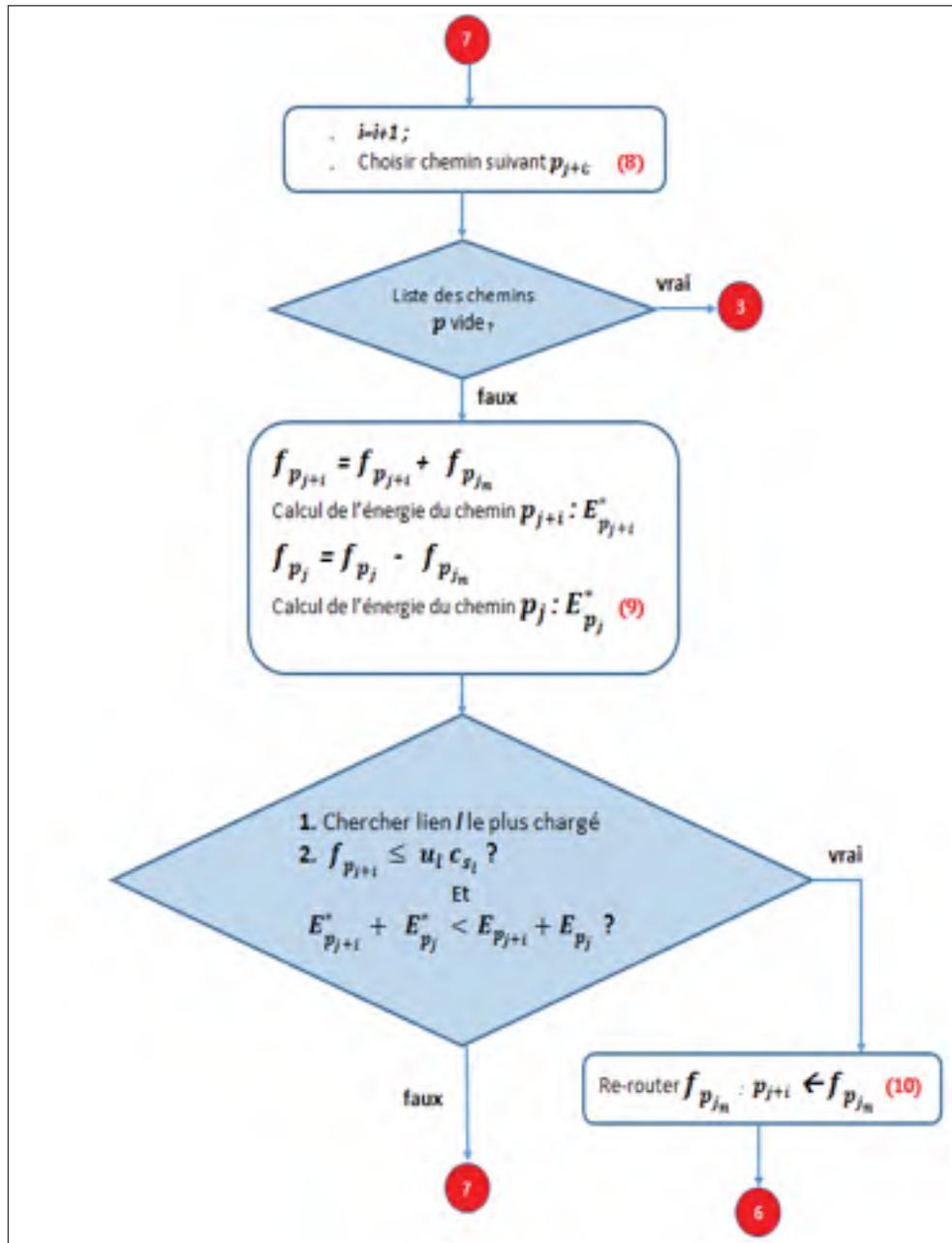


Figure 2.4 Organigramme représentant l'approche heuristique :
First Fit (FF)

Étape 8 : une fois tous les flux d'un chemin p_j parcouru, on passe au chemin suivant p_{j+i} pour essayer de rerouter ses flux sur les chemins restants.

Étape 9 : la charge de trafic de chaque flux $f_{p_{jn}}$, est ajouté à la charge de trafic de chaque lien l composant le chemin p_{j+i} . Dans le cas où la nouvelle charge des liaisons est inférieure ou égale à la capacité maximale définit $u_l c_{s_l}$, on passe à l'étape (10) afin de comparer la consommation énergétique des deux chemins $\{p_j, p_{j+i}\}$. Dans le cas contraire, on passe au flux suivant.

Étape 10 : l'énergie du chemin p_{j+i} est calculée à la base de cette nouvelle charge de trafic. De la même manière on soustrait $f_{p_{jn}}$ de la charge des liens composant le chemin p_j , ensuite l'énergie de ce dernier est actualisée. Si la charge du flux $f_{p_{jn}}$ fournit une moins importante consommation énergétique, le flux est rerouté définitivement sur le chemin p_{j+i} . Dans le cas contraire le flux est maintenu sur le chemin p_j .

2.5.2.2 Algorithme Best Fit (BF)

Cet algorithme consiste à parcourir pour chaque flux sélectionné pour être rerouté, tous les chemins possibles afin de trouver le chemin dont la bande passante non allouée est la plus petite possible pour transporter le flux à rerouter et qui satisfait les contraintes prédéfinies. L'organigramme de cet algorithme est illustré dans la figure 2.5.

Étape 8 : on définit une liste vide (*Best*). Chaque champ de cette liste contient deux valeurs. La première valeur représente l'identifiant d'un des chemins sélectionné pour recevoir le flux qui peut être rerouté. La deuxième valeur représente la bande passante non utilisée $diff_i$ du chemin p_{j+i} dans le cas où le flux choisit $f_{p_{jn}}$ est rerouté.

Étape 9 : similaire à l'étape (8) du diagramme représentant l'approche First Fit.

Étape 10 : similaire à l'étape (9) du diagramme représentant l'approche First Fit.

Étape 11 : une fois tous les chemins capables d'accueillir le flux $f_{p_{jn}}$ parcourus et la liste *Best* rempli, on procède à un tri croissant de cette liste. La liste est triée par rapport à la valeur $diff_i$ de chaque champ. Une fois triée le flux est rerouté sur le premier chemin de cette liste

Étape 12 : dans cette étape on calcule la valeur de $diff_i$ et on l'ajoute dans la liste $Best$. La variable $diff_i$ représente la différence entre la charge du lien l le plus chargé d'entre les liens du chemin p_{j+i} et la capacité maximale des liaisons $u_l c_{s_l}$.

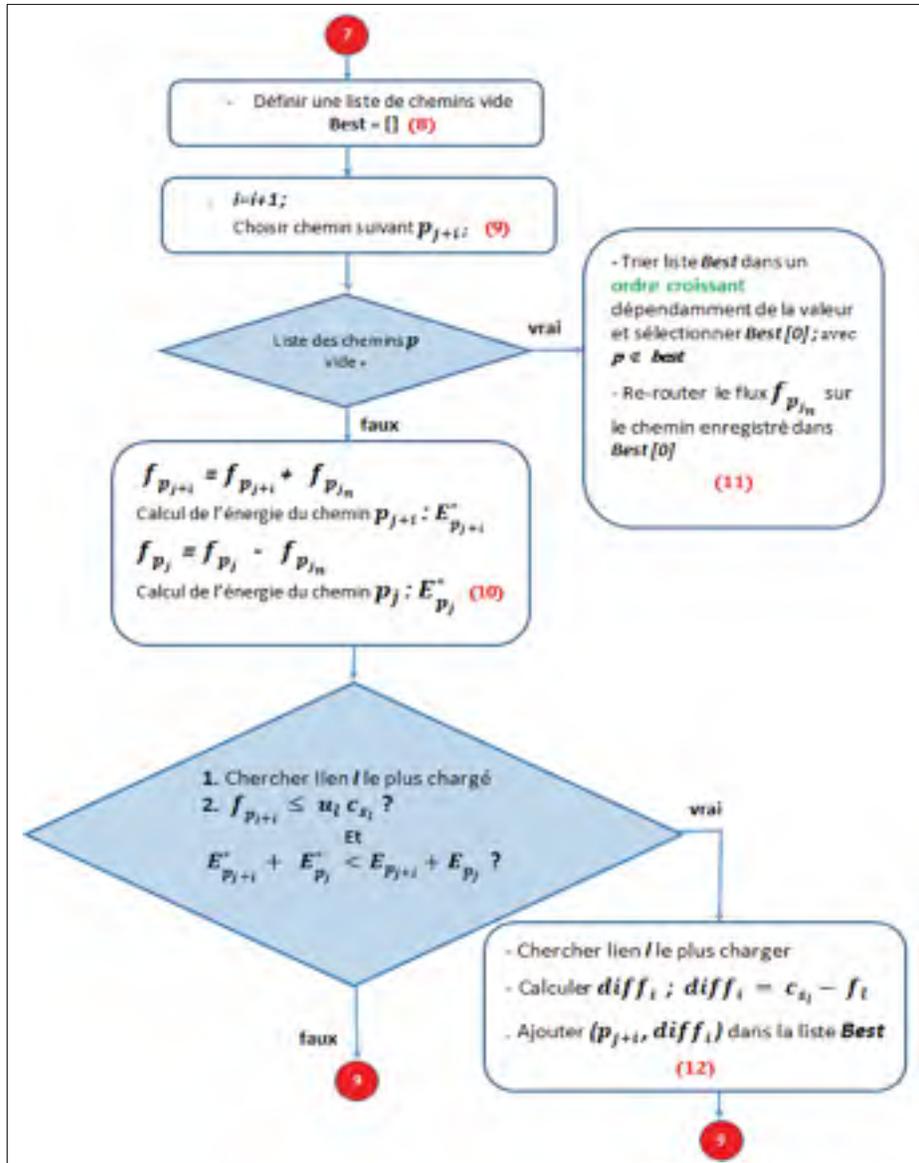


Figure 2.5 Organigramme représentant l'approche heuristique : Best Fit (BF)

2.5.2.3 Algorithme Worst Fit (WF)

Cet algorithme consiste à parcourir pour chaque flux sélectionné pour être rerouté, tous les chemins possibles afin de trouver le chemin dont la bande passante non allouée est la plus large possible pour transporter le flux à rerouter et qui satisfait les contraintes prédéfinies. L'organigramme de cet algorithme est illustré dans la figure 2.6.

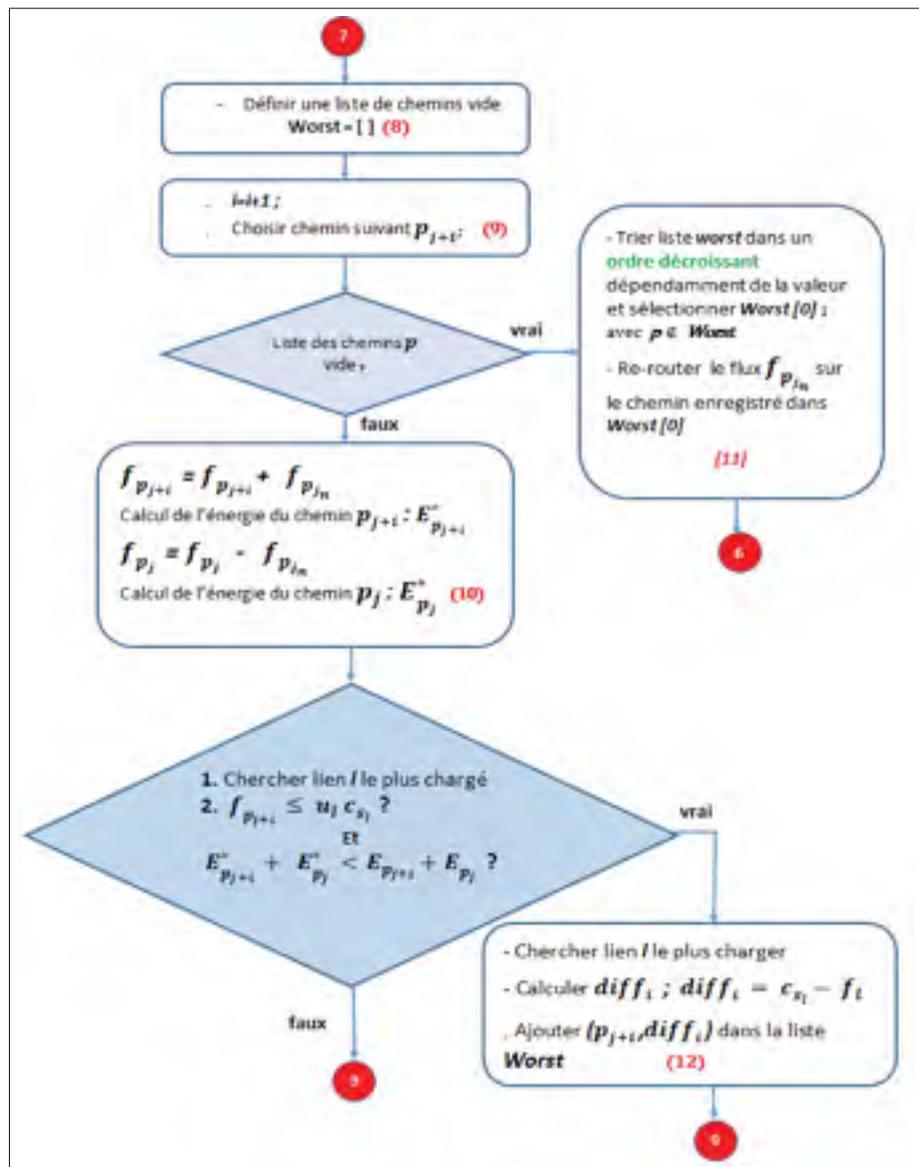


Figure 2.6 Organigramme représentant l'approche heuristique : Worst Fit (WF)

Étape 8 : on définit une liste vide (*Worst*). Chaque champ de cette liste contient deux valeurs. La première valeur représente l'identifiant d'un des chemins sélectionné pour recevoir le flux qui peut être rerouté. La deuxième valeur représente la bande passante non utilisée $diff_i$ du chemin p_{j+i} dans le cas où le flux choisit $f_{p_{j_n}}$ est rerouté.

Étape 9 : similaire à l'étape (8) du diagramme représentant l'approche First Fit.

Étape 10 : similaire à l'étape (9) du diagramme représentant l'approche First Fit.

Étape 11 : une fois tous les chemins capables d'accueillir le flux $f_{p_{j_n}}$ parcourus et la liste *Worst* rempli, on procède à un trie décroissant de cette liste. La liste est triée par rapport à la valeur $diff_i$ de chaque champ. Une fois triée le flux est rerouté sur le premier chemin de cette liste.

Étape 12 : dans cette étape on calcule la valeur de $diff_i$ et on l'ajoute dans la liste *Worst*. La variable $diff_i$ représente la différence entre la charge du lien l le plus chargé d'entre les liens du chemin p_{j+i} et la capacité maximale des liaisons $u_l c_{s_l}$.

2.5.3 Approche métaheuristique

Les algorithmes métaheuristiques sont des algorithmes d'optimisation visant à résoudre des problèmes d'optimisation difficile. Ils ont été étudiés et utilisés longuement dans la littérature pour la gestion des ressources. Ces algorithmes sont capables de trouver des solutions sous-optimales basées sur leurs algorithmes probabilistes. Les algorithmes génétiques (AG) (Goldberg *et al.* (1989)), algorithmes de colonies de fourmis (ACO) (Gambardella et Dorigo (2000)) et l'algorithme hybride (Jalali *et al.* (2013)) sont toutes des approches métaheuristiques. Par rapport à la programmation dynamique et aux heuristiques, les AG génèrent des données aléatoires et pour cette raison, ils ne peuvent pas garantir la recherche de solutions optimales. Cependant, en pratique les métaheuristiques se montrent plus puissantes que les méthodes de parcours exhaustif ou de recherche purement aléatoire.

2.5.3.1 Algorithme génétique (AG)

Durant ces vingt dernières années, plusieurs métaheuristiques ont prouvé leur efficacité pour la résolution de problèmes combinatoires, ensemble auquel appartiennent plusieurs problèmes industriels. Parmi ces métaheuristiques, on cite les algorithmes génétiques. Les algorithmes génétiques sont des méthodes évolutives qui s'inspirent fortement des mécanismes biologiques liés aux principes de sélection et d'évolution naturelle. Développés initialement par (Holland (1975)). Une présentation des concepts théoriques de base des algorithmes génétiques est détaillée dans la section Annexe I. Ces concepts sont nécessaires à la compréhension des AG et l'identification des principaux opérateurs de ces algorithmes.

Adaptation de l'AG à notre problème :

Cette étude nous a permis de voir comment adapter les opérateurs des algorithmes génétiques à notre cas d'étude. L'adaptation est décrite dans les sections suivantes :

La représentation

Dans notre cas, l'ensemble des symboles possible qu'un gène peut prendre sont des entiers compris dans l'ensemble $\{0, \dots, c_{S_i}\}$. Les éléments de cet ensemble représentent les charges de trafic possibles pour un flux donné (eg : figure 2.7). Après le codage d'individus, il faut définir une fonction d'évaluation pour évaluer ces individus par rapport à l'objectif du problème.

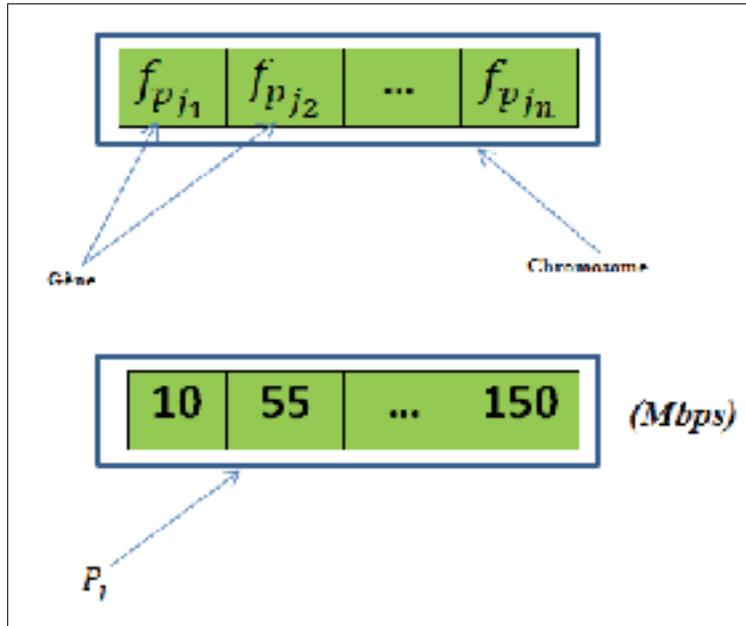


Figure 2.7 Représentation des gènes et des chromosomes

Fitness

La fonction d'évaluation de notre algorithme (3.21) représente l'énergie totale consommée par un chemin sous une distribution de flux qui satisfait la contrainte de la capacité (3.22).

$$Eva(p) = \sum_{\forall l \in p} E(f_l); \quad \forall p \in P_k \quad (2.21)$$

$$avec : f_l \leq u_l c_{s_l}; \quad \forall l \in p \quad (2.22)$$

Population initiale

Étant donné que notre algorithme s'exécute de façon itérative entre les commodités, chaque commodité aura sa propre population initiale. La figure 2.8 illustre la représentation dans notre travail de la population initiale pour chaque commodité. La taille de la population initiale diffère d'une commodité à une autre et est intimement liée au nombre de chemins préétablis pour chaque commodité. Comme cela est défini dans la formulation suivante :

$$Taille\ de\ pop_init = (Nb\ chromosome)^2$$

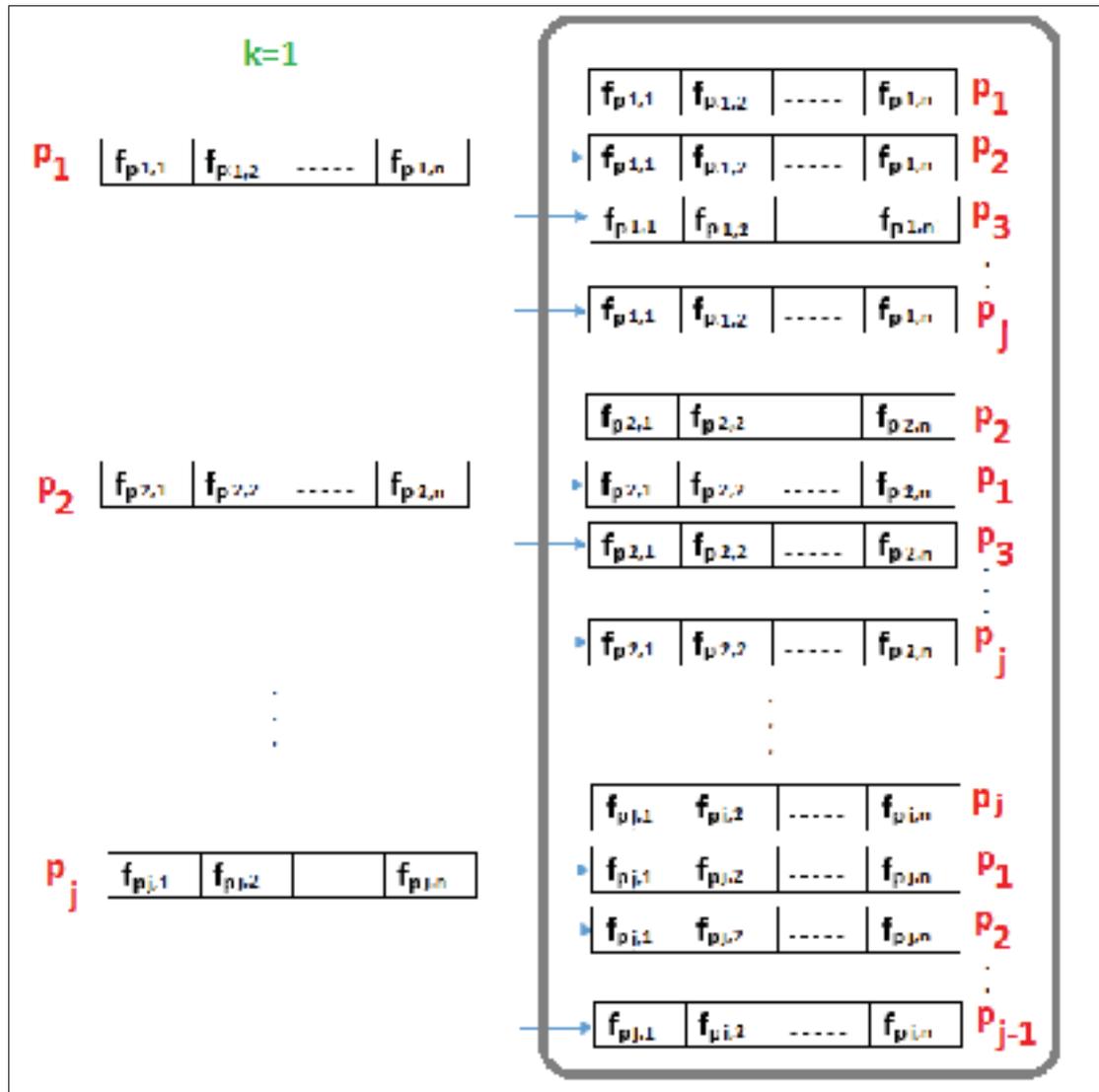


Figure 2.8 Cadre gris : population initiale de la commodit  $k=1$

L'op rateur de s lection

Il existe plusieurs m thodes de s lection d taill e dans l'Annex 1. En ce qui concerne le choix de la technique de s lection pour notre algorithme, on a opt  pour la technique de s lection par rang. La technique de s lection par rang va toujours favoriser les individus avec la meilleur fonction fitness   un moment donn  t , cette caract ristique ne va pas dans le sens de la diversification des individus. N anmoins,  tant donn  que notre but est d'optimiser l' nergie du r seau dans un minimum de temps possible afin de ne pas affecter les param tres de qualit  de service (QoS). Cette technique va permettre   notre algorithme de converger rapidement.

L'opérateur de croisement

Dans notre travail on a procédé à un croisement à un point de coupure. Comme l'illustre la figure 2.9, l'adaptation du croisement à notre cas d'étude s'est fait d'après les étapes suivantes :

- Les deux parents sont choisis de façon aléatoire.
- Un point de croisement aléatoire est choisi pour chaque parent. Ensuite on procède au croisement.
- On vérifie si les deux enfants respectent la contrainte de la capacité des liens.
- Si la contrainte de la capacité est respectée, on calcule l'énergie des deux chemins enfants résultants du croisement. Dans le cas contraire les enfants sont éliminés.
- Si la somme du coût en énergie des deux chemins enfants est inférieure à la somme du coût en énergie des deux chemins parents, les deux chemins enfants sont ajoutés à la population et les parents sont éliminés. Dans le cas contraire, c'est les enfants qui sont éliminés de la population et les parents y sont rajoutés.

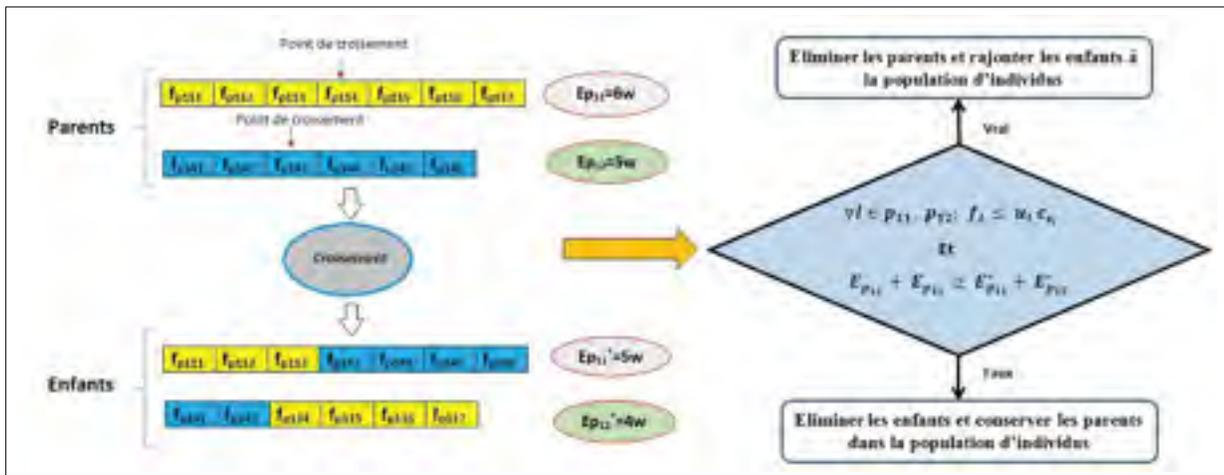


Figure 2.9 Croisement et validation des individus

L'opérateur de mutation

Dans notre travail on a procédé à une mutation aléatoire. Comme l'illustre la figure 2.10, l'adaptation de la mutation à notre cas d'étude s'est fait d'après les étapes suivantes :

- Deux parents sont choisis de façon aléatoire.

- b. Un point de mutation aléatoire est choisi pour chaque parent. Ensuite on procède à la mutation en interchange les flux choisis entre les deux parents.
- c. On vérifie si les deux enfants résultants respectent la contrainte de la capacité des liens.
- d. Si la contrainte de la capacité est respectée on calcule l'énergie des deux chemins enfants. Dans le cas contraire les enfants sont éliminés.
- e. Si la somme du coût en énergie des deux chemins enfants est inférieure à la somme du coût en énergie des deux chemins parents, les deux chemins enfant sont ajoutés à la population et les parents sont éliminés. Dans le cas contraire, c'est les enfants qui sont éliminés de la population et les enfants y sont rajoutés.

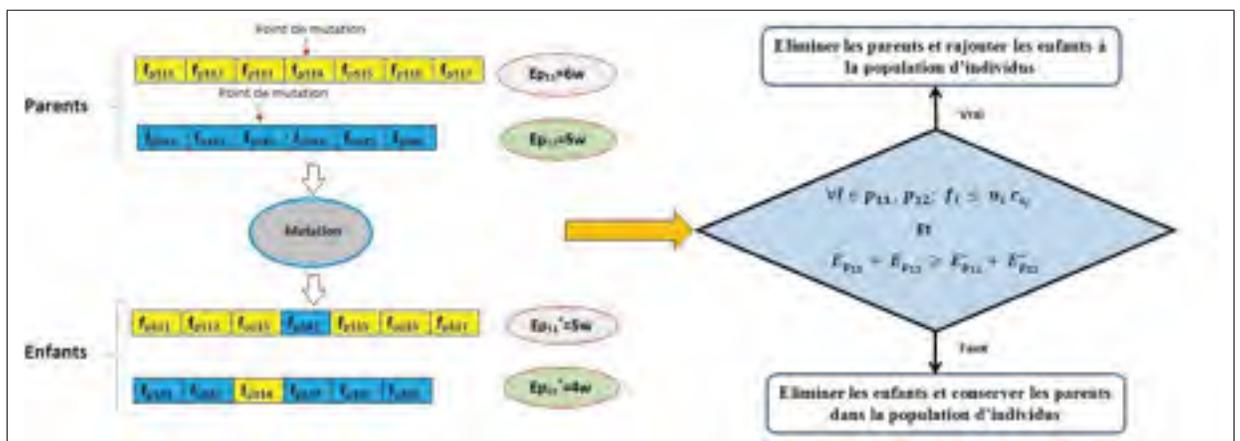


Figure 2.10 Mutation et validation des individus

L'opérateur de remplacement

La stratégie $(\nu + \lambda)$ a été retenue dans notre approche. Dans cette stratégie ce sont les ν meilleurs individus des $\nu + \lambda$ parents plus enfants individus qui sont retenus.

Critères d'arrêt

Nous avons opté pour un nombre d'itérations égales à cinq que nous jugeons suffisant pour obtenir un résultat satisfaisant.

2.6 Conclusion

Réaliser la proportionnalité entre la charge de trafic et l'énergie consommée dans le réseau assure la conservation de l'énergie et permet aux opérateurs de baisser les frais relatifs à leur consommation électrique.

Dans ce chapitre nous avons proposé un mécanisme évolutif économe en énergie en exploitant la capacité de la technique SDN à fournir une vue globale en temps réel du réseau. Pour ce faire, nous avons choisi de voir notre problème comme étant un problème d'optimisation de placement des flux avec comme objectif la minimisation d'énergie au niveau des liaisons. Étant donné que la solution à ce problème agit d'une manière directe sur les flux. Ceci nous a poussés à opter pour le routage écoconscient comme stratégie d'économie d'énergie. Une fois le reroutage des flux effectuée, les débits seront reconfigurés soit par le NIC si elle supporte l'auto-configuration des débits ou par le contrôleur SDN via un protocole de configuration des équipements. D'autre part, le principal inconvénient de la technique de reconfiguration des vitesses de transmission est la fréquence élevée de redirections de flux et de transition d'états de liaison. Ceci engendre du délai et des pertes de paquets élevées, ce qui affecte dangereusement la QoS au sein du réseau.

Suite à ces nouvelles contraintes notre problème d'optimisation a été élargi par la minimisation du nombre de transitions des états d'énergies des liaisons et la fréquence de reroutage des flux. On a modélisé et formulé le problème comme un problème ILP, qui est connu pour être NP-difficile ((Bolla *et al.*, 2016)). Ensuite, nous avons proposé plusieurs approches de résolution du problème au niveau du réseau : une approche formelle, une approche heuristique et une approche métaheuristique.

CHAPITRE 3

EXPÉRIMENTATION ET RÉSULTATS

3.1 Introduction

Ce chapitre présente les différentes expérimentations réalisées afin de tester, valider et comparer les algorithmes d'optimisation proposés. La première partie de ce chapitre décrit l'environnement expérimental et les scénarios exécutés. Ensuite, elle expose et analyse les différents résultats obtenus. Des synthèses sur la performance des algorithmes proposés seront retenues à la fin. La deuxième partie introduit un cas d'étude qui représente une instance d'implémentation et d'intégration de l'algorithme d'optimisation proposé dans l'environnement SDN.

3.2 Protocole expérimental

Le protocole expérimental introduit les différents scénarios exécutés ainsi que leurs environnements de tests.

- **scénarios**

Dans ce travail trois scénarios expérimentaux sont considérés. Le premier scénario vise à comparer les performances des algorithmes proposés. Il est limité à une paire de nœuds. Après avoir effectué une analyse de sensibilité des différents algorithmes basée sur le taux d'économie d'énergie et le temps d'exécution pour un cas simple, l'algorithme le plus performant est choisi pour être testé sur des topologies plus complexes dans les scénarios suivants. Le deuxième scénario évalue l'efficacité temporelle de l'approche formelle résolue avec l'outil Cplex. Le dernier chapitre considère en plus du taux d'économie d'énergie, les contraintes relatives au nombre du flux rerouté et au nombre de changements d'état des liaisons. Ainsi, les algorithmes sont évalués en considérant ces nouvelles contraintes. Une étude de sensibilité est faite pour chacun des scénarios.

- **modèle de consommation d'énergie**

Un modèle de consommation d'énergie pour les liaisons de données est utilisé dans les expérimentations. Ce modèle représente les débits de transfert des données et leurs consommations respectives supportées par la carte NIC.

Tableau 3.1 Modèle de consommation d'énergie
Tirée de Tang Jian (2012)

États	Capacités	Consommation d'énergie
1	10Mbps	0.81 w
2	100Mbps	0.96 w
3	1Gbps	1.4 w
4	10Gbps	10 W

- **topologies simulées et métriques considérées**

Pour évaluer la performance des algorithmes sur des topologies réelles de grandes échelles, deux topologies de réseau bien connues sont simulées : le réseau NSF constitué de 14 nœuds et 21 liens représentés sur la figure 3.1 et le réseau Abilene avec 11 nœuds et 14 Liens représentés sur la figure 3.2. Les métriques considérées pour la comparaison des performances des algorithmes sont le temps de traitement en seconde (s) et l'énergie économisée en pourcentage (%). Le temps de traitement représente le temps d'exécution d'un algorithme pour une topologie précise sous une distribution de trafic donnée. La métrique d'économie d'énergie représente le pourcentage d'économie d'énergie à l'instant (t) entre deux exécutions d'un même algorithme, soit une exécution à (t - 1) et à (t), elle est calculée au moyen de la formule suivante :

$$\% \text{ Energie conomisee } (t) = 1 - \frac{\text{Energie consommee } (t) \text{ w} * 100}{\text{Energie consommee } (t - 1) \text{ w}} \quad (3.1)$$

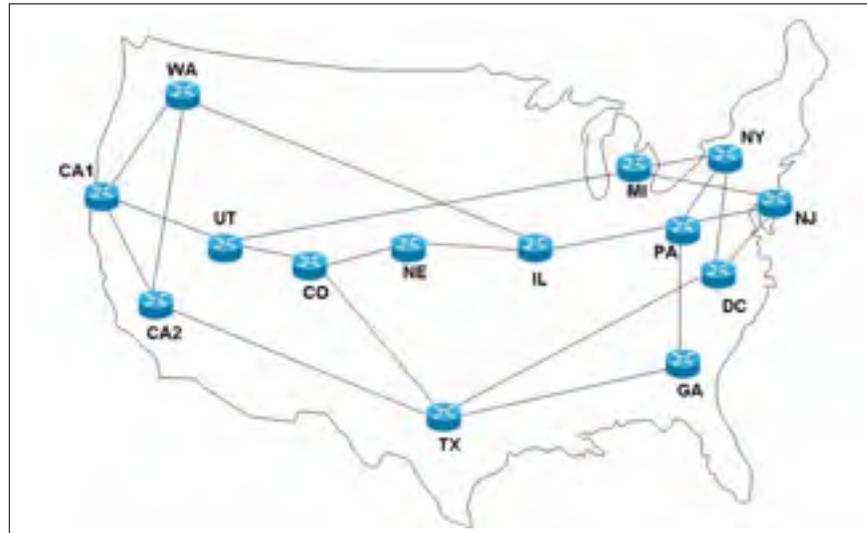


Figure 3.1 Topologie NSF
Tirée de nsf.gov (2008)



Figure 3.2 Topologie Abilene
Tirée de Cs.stanford.edu (2003)

- **implémentation**

Afin d'implémenter l'approche formelle, le solveur CPLEX d'IBM est utilisé. Ce dernier est implémenté dans le logiciel MATLAB (MATLAB (2010)). Les autres solutions d'optimisation ont été développées avec le langage Java. La machine utilisée pour l'exécution des algorithmes

est munie d'un processeur (CPU) Intel de génération i7-4770 avec une puissance de $8 \times 3.40\text{GHz}$ et une mémoire vive (RAM) de 16GB.

3.3 Scénarios des expérimentations et étude de sensibilité

3.3.1 Scénario 1 : mesure des performances des algorithmes pour ($K=1$)

- **objectif**

Dans ce premier scénario, la performance des différents algorithmes proposés pour le cas d'une session ou commodité unique est évaluée. Cela permet de connaître le comportement des algorithmes dans un cas simple où une seule commodité est considérée.

- **réalisation et résultats**

Pour la réalisation du premier test, 10 chemins de longueur aléatoire sont générés entre une paire de nœuds. Ensuite, un nombre différent de flux {50, 100, 150, 200} répartis de façon égale sur les chemins est simulé. Dans le cas de l'algorithme génétique, le test est répété pour différents paramètres présentés dans le tableau 3.2. En ce qui concerne la répartition de la charge de trafic des flux, les premiers 44 % des flux ont leur valeur de charge de trafic générée aléatoirement entre [0,9] Mbps, les 44 % suivants entre [10,99] Mbps, les 10 % des flux suivants avec une valeur de charge de trafic générée entre [100,999] Mbps et les 2 % restants ont leur valeur de charge de trafic générée entre [1000,9000] Mbps. Les états des liaisons, les seuils de capacité correspondants et la valeur de la consommation d'énergie sont donnés par le tableau 3.1 (Tang Jian (2012)). La simulation a été répétée 10 000 fois et la moyenne et le pourcentage d'économie d'énergie maximum ont été recueillis. Les résultats de ce scénario sont présentés dans les tableaux 3.3, 3.4, 3.5.

Tableau 3.2 Paramètres établis pour l'évaluation de l'AG

Paramètres	Valeurs
Population initiale (taille)	(Nb chromosome)^2
Operateur de sélection	Sélection par rang
Nombre de croisements	{1, 3, 5, 10} * Nb de flux par commodité
Probabilité de croisement	1
Probabilité de mutation	{ 0.02, 0.05 }
Critère d'arrêt	5 itérations

Tableau 3.3 Taux d'économie d'énergie moyen et maximal des algorithmes Firsts Fit, Best Fit et Worst Fit

Nb flux	First Fit		Best Fit		Worst Fit	
	Moy (%)	Max (%)	Moy (%)	Max (%)	Moy (%)	Max (%)
50	13.46	74.14	18.64	74.14	21.96	75.08
100	40.26	76.25	53.48	68.41	53.22	72.11
150	72.01	58.62	16.23	68.87	79.97	76.12
200	1.81	37.4	0.99	36.6	1.77	42.5

- **étude de la sensibilité**

L'étude de sensibilité de ce scénario est faite sur la base des paramètres suivants : taux d'économie d'énergie et temps de traitement.

Taux d'économie d'énergie :

Les résultats des algorithmes First Fit, Best Fit, Worst Fit en matière d'économie d'énergie

Tableau 3.4 Taux d'économie d'énergie moyen et maximal de l'AG testé pour une probabilité de mutation = 0,02 et différents nombres de croisements

Algorithme Génétique (AG)								
Probabilité de mutation = 0.02								
Nb flux	Croisements 10* nb flux				10		Croisements 10* nb flux	
	Moy (%)	Max (%)	Moy (%)	Max (%)	Moy (%)	Max (%)	Moy (%)	Max (%)
50	16.4	74.9	17.15	76	17.91	76.38	18.58	77
100	59.58	78.25	61.63	79.34	62.4	77.52	63.61	79.54
150	53.87	73.09	62.04	73.89	62.26	77.68	65.16	73.46
200	54.1	89.57.57	57.53	73.13	56.58	72.28	59.09	71.61

Tableau 3.5 Taux d'économie d'énergie moyen et maximal de l'AG testé avec une probabilité de mutation = 0,05 et différents nombres de croisements

Algorithme Génétique (AG)								
Probabilité de mutation = 0,05								
Nb flux	Croisements: 50 flux		Croisements: 100 flux		Croisements: 150 flux		Croisements: 200 flux	
	Moy (%)	Max (%)	Moy (%)	Max (%)	Moy (%)	Max (%)	Moy (%)	Max (%)
50	44.31	73.71	47.46	76.25	48.09	76.25	48.58	76.37
100	59.66	78.42	62.84	77.31	62.05	76.73	62.68	76.47
150	59.62	74.39	62.98	74.39	62.67	74.39	62.12	75.11
200	54.15	69.49	57.56	71.7	58.61	72.48	58.08	72.12

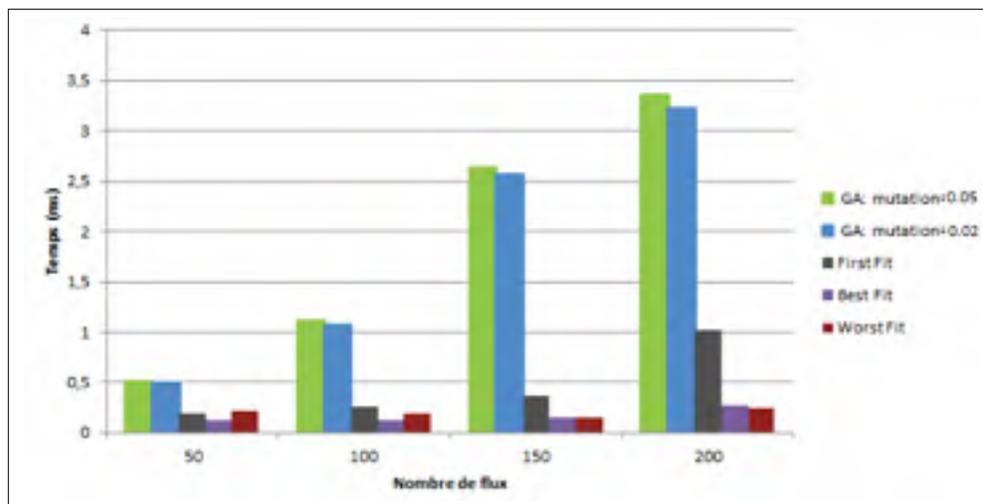


Figure 3.3 Temps de traitement moyens des algorithmes Firsts Fit, Best Fit, Worst Fit et AG testés avec différents nombre de flux

sont présentés dans le tableau 3.3. Les résultats montrent que le taux d'économie d'énergie est croissant entre 50 et 100 flux, le taux maximum est réalisé par l'algorithme First Fit. Cependant, le taux moyen le plus élevé est réalisé par l'algorithme Best Fit. D'autre part entre 150 et 200 flux, les résultats commencent à décroître. Le taux d'économie d'énergie maximum est réalisé par l'algorithme Worst Fit et le taux moyen le plus élevé est réalisé aussi par l'algorithme Worst Fit. Le taux le plus faible est égal à 1 % et réalisé par l'algorithme Best fit pour un nombre de flux égal à 200.

En ce qui concerne l'approche métaheuristique représentée par les AG, les résultats des tableaux 4.4 et 4.5 montrent que le taux d'économie d'énergie est régi par le nombre de croisements effectués. En effet, ce taux augmente d'environ 4 % entre un nombre de croisements égal à une fois le nombre de flux et un nombre de croisements égal à dix fois le nombre de flux, et ce quel que soit le nombre de flux considéré. De ce fait, dans les prochains scénarios le nombre de croisements est fixé à dix fois le nombre de flux par commodité. Par ailleurs, la probabilité de mutation n'améliore que légèrement le pourcentage d'économie d'énergie. Cette amélioration ne dépasse pas les 1 % en général.

Temps de traitement :

Le temps de traitement des algorithmes Best Fit, Worst fit est le plus faible et ne varie que légèrement avec l'augmentation du nombre de flux par commodité. D'autre part, le temps de traitement de l'algorithme First Fit augmente de façon linéaire avec l'augmentation du nombre de flux pour atteindre un maximum de 1 ms.

En ce qui concerne les AG, leur temps de traitement varie aussi linéairement que celui de l'algorithme First Fit. Cependant, en valeur numérique, il représente approximativement 3 fois le temps de calcul de ce dernier. Par ailleurs, la probabilité de mutation influe légèrement sur le temps de traitement par une valeur approximative de 0,1 ms.

Pour conclure, cette étude de sensibilité révèle les limites de l'approche heuristique représentée par les algorithmes Firsts Fit, Best fit et Worst fit en matière d'économie d'énergie. En effet, le taux d'économie d'énergie de ces algorithmes décroît très rapidement avec l'augmentation du nombre de flux actifs. Cette limitation rend ces algorithmes inadéquats pour les réseaux dorsaux (qui concentrent et transportent les flux de données entre des réseaux affluents) caractérisés par un débit élevé. D'autre part, contrairement aux algorithmes de l'approche heuristique le taux d'économie d'énergie de l'algorithme génétique décroît légèrement avec l'augmentation du nombre de flux de 150 à 200 flux tout en gardant une courbe de temps de traitement linéaire croissante. Étant donné performance de l'AG, ceci conduit à considérer seulement l'AG dans les prochains scénarios où de multiples commodités sont prises en compte.

3.3.2 Scénario 2 : mesure du temps de traitement de l’outil Cplex pour l’approche formelle

- **objectif**

Le deuxième scénario a pour objectif d’exprimer et de définir la relation qui existe entre le nombre de flux, la variation de la charge de trafic avec le temps de traitement de l’outil Cplex utilisé pour la résolution du modèle d’économie d’énergie formulé dans le chapitre 3.

- **réalisation et résultats**

La réalisation de ce scénario consiste à générer les 4 et 5 plus courts chemins entre chaque paire origine-destination de la topologie Abilene en utilisant une implémentation de l’algorithme *k-shortest paths*. Ensuite, pour chaque chemin, {10, 30, 50} flux sont générés. Les flux sont générés suivant différentes variations. Ces variations sont régies par les paramètres de variance, moyenne, valeur maximale et valeur minimale de charge de trafic. Les valeurs des différents paramètres de ce scénario sont illustrées dans le tableau 3.6 et les résultats du scénario sont illustrés dans les figures 3.4, 3.5.

Tableau 3.6 Paramètres de variation de la charge de trafic pour les {10, 30, 50} flux générés

Paramètres	Cas 1	Cas 2	Cas 3
max	10 Gbps * 0,9	10 Gbps * 0,9	10 Gbps * 0,9
min	0	10	500
moyenne	10 Mbps	100 Mbps	2,5 Gbps
variance	10 Mbps	90 Mbps	2 Gbps

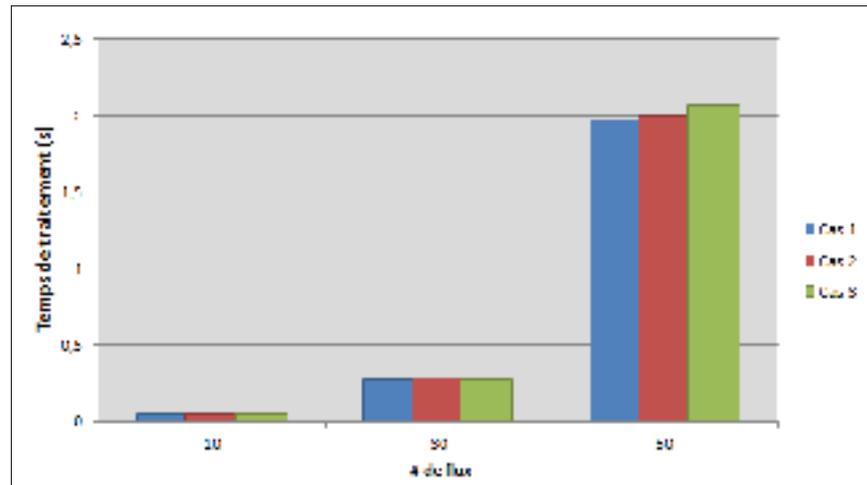


Figure 3.4 Impact du nombre de flux et de la variation du trafic sur le temps de traitement du Cplex (3 chemins précalculés)

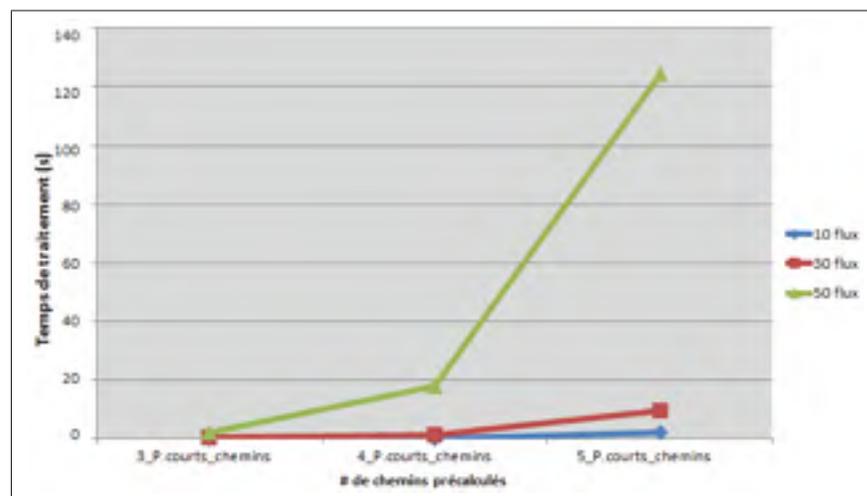


Figure 3.5 Impact du nombre de chemins précalculés sur le temps de traitement du Cplex (3 chemins précalculés)

- **étude de la sensibilité**

La figure 3.4 illustre l'impact du nombre de flux et de la variation du trafic sur le temps de traitement du Cplex. En effet, dans cette figure on remarque qu'en faisant varier la charge de trafic des flux le temps de traitement reste pratiquement inchangé. Cependant, en augmentant le nombre de flux entre les paires (O-D) il en résulte un temps de traitement avec une allure exponentielle. Dans la figure 3.5 on remarque que pour le même nombre de chemins préétablis

le temps de traitement varie de façon exponentielle avec l'augmentation du nombre de flux par chemins.

La conclusion de ce scénario est que l'outil Cplex utilisé pour résoudre le modèle d'économie d'énergie est impacté par l'augmentation du nombre de flux dans le réseau. D'autre part, il est insensible à la variation de la charge de trafic. Ainsi, la résolution par Cplex du modèle d'économie d'énergie formulé est appropriée pour les réseaux à forte variation de trafic. D'autre part, il est moins adapté aux réseaux à forte densité de flux.

3.3.3 Scénario 3 : mesure de la performances de l'approche métaheuristique pour ($K>1$)

- **objectif**

Le troisième scénario évalue la performance de l'algorithme génétique adapté au problème posé, sur des topologies et un trafic réels. Pour ce faire, deux topologies de réseaux dorsales (Abilene et NSF) illustrées dans les figures 3.1 et 3.2 ont été considérées. Pour l'évaluation de la performance de l'AG, dans ce scénario les contraintes de reroutage en terme de nombre de transition d'états des liaisons et nombre de flux rerouté sont prises en compte.

- **réalisation et résultats**

La réalisation de ce scénario se décompose en deux étapes. La première étape consiste à générer les 4 et les 5 plus courts chemins entre 25 paires origine-destination choisies aléatoirement, et ce pour chacune des topologies. Le nombre total de chemins générés dans le réseau est égal à 50 et 125 chemins respectivement. L'algorithme utilisé pour la génération des k-plus courts chemins sans boucles est le "*Yen's en's k shortest path algorithm*" (Yen (1971)).

Dans la deuxième étape de ce scénario, les matrices de trafic réel transmises le 2004-03-01 entre 00 :05 :00 et 01 :00 :00 par les nœuds du réseau Abilene (Uhlig *et al.* (2006)) sont considérées. Ces matrices contiennent des échantillons de charge de trafic à l'échelle de 5 minutes. Les tests ont été effectués à différentes échelles de temps, dont 5, 10, 15, 20, 25, 30, 45 et 60 minutes. Il faut noter aussi qu'en raison du faible débit transmis entre les paires (O-D)

au moment de la collecte des données (voir figure 3.6), dans notre travail on a multiplié chaque valeur par 100. Un échantillon du comportement de trafic échangé entre deux paires (O-D) est illustré dans la figure 3.7.

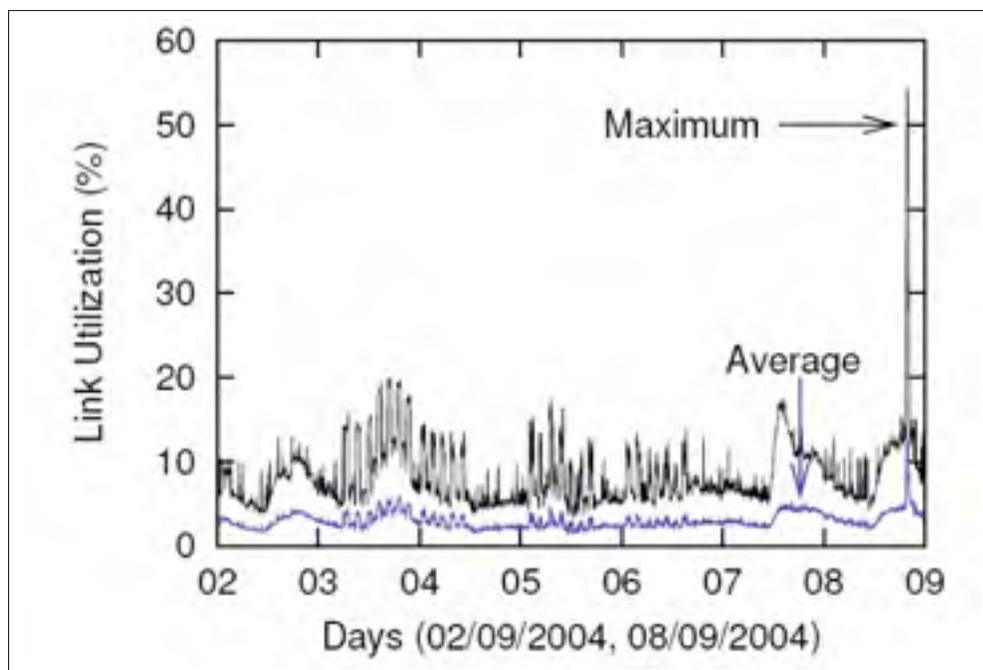


Figure 3.6 Utilisation maximale et moyenne des liens dans le réseau Abilene
Adaptée de Mingui *et al.* (2010)

Durant cette étape, 20 charges de flux ont été générées entre les 25 chemins de chaque paire (O-D) et ce pour les 2 topologies. Ensuite, 25 paires (O-D) de la matrice de trafic du réseau Abilene sont choisies aléatoirement. La charge de trafic de chacune de ces paires (O-D) est dévisée aléatoirement entre les chemins des paires (O-D) des topologies simulées. De cette façon, 1000 et 2500 flux sont générés en considérant l'existence de 2 et 5 chemins précalculés respectivement.

Pour chacune des topologies générées, le nombre de transitions d'états des liaisons, le nombre de reroutages des flux ainsi que le taux d'économie d'énergie sont calculés. Pour respecter ces contraintes le deuxième modèle formalisé dans le chapitre 3 est considéré et exécuté avec le

solveur Cplex. Pour évaluer la performance de l'AG sous ces nouvelles contraintes, le critère d'arrêt de l'AG est modifié et réadapté pour dépendre de la satisfaction des contraintes établies ou de l'arrivée à 1000 itérations sans satisfaction des contraintes. Dans ce dernier cas, en pratique aucun changement ne sera appliqué au niveau du réseau. Pour ce scénario, le nombre maximal de reroutage de flux est fixé à 10 %, soit 100 flux et 250 flux pour les deux cas considérés. Le nombre maximal de transitions des liaisons est fixé à 60 %, soit 25 et 34 transitions pour la topologie Abilene et NSF respectivement. Les résultats sont illustrés dans les figures 3.8, 3.9, 3.10, 3.11 et le tableau 3.7.

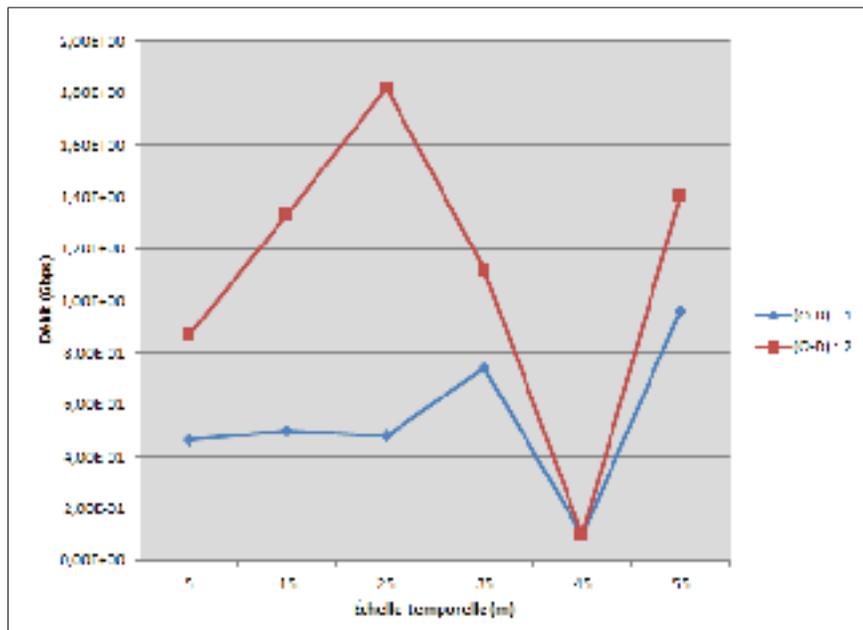


Figure 3.7 Échantillon de comportement du trafic échangé entre deux paires (O-D) du réseau Abilene

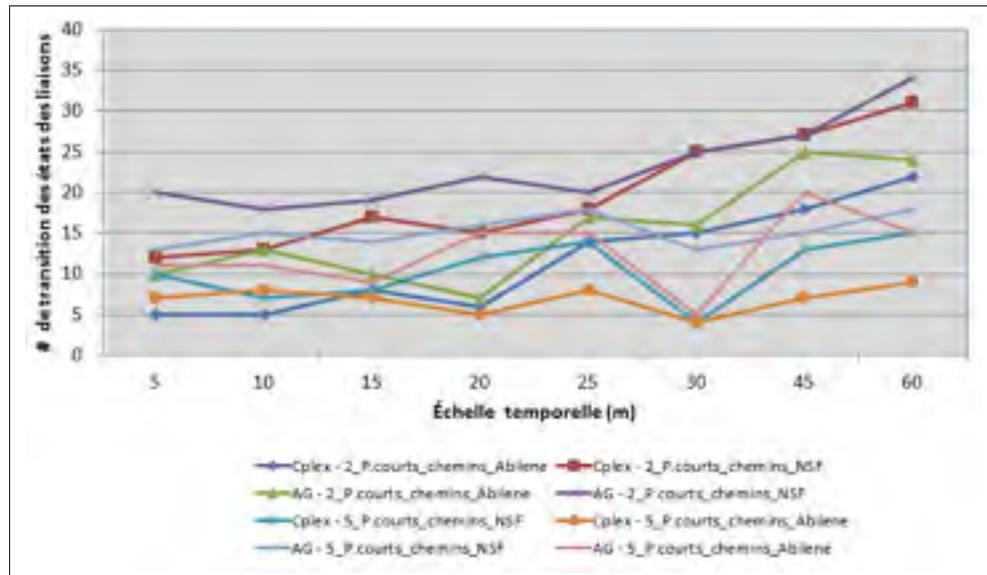


Figure 3.8 Nombre de transition des états des liaisons résultantes de l'AG et l'outil Cplex pour différentes matrices de trafic, différents nombres de chemins et différentes topologies (Abilene, NSF)

Tableau 3.7 Nombre d'itérations de l'AG pour différentes matrices de trafic, différents nombre de chemins et différentes topologies (Abilene, NSF)

# itérations de l'AG								
	5	10	15	20	25	30	45	60
AG - 2_P.courts_chemins_Abilene	40	38	39	42	40	45	47	44
AG - 2_P.courts_chemins_NSF	80	88	81	27	87	86	45	44
AG - 5_P.courts_chemins_Abilene	31	299	29	30	29	25	340	35
AG - 5_P.courts_chemins_NSF	28	80	29	26	82	28	30	330

- **étude de la sensibilité**

La figure 3.8 représente le nombre de transitions des états des liaisons induites par l'AG et l'outil Cplex pour différentes matrices de trafic du réseau Abilene, différents nombres de chemins préétablis pour les deux topologies Abilene et NSF. Si on compare entre le nombre de transitions des états des liaisons pour la topologie NSF et la topologie Abilene on trouve que le nombre de transitions de la topologie NSF est toujours supérieur à celui de la topologie Abilene et ce dans le cas de l'AG et le solveur Cplex. De plus en remarque que les résultats de l'AG

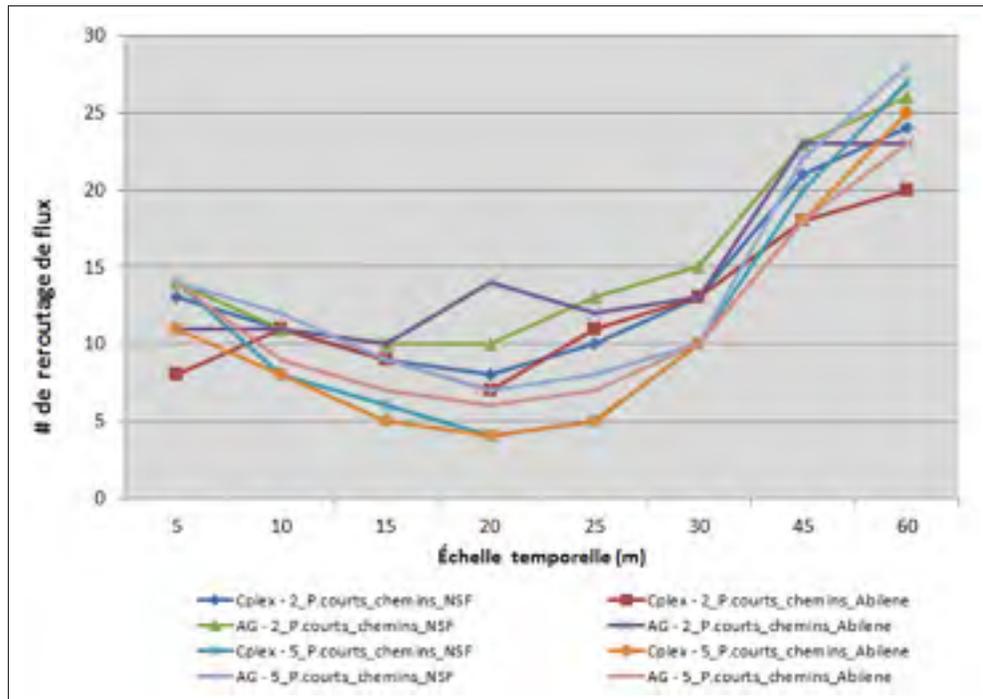


Figure 3.9 Nombre de reroutage de flux résultants de l'AG et l'outil Cplex pour différentes matrices de trafic, différents nombres de chemins et différentes topologies (Abilene, NSF)

sont très proches et des fois égale à ceux du Cplex comme à la 30e minute (25 transitions). En dernier lieu on constate que le nombre de transitions est proportionnel à la charge de trafic échangé entre les paires (O-D) comme illustré dans la figure 3.7.

Dans la figure 3.9, le nombre de reroutage des flux engendrés par l'AG et l'outil Cplex dépendamment du nombre de chemins précalculé est illustré. Comme dans la figure 3.8, on remarque que le nombre de reroutage dans la topologie NSF est toujours supérieur à celui de la topologie Abilene et ce dans le cas de l'AG et l'outil Cplex. De même, on remarque que les résultats de l'AG sont très proches dans la plupart des cas et des fois égales à ceux du Cplex comme à la minute 10 (11 flux reroutés) et la minute 30 (13 flux reroutés). En dernier lieu on constate que le nombre de reroutage augmente avec l'augmentation de la charge de trafic échangée entre les paires (O-D) comme illustrée dans la figure 3.7.

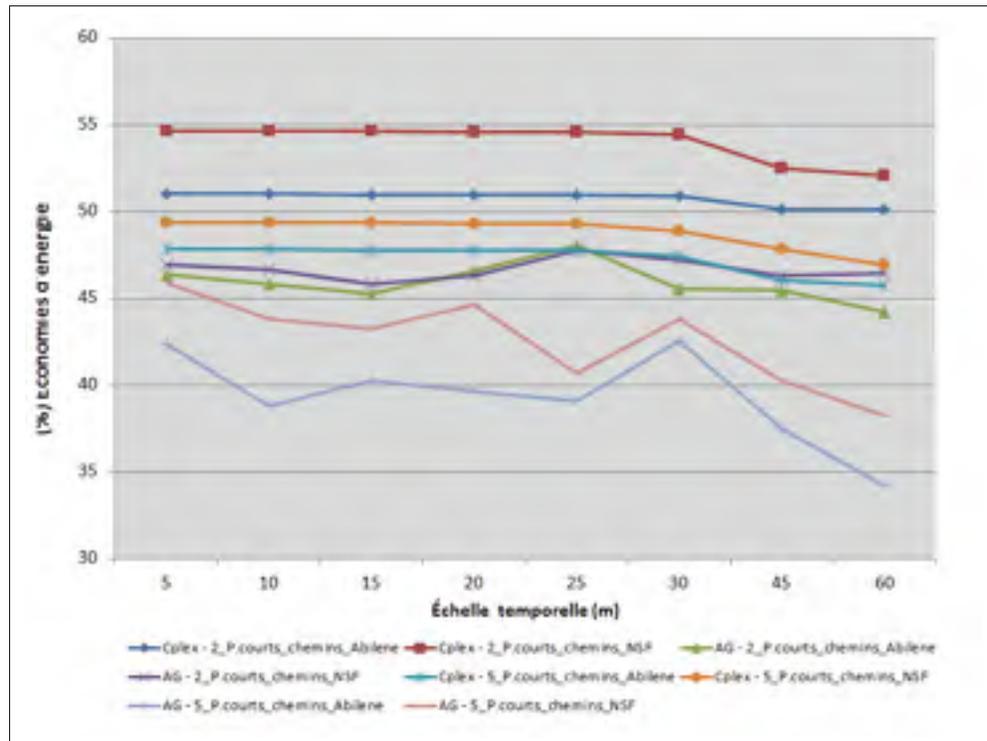


Figure 3.10 Taux d'économie d'énergie de l'AG et l'outil Cplex pour différentes matrices de trafic, différents nombres de chemins et différentes topologies (Abilene, NSF)

En ce qui concerne la courbe d'économie d'énergie illustrée dans la figure 3.10, on remarque que le taux d'économie engendré par le Cplex dépasse d'approximativement 5 % le taux d'économie engendré par l'AG pour tous les cas considérés ($\{2,5\}$ plus courts chemins NSF, Abilene). On remarque aussi que les meilleurs résultats de l'algorithme génétique sont présentés par la topologie NSF avec un maximum de 46 % dans le cas où 5 chemins sont préétablis et un maximum de 47 % dans le cas où 2 chemins sont préétablis. En dernier lieu on remarque que contrairement au nombre du reroutage des flux et au nombre de transitions des liaisons le taux d'économie d'énergie décroît avec l'augmentation de la charge de trafic.

Le temps de traitement de l'AG et le nombre d'itérations correspondant après introduction des nouvelles contraintes sont représentés par la figure 3.11 et le tableau 3.7 respectivement. On peut voir que pour chaque temps de traitement élevé correspond un pic dans le nombre d'itérations ce qui veut dire que le temps de traitement est proportionnel au nombre d'itérations

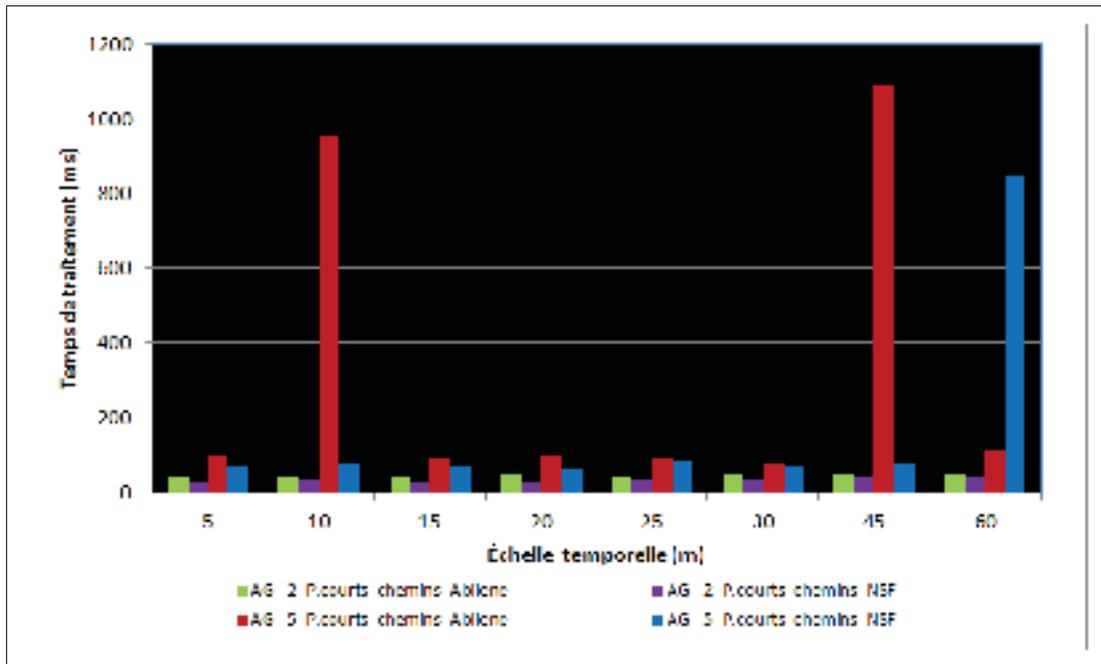


Figure 3.11 Temps de traitement de l'AG pour différentes matrices de trafic, différents nombre de chemins et différentes topologies (Abilene, NSF)

effectuées par l'AG. De plus, on remarque que le nombre d'itérations maximales est de 340 itérations, ce qui veut dire que l'AG arrive à satisfaire les contraintes de reroutage (10 %) et de transition des liaisons (60 %) sur un maximum de 340 itérations (1000 itérations). Cependant, contrairement aux autres paramètres considérés, le temps de traitement est plus important pour la topologie Abilene. Ceci est dû au fait que la topologie NSF est plus dense que la topologie Abilene en matière de liaisons et présente moins de liaisons jointes entre les chemins préétablis. De ce fait, l'AG prend plus de temps à trouver une solution faisable pour la topologie Abilene.

Les conclusions de ce scénario sont :

- l'AG présente un taux d'économie proche du taux optimal présenté par l'outil Cplex ;
- l'AG satisfait aux contraintes de reroutage et nombre de transitions fixées ;
- l'AG est plus efficace en temps de traitement pour les topologies denses présentant moins de liaisons jointes entre les chemins ;

- les nombres de reroutage des flux et le nombre de transitions des états des liaisons sont proportionnels à la charge de trafic.

3.4 Cas d'étude

3.4.1 Architecture d'implémentation

Ce projet a proposé une architecture pour l'adaptation dynamique de l'énergie des réseaux capables de réagir aux pics de trafic. Néanmoins, la réaction au changement de trafic est fonction du temps requis pour recueillir les statistiques des flux, exécuter la solution et enfin reconfigurer l'état des liaisons et pousser les nouvelles routes dans les tables des commutateurs. Cette architecture se compose de trois modules logiques illustrés dans la figure 3.12 : le module de collecte et traitement des données, le module d'optimisation et le module de contrôle.

3.4.1.1 Module de collecte des données

Le module de collecte des données est responsable de l'extraction des informations concernant les flux, comme le mappage ports/flux ainsi que la bande passante consommée par ce flux sur un port donné en temps réel. L'extraction de ces informations se fait à l'aide des interfaces de programmation (*API*) fournies par l'outil de supervision des flux sFlow (voir Annexe III). Dans notre travail l'échantillonnage des flux effectué par l'agent sflow est fixe à 1 afin d'avoir l'information sur chaque flux qui traverse un port donné du commutateur. Quant à l'échantillonnage des statistiques d'interface, il est fixé à 1 seconde.

Ce module est responsable aussi de l'extraction des informations concernant la topologie réelle à l'instant t au moyen des interfaces de programmation (*API*) fournies par le contrôleur OpenDaylight. Ce banc d'essai utilise l'*API topology manager* pour extraire la topologie de notre réseau (voir figure 3.13). Celle-ci est encodée dans un fichier Json ou Xml (voir Annexe III).

Une fois les informations à propos des flux extraites et la topologie connue, le module traite ces données afin de reconstituer une vue globale sur l'état du réseau. Le résultat du traitement

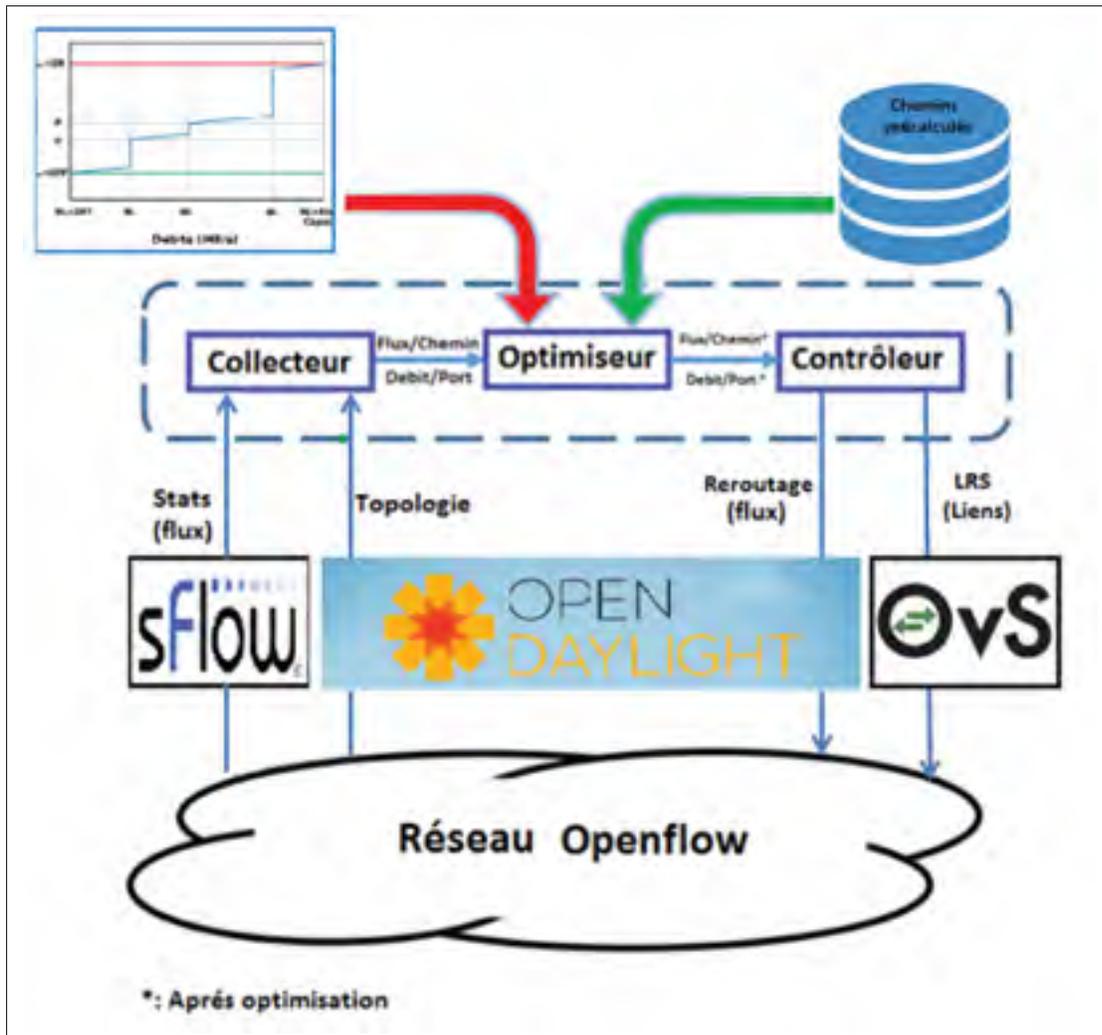


Figure 3.12 Architecture proposée

constitue une partie des données en entrées du module d'optimisation. Les traitements effectués sont :

- calcul de la bande passante utilisée au niveau de chaque port ;
- mappage entre les liens et leurs chemins respectifs ;
- mappage entre les flux et les chemins précalculés qui les transportent ;
- calcul de l'énergie consommée par chaque chemin, chaque paire origine-destination ainsi que l'énergie consommée par la globalité du réseau a l'instant $t=0$;
- assignation des paires origine-destination aux liens, chemins, flux qui leur sont associés.

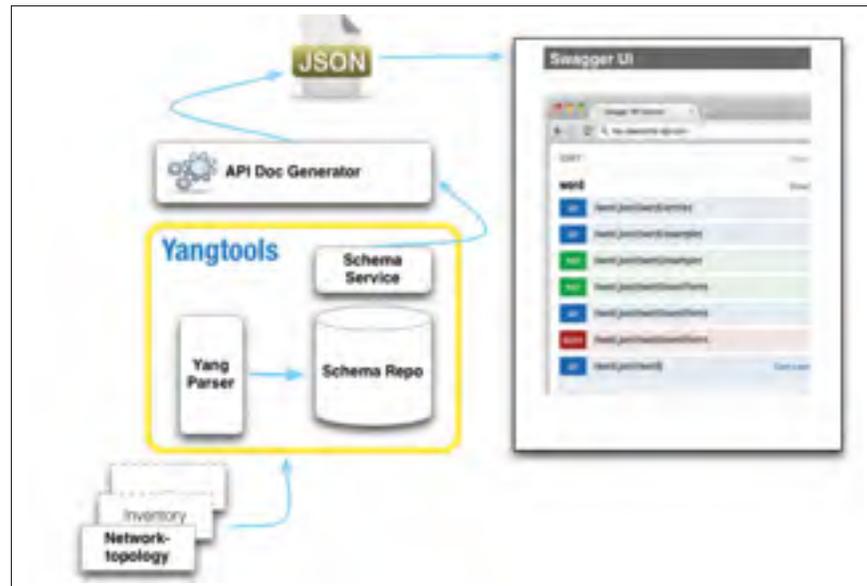


Figure 3.13 Génération de la topologie par le contrôleur Opendaylight
Tirée de opendaylight.org (2017)

3.4.1.2 Module d'optimisation

Le module d'optimisation est responsable de l'exécution de l'algorithme d'optimisation afin de trouver une nouvelle distribution des flux qui réalise une économie d'énergie et qui satisfait les contraintes des flux du moment. Ses entrées sont la topologie, la matrice de trafic, un modèle de consommation énergétique pour les liaisons et les propriétés de tolérance aux pannes désirées (capacité de réserve). Après l'exécution de l'algorithme d'optimisation, ce module délivre au module de contrôle la nouvelle distribution des flux ainsi que la nouvelle configuration des ports (débits) subséquents.

3.4.1.3 Module de contrôle

Le module de contrôle permet de pousser les nouvelles routes dans les commutateurs du réseau et changer ensuite l'état d'énergie des ports. Étant donné que les nœuds simulés sont des commutateurs virtuels *Openvswitch* (Pfaff *et al.* (2015)) des commandes Unix sont utilisées (voir figure III-6) pour simuler la limitation et l'augmentation des débits après un changement d'état

d'une liaison. Pour le reroutage des flux l'outil *Curl* (voir figure III-5) est utilisé pour envoyer des requêtes de suppression, ajout et mises à jour des chemins des flux à travers l'API fournie par le contrôleur SDN.

3.5 Conclusion

Dans ce chapitre, différents scénarios d'expérimentations ont été testés dans le but de mesurer la performance des algorithmes proposés. Le premier scénario s'est concentré sur la mesure de la performance des approches heuristiques et métaheuristiques en termes d'économie d'énergie et de temps de traitement. Les résultats de ce scénario ont démontré la supériorité de l'AG en matière d'économie d'énergie par rapport aux autres algorithmes notamment dans les réseaux qui présentent de fortes demandes.

Le deuxième scénario s'est focalisé sur la résolution du modèle d'économie d'énergie avec l'outil Cplex. Dans ce scénario, différentes matrices de trafic ont été générées pour simuler différentes variations de trafic. Les résultats obtenus prouvent que le nombre de flux généré affecte le temps de traitement plus que la variation de la charge des flux. En effet, le temps de traitement augmente de façon exponentielle avec l'augmentation du nombre de flux et présente un comportement linéaire avec l'augmentation de la charge de trafic.

Les résultats des deux premiers scénarios ont démontré inefficacité en matière de temps de traitement et d'économie d'énergie de l'approche formelle et l'approche heuristique respectivement. De ce fait, uniquement l'AG a été sélectionné pour le troisième scénario.

Dans le dernier scénario, l'AG a été évalué sous les contraintes de reconfigurations de routage, taux d'économie d'énergie et temps de traitement. Les résultats de ce scénario ont démontré que l'AG est très efficace en matière de temps de traitement et qu'il présente un taux d'économie d'énergie comparable à celui de l'approche formelle. En effet il présente un temps de calcul inférieur à 1 ms et un taux d'économie égal à plus de 47 % dans certains cas.

Pour finir, ce chapitre a introduit un cas d'étude qui représente une des implémentations possibles de l'AG dans un environnement SDN.

CHAPITRE 4

CONCLUSION

Au cours de la dernière décennie, la question de l'efficacité énergétique au niveau des réseaux de communications a reçu une attention considérable en raison de ses impacts économiques et environnementaux. Cependant, la reconfiguration des réseaux traditionnels pour mettre en œuvre des politiques écoénergétiques reste très difficile en raison de l'intégration verticale du plan de contrôle et de données à chaque dispositif de mise en réseau.

Pour simplifier cette complexité et permettre l'innovation au niveau des réseaux de communication, un nouveau paradigme a été développé. Le SDN ou réseaux définis par logiciel est un paradigme émergent qui découple le plan de contrôle du plan de données et introduit la programmabilité du réseau pour le développement d'applications réseau. Ainsi, il devient évident que le paradigme SDN ouvre de nouvelles opportunités prometteuses pour améliorer la performance du réseau en général et l'efficacité énergétique en particulier.

Dans ce mémoire, nous avons tenté de répondre au problème d'optimisation de la consommation énergétique des réseaux définis par logiciel à travers le routage. Pour y parvenir, nous avons commencé par étudier, analyser et comparer dans le chapitre 2 les différentes stratégies et techniques existant dans le champ de recherches relatives au sujet de ce mémoire, notamment le *Green Networking* et ce pour les réseaux traditionnels et les réseaux définis par logiciel.

Cette étude de littérature, nous a permis de connaître les critères de classification des différentes stratégies d'économie d'énergie et de là, à choisir une stratégie bien précise. La stratégie choisie est le routage écoconscient appliqué aux réseaux définis par logiciel. Cette strate agit en ligne au niveau global du réseau. En effet, elle exploite la vue globale du réseau fournie par le contrôleur SDN afin de réaliser un reroutage des flux en tenant compte des contraintes de bout en bout du réseau. La stratégie choisie est couplée avec la technique *Link Rate Switching*. Cette technique permet de réduire le débit de transmission lorsque le lien est sous-utilisé. Elle est

basée sur le fait que les cartes d'interface réseau consomment une plus petite quantité d'énergie lorsqu'elles transmettent des débits plus faibles.

À partir de là nous avons pu établir dans notre méthodologie l'objectif principal à atteindre pour répondre au problème posé. L'objectif principal est de proposer un algorithme de placement dynamique des flux de façon à minimiser l'utilisation maximale des liaisons actives dans le but de minimiser la consommation énergétique dans un réseau défini par logiciel.

Nous avons formulé le problème d'optimisation de la consommation d'énergie des réseaux définis par logiciel comme un problème ILP connu pour être NP-difficile. Ensuite nous avons proposé des solutions pour minimiser la consommation d'énergie du réseau tout en satisfaisant les exigences du réseau en matière de reroutage et transition des liaisons. Les approches proposées sont une approche heuristique représentée par les algorithmes Firsts Fit, Worst fit et Best Fit, une approche métaheuristique représentée par l'algorithme génétique et une solution exacte réalisée avec l'outil Cplex.

Pour évaluer les performances des solutions proposées, nous avons effectué dans le chapitre 4 différents scénarios de simulations approfondies pour évaluer les algorithmes proposés sur des topologies de réseau bien connues Abilene et NSF et une matrice de trafic réelle.

Les résultats de la simulation montrent que l'approche métaheuristique basée sur l'AG surpasse systématiquement les autres solutions proposées et qu'en appliquant cette solution, entre 32 % et 47 % de l'énergie peut être économisée en fonction de la taille et de la densité de la topologie du réseau et ceux en respectant les contraintes posées en matière de reconfiguration des états de liaison et reroutage des flux.

À la fin de ce projet, après avoir évalué les performances des algorithmes, nous avons proposé une implémentation possible de l'AG dans un environnement SDN. L'architecture fonctionnelle de cette implémentation est composée de trois modules logiques ; le module de collecte et traitement des données, le module d'optimisation et le module de contrôle.

Travaux futurs

Dans le but d'affiner les résultats en matière de respect des contraintes de reroutage et de reconfiguration des états des liaisons, nous envisageons dans le futur d'utiliser la prédiction dans notre processus d'optimisation de l'énergie du réseau.

La prédiction servira à générer plusieurs matrices de trafic à partir du trafic historique. Ces matrices de trafic représentent le trafic futur à long terme et serviront à prévoir les pics de trafic prévus. Ceci permettra de réaliser un meilleur compromis entre l'économie d'énergie réalisée et la qualité de service (QoS) qui est régi par les contraintes de reroutage, de reconfiguration des états des liaisons et de capacité de réserve.

D'autre part, l'erreur de prédiction du trafic ne peut être égale à zéro et elle doit être prise en compte dans le processus d'allocation des flux. Autrement dit, la performance de l'algorithme ne doit pas reposer entièrement sur la précision de la prédiction. Par conséquent, nous envisageons d'intégrer l'incertitude de prédiction dans le processus d'optimisation en exploitant la distribution de probabilité estimée des valeurs prédites.

ANNEXE I

ALGORITHME GÉNÉTIQUE (AG)

Un algorithme génétique (AG) est une méthode pour résoudre à la fois des problèmes d'optimisation avec ou sans contraintes. Ils sont basés sur un processus de sélection naturelle qui imite l'évolution biologique. : les chromosomes sont les éléments à partir desquels sont générées les solutions ou «individus». Le regroupement de chromosomes forme une population et leurs combinaisons représentent l'étape de reproduction, qui est réalisée à l'aide de l'opérateur de croisement et/ou l'opérateur de mutation. D'autres notions sont propres aux AG telles que l'indice de performance (fitness), qui est une mesure permettant d'évaluer les chromosomes. Il en va de même pour la fonction de coût qui représente la formule théorique permettant de calculer l'indice de performance d'un chromosome.

La Figure I-1 illustre le fonctionnement itératif simplifié d'un AG. En premier lieu, une population initiale $P(0)$ est construite. Ensuite, itérativement, un calcul du fitness $f(i)$ de chacun des individus est effectué, puis l'algorithme procède à une sélection des parents pour la reproduction à travers le croisement et la mutation. Ce processus est répété jusqu'à ce qu'un critère d'arrêt soit atteint. Le remplacement représente l'insertion des nouveaux individus générés, appelés enfants, dans la population en cours de traitement. Il s'agit donc du processus de passage de la population $P(t)$ à un instant t à la population $P(t+1)$.

Les AG sont utilisés pour résoudre un grand nombre de problèmes d'optimisation combinatoire, tels le problème du voyageur de commerce asymétrique (Choi *et al.* (2003)), le problème de partitionnement (Chu et Beasley (1998)), la fouille des données (Szpiro (1997)), le séquençage de protéines d'ADN (Boisson et Rolando (2006)), les problèmes de minimisation de fonctions convexes (Hussain et Al-Sultan (1997)) pour ne nommer que ceux-là.

La représentation

La représentation doit être globale, c'est-à-dire que l'ensemble des solutions possibles au problème doivent pouvoir être codifiées par la représentation établie. De plus, toutes les solutions doivent correspondre à des solutions réalisables.

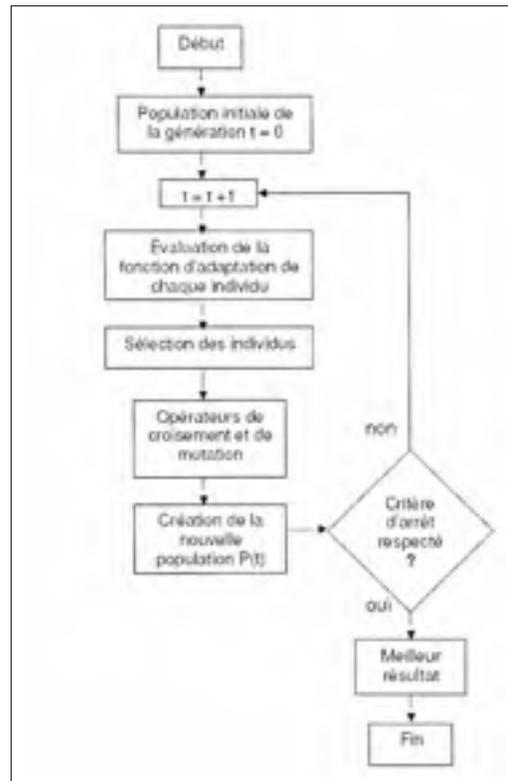


Figure-A I-1 Organigramme représentant le fonctionnement général d'un algorithme génétique

Tout d'abord, le codage doit contenir des blocs d'information ayant un sens. Ensuite, l'ensemble des symboles ou alphabets utilisés pour ce codage, doivent permettre une expression naturelle du problème et ne doit pas contenir des redondances. En effet, chaque symbole d'une solution, qui peut être une lettre ou un entier par exemple, est assimilé à un gène, et toutes les valeurs qu'il peut prendre sont les allèles de ce gène. Une suite de gènes forme un chromosome. Chaque individu est représenté par un ensemble de chromosomes, et une population est un ensemble de chromosomes.

Historiquement, le codage utilisé par les algorithmes génétiques est représenté sous forme de chaînes de bits contenant l'information nécessaire à la description d'une solution probable dans l'espace de recherche. Ce type de codage a pour objectif de permettre de créer des opérateurs de croisement et de mutation simples. Cependant, ce type de codage a rapidement montré ses limites. En effet, dans le cas d'un codage binaire, la distance de Hamming est souvent utilisée

comme mesure de la dissimilarité entre deux individus de la population, et cette mesure compte les différences de bits de même rang de ces deux séquences. C'est toutefois là que le codage binaire commence à montrer ses limites. En effet, deux éléments voisins en termes de distance de Hamming ne correspondent pas nécessairement à deux éléments proches dans l'espace de recherche.

De même, en utilisant un codage binaire, les performances de l'algorithme sont diminuées lorsque la longueur de la chaîne augmente. En effet, chaque variable est représentée par une partie de la chaîne de bits et la structure du problème n'est pas bien reflétée. Dans ce cas, l'ordre des variables a une importance dans la structure du chromosome, alors qu'il n'en a pas forcément dans la structure du problème. Les représentations qui utilisent des vecteurs réels (Goldberg *et al.* (1989)) détournent ce problème en conservant les variables du problème dans le codage de l'élément de population sans passer par le codage binaire intermédiaire. De ce fait, la structure du problème est conservée dans le codage.

La fitness

La *fitness*, ou encore fonction d'évaluation, quantifie la qualité de chaque chromosome par rapport au problème. Elle est généralement utilisée pour sélectionner les chromosomes pour la reproduction. Les chromosomes ayant une bonne qualité ont alors plus de chances d'être sélectionnés pour la reproduction, faisant en sorte que la prochaine génération de la population hérite de leur matériel génétique.

Population initiale

La population initiale représente la population de départ de l'AG. Elle regroupe des chromosomes qui sont répartis dans l'espace des solutions pour fournir à l'AG une base génétique variée. Le choix de la population initiale d'individus est très important, car elle affecte fortement la convergence de l'algorithme. Étant donné que l'optimum est totalement inconnu dans l'espace de solutions, il est naturel et nécessaire de générer des individus aléatoirement en procédant à des tirages uniformes dans l'espace de solutions et en veillant à ce que les individus choisis respectent les contraintes pour demeurer dans cet espace Shayan et Chittilappilly (2004). Dans certains cas, la génération de solutions initiale aléatoirement produit des solutions

infaisables, qui ne satisfont pas les contraintes du problème traité. En revanche, dans le cas où des informations sur le problème sont disponibles a priori, il paraît naturel et évident de générer les individus dans un sous-ensemble particulier afin d'accélérer la convergence de l'algorithme. Il est alors avantageux d'utiliser des méthodes de génération d'individus pseudo-aléatoires ou même voraces en utilisant les caractéristiques du problème. Dans ce dernier cas, favoriser le placement d'un ou de plusieurs allèles à des positions qui sont supposées prometteuses dans un chromosome de la population initiale peut être une bonne méthode.

L'opérateur de sélection

Il existe plusieurs méthodes de sélection. Nous décrivons, dans ce qui suit, les deux méthodes utilisées dans le chapitre suivant : la sélection par la fitness et la sélection par tournoi.

- la sélection proportionnelle

La sélection proportionnelle, ou sélection par fitness, est en général implémentée comme une roulette biaisée (Goldberg *et al.* (1989)). Chaque individu est représenté par une partie de la roue de taille proportionnelle à son fitness autrement dit à son adaptation au problème. Ainsi, pour un problème de maximisation, un individu "c", a la probabilité suivante d'être sélectionné :

$$P_{\text{Sélection}}(c_i) = \frac{F_{\text{évaluation}}(c_i)}{\sum_{j=1}^n F_{\text{évaluation}}(c_j)}$$

où "n" représente la taille de la population de l'AG.

Cette méthode a l'avantage d'être simple et intuitive, mais elle présente certains inconvénients (Deb et Beyer (1999)). Par exemple, elle requiert un temps de calcul important pour calculer la somme des fitness et une forte variance. Il n'est pas impossible que, sur n sélections successives destinées à désigner les parents pour la phase de reproduction, la quasi-totalité, voire la totalité des n individus sélectionnés, soient des individus ayant une fitness très mauvaise. Cela fait

en sorte que pratiquement aucun individu avec un bon fitness ne fait partie des parents. Ce phénomène va complètement à l'encontre du principe des algorithmes génétiques qui veut que les meilleurs individus soient sélectionnés de sorte à converger vers la meilleure solution possible.

Dans le cas inverse ou une la grande majorité des individus sont localement supérieur ou très bon en termes de fitness, ceci va entraîner une grave perte de diversité. Par exemple, dans le cas d'individu ayant un fitness dix fois supérieur par rapport au reste de la population ; il n'est pas impossible qu'après quelques générations successives la population ne contienne que des copies de cet individu. Certes, cet individu a un très fitness, mais relativement avec les autres individus. Ce problème est connu sous comme étant un problème de convergence prématurée, où l'évolution se met à stagner et bloque sur un optimum local.

- la sélection par tournoi

Dans cette méthode de sélection, K individus dans la population sont sélectionnés de la manière suivante : K individus sont tirés aléatoirement parmi une population de n individus, où K représente la taille du tournoi. Il existe deux types de sélection par tournoi, déterministe et probabiliste. Dans le cas du tournoi déterministe, le meilleur des K individus remporte le tournoi. Dans le cas probabiliste, chaque individu a une probabilité d'être choisi comme vainqueur. Cette probabilité est proportionnelle à sa fonction d'évaluation. Il est possible que certains individus participent à plusieurs tournois. S'ils gagnent plusieurs fois, ils pourront participer à la phase de reproduction plusieurs fois, ce qui favorise la pérennité de leurs gènes.

- la sélection par rang

Cette technique de sélection choisit toujours les individus possédant les meilleurs fitness. D'une manière plus parlante, il faut trier la population en fonction de la qualité des individus puis attribuer à chacun un rang. Les individus de moins bonne qualité obtiennent un rang faible. Et ainsi en itérant sur chaque individu on finit par attribuer le plus haut rang au meilleur individu. La suite de la méthode consiste uniquement en l'implémentation d'une roulette basée

sur les rangs des individus. L'angle de chaque secteur de la roue sera proportionnel au rang de l'individu qu'il représente.

Il existe quelques autres méthodes de sélection moins utilisées étudiées et discutées dans (Back et Michalewicz (2000)).

L'opérateur de croisement (*crossover*) est la transposition informatique du mécanisme qui permet la production de chromosomes qui héritent partiellement des caractéristiques des parents. Son rôle fondamental est de permettre la recombinaison des informations présentes dans le patrimoine génétique de la population. Avec l'opérateur de mutation, ils représentent la phase de reproduction. C'est après avoir appliqué l'opérateur de sélection sur la population courante $P(t)$ que vient la phase de croisement. Ainsi, un ou plusieurs parents sélectionnés permettent de générer un ou plusieurs enfants. En général les croisements sont envisagés avec deux parents qui génèrent deux enfants. Les individus générés sont ensuite mis à la disposition de l'opérateur de mutation puis celui du remplacement, ou directement à celui du remplacement pour former la population $(t+1)$. Le principal but du croisement est d'enrichir la diversité de la population en manipulant la structure des chromosomes. La probabilité de croisement représente la fréquence à laquelle les croisements sont appliqués. Ainsi on peut déduire que si la probabilité de croisement est de 0%, la nouvelle génération sera la copie de la précédente et que si la probabilité est égale à 100%, tous les antécédents seront générés par croisement.

Il existe plusieurs techniques de croisement qu'on retrouve dans la littérature, nous pouvons citer par exemple :

- croisement à un point de coupure (simple) (figure I-2);
- croisement à multipoint de coupure (figure I-3);
- croisement uniforme;
- croisement arithmétique, etc.

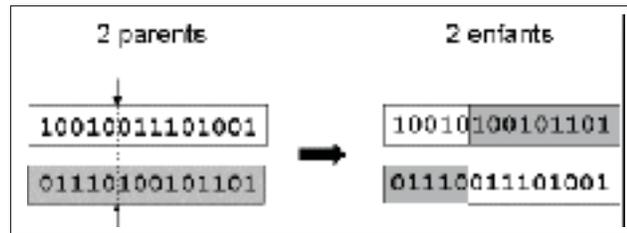


Figure-A I-2 Croisement à un point de coupure

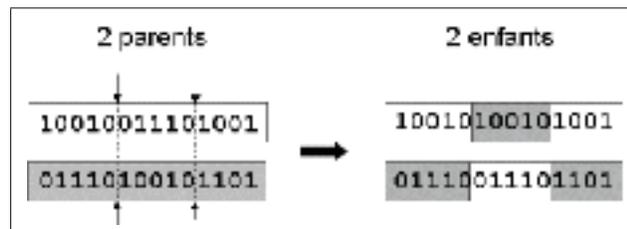


Figure-A I-3 Croisement à multipoint de coupure

L'opérateur de mutation

Cet opérateur apporte aux AG l'ergodicité de parcours de l'espace des solutions. La propriété d'ergodicité indique que l'algorithme génétique est susceptible d'atteindre tous les points de l'espace de recherche sans tous les parcourir pour autant. La convergence des algorithmes génétiques est donc fortement dépendante de cet opérateur sur le plan théorique. La mutation est aussi prévue pour éviter à l'AG de s'enliser dans des optima locaux, mais si elle est trop fréquente, l'algorithme est orienté vers une recherche aléatoire de la bonne solution. La mutation consiste à changer la valeur allélique d'un gène avec une probabilité très faible P_m , comprise généralement entre 0.01 et 0.001 (Back et Michalewicz (2000)). Dans le cas d'un codage binaire, une mutation peut consister simplement en l'inversion d'un bit ou plusieurs bits se trouvant dans une position particulière et également déterminée de manière aléatoire. Dans le cas des problèmes discrets, l'opérateur de mutation consiste généralement à tirer aléatoirement un gène dans le chromosome et à le remplacer par une valeur aléatoire. Si la notion de distance existe, cette valeur peut être choisie dans le voisinage de la valeur initiale. Quant à la mutation par échange, elle consiste à sélectionner de manière aléatoire deux gènes d'un individu et à échanger les positions respectives des deux éléments choisis. Tout comme le croisement,

plusieurs techniques existent dans la littérature pour la mutation ; nous citons, par exemple, la mutation aléatoire (figure I-4)

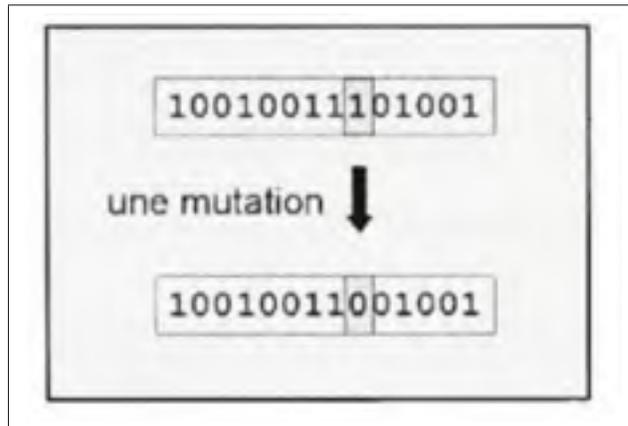


Figure-A I-4 Représentation de la mutation

L'opérateur de remplacement

Cette dernière étape consiste à incorporer de nouvelles solutions dans la population courante. Les nouvelles solutions sont ajoutées à la population courante par remplacement total ou partiel des solutions existantes. Généralement, les meilleures solutions remplacent les plus mauvaises et il en résulte une amélioration de la population. Lorsque la nouvelle population n'est constituée que des nouvelles solutions, l'AG est dit générationnel. Le remplacement peut également se formaliser par l'utilisation de stratégies appelées $(v + \lambda)$ ou (v, λ) où v correspond au nombre de parents utilisés pour générer λ enfants. Dans le schéma (v, λ) avec $(v > \lambda)$, les v meilleurs enfants deviennent les parents de la génération suivante, alors que dans le schéma $(v + \lambda)$, ce sont les v meilleurs des $v + \lambda$ parents plus enfants qui survivent.

Critères d'arrêt

Il est évident qu'un AG ne peut évoluer indéfiniment. Nous pouvons citer différents critères d'arrêt comme le temps, un nombre déterminé de générations, un nombre fixé d'évaluations, une valeur particulière de la fonction objective, etc. L'intérêt des AG est de produire des solutions diversifiées et de qualité. Afin de les appliquer à un problème particulier, il est nécessaire de définir convenablement la fonction d'évaluation et la représentation des individus. Ces al-

gorithmes nécessitent également la définition d'opérateurs, aussi bien de mutation que de croisement, en adéquation avec le problème. Dans de nombreux cas, ils peuvent être difficiles à paramétrer. En général les critères d'arrêt sont de deux natures :

- a. Arrêt, après un nombre de générations fixé a priori. C'est la solution retenue lorsqu'un impératif de temps de calcul est imposé.
- b. Arrêt, lorsque la population cesse d'évoluer ou n'évolue plus suffisamment rapidement, on est alors en présence d'une population homogène dont on peut penser qu'elle se situe à proximité du ou des optimums.

ANNEXE II

OUTILS DE SUPERVISION ET CONTRÔLE DU RÉSEAU

1. Outil de supervision des flux : sFlow

Avec la dépendance croissante des services réseau pour les applications stratégiques, le plus petit changement dans l'utilisation du réseau peut avoir un impact sur les performances et la fiabilité du réseau. En fournissant une visibilité sans précédent sur l'utilisation du réseau et sur les routes actives des réseaux complexes et ultrarapides d'aujourd'hui, sFlow fournit les données nécessaires au contrôle et à la gestion efficace de l'utilisation du réseau, garantissant ainsi un avantage concurrentiel. C'est pourquoi dans notre banc d'essai il représentera notre outil de surveillance des flux.

sFlow est une technologie d'échantillonnage multifournisseur intégrée dans les commutateurs et les routeurs qui répond aux principales exigences d'une solution de surveillance du trafic réseau suivante (sflow.org (2003)) :

- il fournit une vue d'ensemble du réseau de l'utilisation et des itinéraires actifs. Il s'agit d'une technique évolutive pour mesurer le trafic réseau, collecter, stocker et analyser les données de trafic. Cela permet de contrôler des dizaines de milliers d'interfaces réseau à partir d'un seul emplacement ;
- il est évolutif, ce qui lui permet de surveiller des liaisons allant jusqu'à 10 Gb / s et au-delà sans affecter les performances des principaux routeurs et commutateurs, sans ajouter de charge réseau importante ;
- il représente une solution économique. Il a été implémenté sur une large gamme d'équipement des simples commutateurs couche 2 à des routeurs haut de gamme, et cela sans nécessiter de mémoire supplémentaire ni de traitement CPU ;
- sFlow est une norme de l'industrie avec un nombre croissant de fournisseurs livrant des produits avec le support sFlow.

- **agent sFlow**

L'agent sFlow est un processus logiciel exécuté faisant partie des logiciels de gestion de réseau au sein d'un équipement (voir figure II-1) (sflow.org (2003)). Il combine des compteurs d'interface et des échantillons de flux dans des datagrammes sFlow envoyés sur le réseau à un collecteur sFlow. L'état des entrées de la table de transfert associées à chaque paquet échantillonné est également enregistré. L'agent sFlow effectue très peu de traitement. Il permet simplement de stocker des données dans des datagrammes sFlow qui sont immédiatement envoyés sur le réseau. L'envoi immédiat des données minimise les besoins en mémoire et CPU associés à l'agent sFlow.

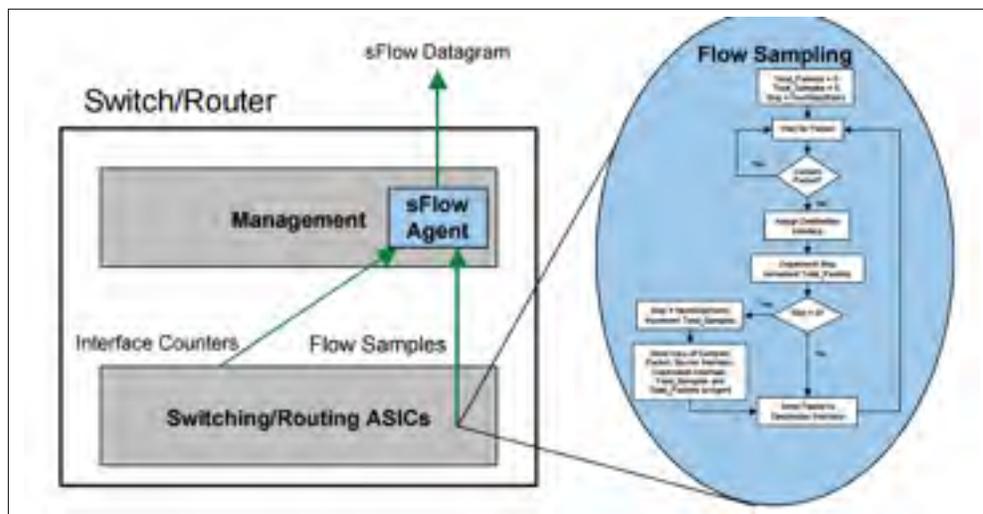


Figure-A II-1 Agent sFlow intégré dans le commutateur /
routeur
Tirée de sflow.org (2003)

- **données sFlow**

L'agent sFlow utilise deux types d'échantillonnage (Arista (2008)) : l'échantillonnage statistique par paquets de flux et l'échantillonnage basé sur le temps des statistiques d'interface réseau. L'échantillonnage est Utilisé pour réduire le nombre de paquets qui sont sélectionnés pour l'analyse. Il est appliqué au niveau du sFlow agent pour réduire la surcharge car le nombre

de paquets que l'agent sFlow doit analyser est réduit. Un datagramme sFlow intègre les deux types d'échantillons.

- échantillonnage des flux : Un échantillon est prélevé en copiant l'en-tête du paquet ou en extrayant les données des caractéristiques du paquet ;
- échantillonnage des statistiques d'interface : le comptage d'échantillons extrait des statistiques en interrogeant périodiquement chaque source de données sur l'appareil.

- **collecteur sFlow**

Le collecteur sFlow est un composant matériel / logiciel qui reçoivent respectivement des échantillons de flux et compteurs d'échantillons en tant que datagrammes sFlow des agents sFlow (sflow.org (2003)). Cela leur permet de présenter une vue du trafic et d'autres paramètres de réseau (voir figure II-2).

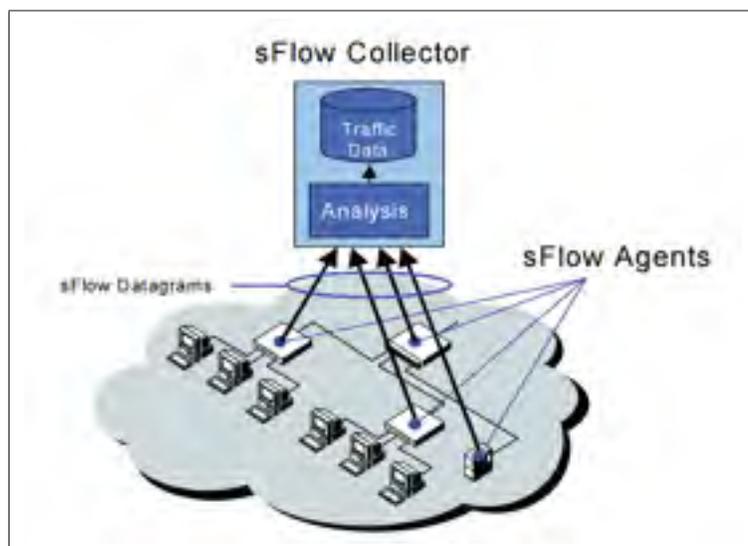


Figure-A II-2 Agent et collecteur sFlow
Tirée de sflow.org (2003)

Outil de contrôle : Opendaylight (ODL)

OpenDaylight est un projet logiciel open source créé en 2013 sous la supervision de la "Linux Foundation". Il a été créé dans le but de favoriser l'adoption et l'innovation du SDN (Software Defined Networking) par la création d'une plateforme modulaire commune supportée par l'industrie. La première version de Opendaylight est sortie février 2014 suivie de plusieurs autres versions illustres dans la figure II-3.

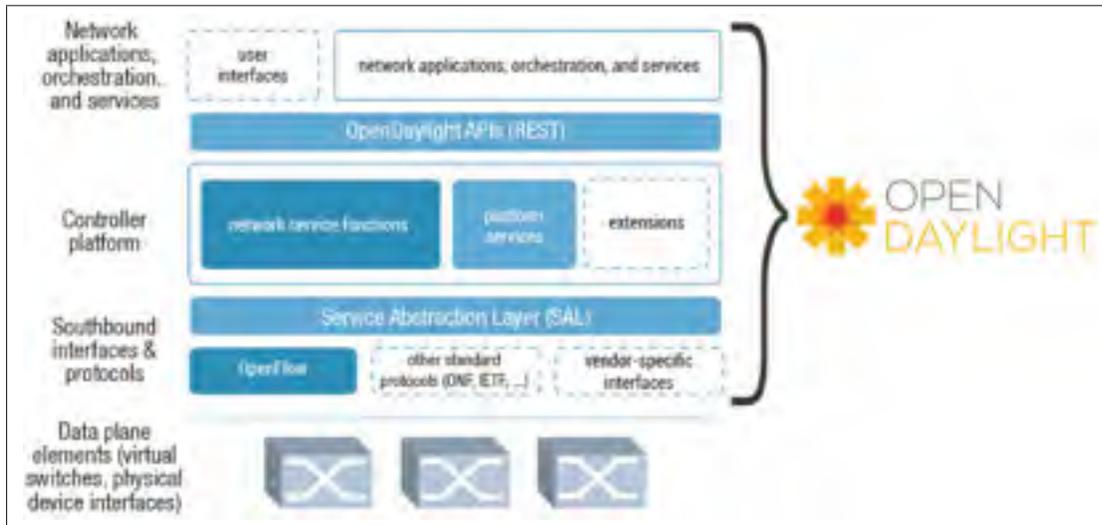


Figure-A II-3 Contrôleur SDN : Opendaylight
Tirée de opendaylight.org (2017)

Le contrôleur OpenDaylight est un logiciel JVM et peut être exécuté à partir de n'importe quel système d'exploitation et matériel, à condition qu'il supporte Java. Le contrôleur est une implémentation du concept (SDN) et utilise les outils suivants :

Maven : OpenDaylight utilise Maven pour une automatisation plus facile. Maven utilise pom.xml (Project Object Model) pour scripter les dépendances entre modules et aussi pour décrire les modules à charger et exécuter.

OSGi : Ce framework est l'arrière-plan d'OpenDaylight, car il permet de charger dynamiquement les modules et les fichiers JAR et de regrouper des modules ensemble pour échanger des informations.

REST API : Les interfaces RESTful sont conçues sur la base du protocole RESTCONF. RESTCONF prend en charge deux formats d'encodage : XML et JSON. Ces interfaces sont générées dynamiquement au moment de l'exécution en fonction des modèles YANG qui définissent ses données, Parmi les API exposées par le contrôleur on retrouve : topology manager, host tracker, flow programmer, static routing , etc.

Pour que le contrôleur puisse contrôler les équipements de son domaine, il doit connaître les équipements concernés, leurs capacités, leurs accessibilités, etc. Dans les réseaux SDN basés sur le protocole openFlow, les commutateurs envoient les messages LLDP (voir figure II-4) pour découvrir la topologie sous-jacente à la demande du contrôleur. Ces informations sont stockées et gérées par le gestionnaire de topologie (topologie manager). Les autres composants comme le ARP handler, Host Tracker, Device Manager et le Switch Manager aident à générer la base de données de topologie pour le gestionnaire de topologie.

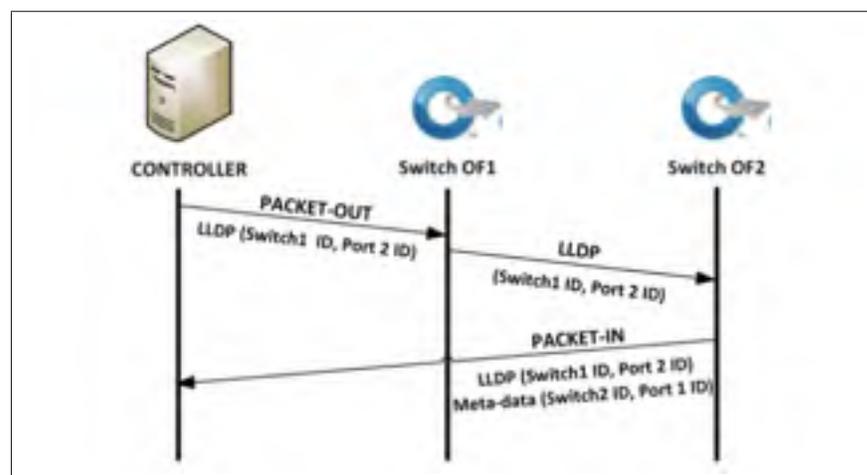


Figure-A II-4 Messages échangés entre le contrôleur et les commutateurs OF pour la découverte de topologie basée sur le protocole LLDP

Tirée de Ochoa-Aday *et al.* (2015)


```

<network-topology xmlns="urn:TBD:params:xml:ns:yang:network-topology">
  <topology>...</topology>
  <topology>...</topology>
  <topology>
    <topology-id>flow :1</topology-id>
    <node>
      <node-id>openflow :1</node-id>
      <inventory-node-ref xmlns="urn:opendaylight:model:topology:inventory"
xmlns:a="urn:opendaylight:inventory">/a:nodes/a:node[a:id='openflow :1']</inventory-
node-ref>
      <termination-point>
        <tp-id>openflow :1 :1</tp-id>
        <inventory-node-connector-ref xmlns="urn:opendaylight:model:topology:inventory"
xmlns:a="urn:opendaylight:inventory"> /a:nodes/a:node[a:id='openflow :1']/a:node-
connector[a:id='openflow :1 :1']
      </inventory-node-connector-ref>
    </termination-point>
    <termination-point>
      <tp-id>openflow :1 :2</tp-id>
      <inventory-node-connector-ref xmlns="urn:opendaylight:model:topology:inventory"
xmlns:a="urn:opendaylight:inventory"> /a:nodes/a:node[a:id='openflow :1']/a:node-
connector[a:id='openflow :1 :2']
    </inventory-node-connector-ref>
    </termination-point>
    <termination-point>
      <tp-id>openflow :1 :LOCAL</tp-id>
      <inventory-node-connector-ref xmlns="urn:opendaylight:model:topology:inventory"
xmlns:a="urn:opendaylight:inventory"> /a:nodes/a:node[a:id='openflow :1']/a:node-
connector[a:id='openflow :1 :LOCAL']

```

```

</inventory-node-connector-ref>
</termination-point>
</node>
<node>...</node>
<link>
<link-id>openflow :2 :1</link-id>
<destination>
<dest-tp>openflow :1 :2</dest-tp>
<dest-node>openflow :1</dest-node>
</destination>
<source>
<source-tp>openflow :2 :1</source-tp>
<source-node>openflow :2</source-node>
</source>
</link>
<link>...</link>
</topology>
</network-topology>

```

- simulation de la topologie

Pour la simulation de la topologie l'émulateur Mininet (Mininet (2014)) a été utilisé. l'émulateur Mininet permet des créer une topologie virtuelle et de connecter les noeuds au contrôleur SDN. La figure III-2 illustre la topologie NSF simulée avec Mininet.

Le script suivant sert à générer la topology NSF dans le simulateur Mininet :

```

#!/usr/bin/python
from mininet.net import Mininet
from mininet.node import Controller, RemoteController, OVSController

```

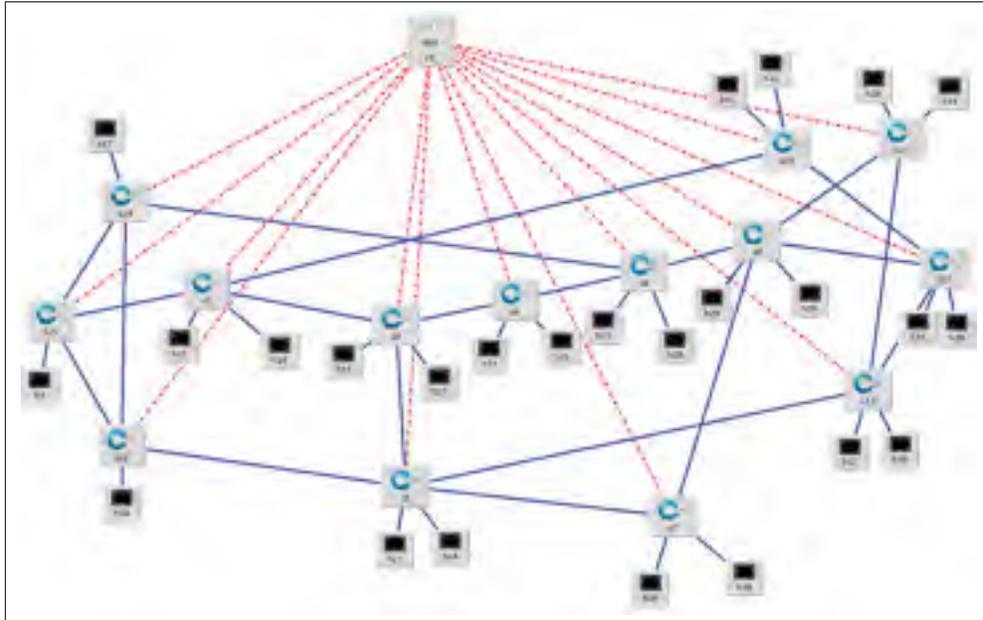


Figure-A III-2 Topologie NSF simulée avec l'outil Mininet

```

from mininet.node import CPULimitedHost, Host, Node
from mininet.node import OVSKernelSwitch, UserSwitch
from mininet.node import IVSSwitch
from mininet.cli import CLI
from mininet.log import setLogLevel, info
from mininet.link import TCLink, Intf
from subprocess import call

def myNetwork():
    net = Mininet( topo=None, build=False, ipBase='10.0.0.0/8')
    info( '*** Adding controller' )
    c0=net.addController(name='c0', controller=RemoteController, ip='127.0.0.1', protocol='tcp',
    port=6633)
    info( '*** Add switches')
    s10 = net.addSwitch('s10', cls=OVSKernelSwitch)
    s3 = net.addSwitch('s3', cls=OVSKernelSwitch)
    s13 = net.addSwitch('s13', cls=OVSKernelSwitch)

```

```
s8 = net.addSwitch('s8', cls=OVSKernelSwitch)
s5 = net.addSwitch('s5', cls=OVSKernelSwitch)
s7 = net.addSwitch('s7', cls=OVSKernelSwitch)
s4 = net.addSwitch('s4', cls=OVSKernelSwitch)
s12 = net.addSwitch('s12', cls=OVSKernelSwitch)
s6 = net.addSwitch('s6', cls=OVSKernelSwitch)
s2 = net.addSwitch('s2', cls=OVSKernelSwitch)
s11 = net.addSwitch('s11', cls=OVSKernelSwitch)
s1 = net.addSwitch('s1', cls=OVSKernelSwitch)
s9 = net.addSwitch('s9', cls=OVSKernelSwitch)
s14 = net.addSwitch('s14', cls=OVSKernelSwitch)
info( '*** Add hosts' )
h33 = net.addHost('h33', cls=Host, ip='10.0.0.33', defaultRoute=None)
h22 = net.addHost('h22', cls=Host, ip='10.0.0.22', defaultRoute=None)
h20 = net.addHost('h20', cls=Host, ip='10.0.0.20', defaultRoute=None)
h21 = net.addHost('h21', cls=Host, ip='10.0.0.21', defaultRoute=None)
h23 = net.addHost('h23', cls=Host, ip='10.0.0.23', defaultRoute=None)
h1 = net.addHost('h1', cls=Host, ip='10.0.0.1', defaultRoute=None)
h26 = net.addHost('h26', cls=Host, ip='10.0.0.26', defaultRoute=None)
h30 = net.addHost('h30', cls=Host, ip='10.0.0.30', defaultRoute=None)
h36 = net.addHost('h36', cls=Host, ip='10.0.0.36', defaultRoute=None)
h13 = net.addHost('h13', cls=Host, ip='10.0.0.13', defaultRoute=None)
h35 = net.addHost('h35', cls=Host, ip='10.0.0.35', defaultRoute=None)
h15 = net.addHost('h15', cls=Host, ip='10.0.0.15', defaultRoute=None)
h16 = net.addHost('h16', cls=Host, ip='10.0.0.16', defaultRoute=None)
h29 = net.addHost('h29', cls=Host, ip='10.0.0.29', defaultRoute=None)
h14 = net.addHost('h14', cls=Host, ip='10.0.0.14', defaultRoute=None)
h31 = net.addHost('h31', cls=Host, ip='10.0.0.31', defaultRoute=None)
h24 = net.addHost('h24', cls=Host, ip='10.0.0.24', defaultRoute=None)
```

```
h17 = net.addHost('h17', cls=Host, ip='10.0.0.17', defaultRoute=None)
h38 = net.addHost('h38', cls=Host, ip='10.0.0.38', defaultRoute=None)
h27 = net.addHost('h27', cls=Host, ip='10.0.0.27', defaultRoute=None)
h32 = net.addHost('h32', cls=Host, ip='10.0.0.32', defaultRoute=None)
h37 = net.addHost('h37', cls=Host, ip='10.0.0.37', defaultRoute=None)
h25 = net.addHost('h25', cls=Host, ip='10.0.0.25', defaultRoute=None)
h28 = net.addHost('h28', cls=Host, ip='10.0.0.28', defaultRoute=None)
h34 = net.addHost('h34', cls=Host, ip='10.0.0.34', defaultRoute=None)
h19 = net.addHost('h19', cls=Host, ip='10.0.0.19', defaultRoute=None)
h2 = net.addHost('h2', cls=Host, ip='10.0.0.2', defaultRoute=None)
info( '*** Add links' )
net.addLink(h17, s14)
net.addLink(h13, s2)
net.addLink(s3, h15)
net.addLink(s3, h16)
net.addLink(s4, h19)
net.addLink(s4, h20)
net.addLink(s6, h21)
net.addLink(s6, h22)
net.addLink(s8, h27)
net.addLink(s8, h28)
net.addLink(s9, h30)
net.addLink(s9, h29)
net.addLink(s12, h37)
net.addLink(s12, h38)
net.addLink(s7, h25)
net.addLink(s7, h26)
net.addLink(s5, h23)
net.addLink(s5, h24)
```

```
net.addLink(s13, h35)
net.addLink(s13, h36)
net.addLink(s11, h33)
net.addLink(s11, h34)
net.addLink(s10, h31)
net.addLink(s10, h32)
net.addLink(s1, s14)
net.addLink(s1, s3)
net.addLink(s14, s8)
net.addLink(s3, s4)
net.addLink(s4, s6)
net.addLink(s6, s8)
net.addLink(s4, s5)
net.addLink(s5, s7)
net.addLink(s5, s12)
net.addLink(s12, s13)
net.addLink(s12, s11)
net.addLink(s8, s9)
net.addLink(s9, s11)
net.addLink(s9, s13)
net.addLink(s9, s7)
net.addLink(s3, s10)
net.addLink(s10, s13)
net.addLink(s1, s2)
net.addLink(s2, s5)
net.addLink(h1, s1)
net.addLink(s1, h2)
net.addLink(s2, h14)
info( '*** Starting network')
```

```
net.build()
info( '*** Starting controllers')
for controller in net.controllers :
    controller.start()
info( '*** Starting switches')
net.get('s10').start([c0])
net.get('s3').start([c0])
net.get('s13').start([c0])
net.get('s8').start([c0,c0])
net.get('s5').start([c0])
net.get('s7').start([c0,c0])
net.get('s4').start([c0])
net.get('s12').start([c0])
net.get('s6').start([c0,c0])
net.get('s2').start([c0])
net.get('s11').start([c0])
net.get('s1').start([c0])
net.get('s9').start([c0,c0])
net.get('s14').start([c0])
info( '*** Post configure switches and hosts')
s10.cmd('ifconfig s10 192.168.0.110')
s3.cmd('ifconfig s3 192.168.0.13')
s13.cmd('ifconfig s13 192.168.0.113')
s8.cmd('ifconfig s8 192.168.0.18')
s5.cmd('ifconfig s5 192.168.0.15')
s7.cmd('ifconfig s7 192.168.0.17')
s4.cmd('ifconfig s4 192.168.0.14')
s12.cmd('ifconfig s12 192.168.0.112')
s6.cmd('ifconfig s6 192.168.0.16')
```

```

s2.cmd('ifconfig s2 192.168.0.12')
s11.cmd('ifconfig s11 192.168.0.111')
s1.cmd('ifconfig s1 192.168.0.11')
s9.cmd('ifconfig s9 192.168.0.19')
s14.cmd('ifconfig s14 192.168.0.114')
CLI(net)
net.stop()
myNetwork()

```

- activation des agents sFlow au niveau des commutateurs virtuels

Le script suivant est utilisé pour l'activation des agents sFlow au niveau des commutateurs virtuels : *i=1 while ((i<15))*

```

do
sudo ovs-vsctl --id=@sflow create sflow agent=s$i target="127.0.0.1 :6343" sampling=1 polling=1 -- set bridge s$i sflow=@sflow
((i+=1))
done
exit 0

```

- activation du collecteur sFlow

Dans notre cas d'étude le collecteur sFlow est représenté par l'outil *sflow - rt*. Le script suivant présente l'installation et l'activation de l'outil sflow :

```

wget http://www.inmon.com/products/sFlow-RT/sflow-rt.tar.gz
tar -xvzf sflow-rt.tar.gz
cd sflow-rt
./start.sh

```

- exécution de la simulation

L'exécution de la simulation passe par l'exécution des trois modules qui constituent notre *framework*. La figure III-3 illustre le digramme de classe du module de collecte et traitement des informations qui est composé des classes de traitement des flux, classes de traitement de topologie et la classe pour le mappage entre les flux et la topologie.



Figure-A III-3 Diagramme de classe du module de collecte et traitement

La figure III-4 illustre le module d'optimisation.

Les figures III-5 et III-6 représentent les scripts utilisés par le module de contrôle pour le routage des flux et le contrôle des débits des ports respectivement.


```

#!/usr/bin/python

import sys
import os

def policing_burst(node_id, port_dst, state):

    if state == 1:
        command="sudo ovs-vsctl set interface s"+node_id+"-eth"+port_dst+" ingress_policing_burst=1000"
    elif state == 2:
        command="sudo ovs-vsctl set interface s"+node_id+"-eth"+port_dst+" ingress_policing_burst=10000"
    elif state == 3:
        command="sudo ovs-vsctl set interface s"+node_id+"-eth"+port_dst+" ingress_policing_burst=100000"
    elif state == 4:
        command="sudo ovs-vsctl set interface s"+node_id+"-eth"+port_dst+" ingress_policing_burst=1000000"

    os.system(command)

def policing_rate(node_id, port_dst, state):

    if state == 1:
        command="sudo ovs-vsctl set interface s"+node_id+"-eth"+port_dst+" ingress_policing_rate=1000"
    elif state == 2:
        command="sudo ovs-vsctl set interface s"+node_id+"-eth"+port_dst+" ingress_policing_rate=10000"
    elif state == 3:
        command="sudo ovs-vsctl set interface s"+node_id+"-eth"+port_dst+" ingress_policing_rate=100000"
    elif state == 4:
        command="sudo ovs-vsctl set interface s"+node_id+"-eth"+port_dst+" ingress_policing_rate=1000000"

    os.system(command)

def setup():

    node_id = sys.argv[2]
    port_dst = sys.argv[3]
    state = int(sys.argv[4])

    policing_burst(node_id, port_dst, state)
    policing_rate(node_id, port_dst, state)

setup()

```

Figure-A III-6 Script utilisé pour le contrôle des débits des ports

BIBLIOGRAPHIE

- Ananthanarayanan, Ganesh et R. H. Katz. 2008. « Greening the Switch ».
- André, Barroso Luiz et Urs Hölzle. 2007. « The case for energy-proportional computing ». *The Journal of Supercomputing*.
- Arista. novembre 2008. « User Manual : Arista Networks ».
- Asfandyar, Qureshi, Rick Weber, Hari Balakrishnan, John Gutttag et Bruce Maggs. 2009. « Cutting the electric bill for internet-scale systems ». *ACM SIGCOMM computer communication review*, vol. 39, n° 4, p. 123-134.
- Athanasiou, G. 2013. « Energy-Efficient Traffic Engineering for Future Networking Infrastructures ». In *International Conference on Distributed Computing and Networking*, p. 1-12.
- Awad, Mohamad Khattar, Yousef Rafique et Rym A M'Hallah. 2017. « Energy-aware routing for software-defined networks with discrete link rates : A benders decomposition-based heuristic approach ». *Sustainable Computing : Informatics and Systems*, vol. 13, p. 31-41.
- Back,Thomas, David B. Fogel et Zbigniew Michalewicz. 2000. « Evolutionary computation 1 : basic algorithms and operators ». *CRC Press*, .
- Bianzino, Aruna Prem, Claude Chaudet, Dario Rossi, Jean-Louis Rougier et al. 2012. « A survey of green networking research ». *IEEE Communications Surveys & Tutorials*, vol. 14, n° 1, p. 3-20.
- Bilal, Kashif, Samee U Khan, Sajjad A Madani, Khizar Hayat, Majid I Khan, Nasro Min-Allah, Joanna Kolodziej, Lizhe Wang, Sherali Zeadally et Dan Chen. 2013. « A survey on green communications using adaptive link rate ». *Cluster Computing*, vol. 16, n° 3, p. 575-589.
- Boisson,J-C., Laetitia Jourdan E-G. Talbi et Christian Rolando. 2006. « Protein sequencing with an adaptive genetic algorithm from tandem mass spectrometry ». In *2006 IEEE International Conference on Evolutionary Computation*, p. 1412-1419.
- Bolla, Raffaele, Maurizio Giribaldi, Rafiullah Khan et Matteo Repetto. 2016. « Smart Proxying : Reducing Energy Waste in Network Devices During Idle Periods ».
- Brandon, Heller, Srinivasan Seetharaman, Priya Mahadevan, Yiannis Yiakoumis, Puneet Sharma, Sujata Banerjee et Nick McKeown. 2010. « ElasticTree : Saving Energy in Data Center Networks ». In *NSDI*, vol. 10, p. 249-264.
- Brocade. 2012. « Network Transformation with Software-Defined Networking and Ethernet Fabrics ». En Ligne. <www.brocade.com/downloads/documents/positioningpapers/network-transformation-sdn-wp.pdf>. Consulté le 17 février 2017.

- Brunilde, Sansò et Hakim Mellah. 2009. « On reliability, performance and Internet power consumption ». *IEEE Transactions on Computers*, , p. 259-264.
- Celenlioglu, M.R., S.B. Goger et H.A. Mantar. 2014. « An SDN-based energy-aware routing model for intra-domain networks ». In *Proceeding of 22nd International IEEE Conference on Telecommunications and Computer Networks (SoftCOM)*. p. 61-66.
- Chamara, Gunaratne, Ken Christensen et Bruce Nordman. 2005. « Managing energy consumption costs in desktop PCs and LAN switches with proxying, split TCP connections, and scaling of link speed ». *International Journal of Network Management*, , p. 297-310.
- Chamara, Gunaratne, Kenneth Christensen, Bruce Nordman et Stephen Suen. 2006. « Ethernet Adaptive Link Rate (ALR) : Analysis of a buffer threshold policy ». in *Proc. IEEE Global Communications Conference (GLOBECOM 2006)*, (San Francisco, California, USA).
- Chamara, Gunaratne, Kenneth Christensen, Bruce Nordman et Stephen Suen. 2008. « Reducing the energy consumption of Ethernet with adaptive link rate (ALR) ». *IEEE Transactions on Computers*, vol. 4, n° 57, p. 448-461.
- Chatterjee, Bijoy Chand, Nityananda Sarma, Partha Pratim Sahu et Eiji Oki. 2017. « End-to-End Traffic Grooming ». In *Routing and Wavelength Assignment for WDM-based Optical Networks*, p. 45–50. Springer.
- Chiaraviglio, Luca, Marco Mellia et Fabio Neri. 2009. *Energy-aware backbone networks : a case study*. 1-5 p.
- Chih-Lin, I, Corbett Rowell, Shuangfeng Han, Zhikun Xu, Gang Li et Zhengang Pan. 2014. « Toward green and soft : a 5G perspective ». *IEEE Communications Magazine*, vol. 52, n° 2, p. 66–73.
- Choi, In-Chan, Seong-In Kim et Hak-Soo Kim. 2003. « A genetic algorithm with a mixed region search for the asymmetric traveling salesman problem ». *Computers et Operations Research* 30, , p. 773-786.
- Christensen, J.Kenneth, Chamara Gunaratne, Bruce Nordman et Alan D. George. 2004. « The next frontier for communications networks : power management ». *Computer Communications*, vol. 27, n° 18, p. 1758-1770.
- Chu, P. C. et John E. Beasley. 1998. « Constraint handling in genetic algorithms : the set partitioning problem ». *Journal of Heuristics* 4, , p. 323-357.
- Costa, Da, Georges, Jean-Patrick Gelas, Yiannis Georgiou, Laurent Lefevre, Anne-Cécile Orgerie, Jean-Marc Pierson, Olivier Richard et Kamal Sharma. 2009. « The green-net framework : Energy efficiency in large scale distributed systems ». *IPDPS 2009. IEEE International Symposium on*, p. 1-8.

- Cs.stanford.edu. 2003. « Abilene Topology ». En Ligne. <<http://cs.stanford.edu/people/eroberts/courses/soco/projects/2003-04/internet-2/architecture.html>>. Consulté le 12 février 2015.
- Dabaghi, Fahimeh, Zeinab Movahedi et Rami Langar. 2017. « A survey on green routing protocols using sleep-scheduling in wired networks ». *Journal of Network and Computer Applications*, vol. 77, p. 106–122.
- Dabaghi-Zarandi, Fahimeh et Zeinab Movahedi. 2017. « An energy-efficient algorithm based on sleep-scheduling in IP backbone networks ». *International Journal of Communication Systems*.
- Deb, Kalyanmoy et Hans-Georg Beyer. 1999. « Self-adaptation in real-parameter genetic algorithms with simulated binary crossover ». *In Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 1*, , p. 172-179.
- Eaton, John W. 2004. *Matlab*, version 7.0.1. Logiciel. Lieu de publication : Nom de l'éditeur.
- Eyupoglu, Can et Muhammed Ali Aydin. 2015. « Energy Efficiency in Backbone Networks ». *Procedia-Social and Behavioral Sciences*, vol. 195, p. 1966–1970.
- Fouli, Kerim et Martin Maier. 2009. « The road to carrier-grade Ethernet ». *IEEE Communications Magazine*, vol. 47, n° 3, p. 30-38.
- Frank, David J, Robert H Dennard, Edward Nowak, Paul M Solomon, Yuan Taur et Hon-Sum Philip Wong. 2001. « Device scaling limits of Si MOSFETs and their application dependencies ». *Proceedings of the IEEE*, vol. 89, n° 3, p. 259–288.
- Frédéric, Giroire, Joanna Moulierac et Truong Khoa Phan. 2014. « Optimizing rule placement in software-defined networks for energy-aware routing ». *In 2014 IEEE Global Communications Conference*, p. 2523-2529.
- Gambardella, Luca Maria et Marco Dorigo. 2000. « An ant colony system hybridized with a new local search for the sequential ordering problem ». *INFORMS Journal on Computing* 12, , p. 237-255.
- Gartner, G. 2007. « Says data centers account for 23 per cent of global ICT CO2 emissions ». En Ligne. <<http://www.gartner.com/it/page.jsp?id=530912>>. Consulté le 11 Novembre 2015.
- Gelenbe, Erol et Yves Caseau. 2015. « The impact of information technology on energy consumption and carbon emissions ». *Ubiquity*, vol. 2015, n° June, p. 1.
- Giroire, Frédéric, Joanna Moulierac, Truong Khoa Phan et Frédéric Roudaut. 2015. « Minimization of network power consumption with redundancy elimination ». *Computer communications*, vol. 59, p. 98–105.
- Goldberg, David, Kalyanmoy Deb et Bradley Korb. 1989. « Messy genetic algorithms : Motivation, analysis, and first results ». *Complex systems*, , p. 493-530.

- Group, Network Working. Decembre 2006. « NETCONF Configuration Protocol ». *RFC 4741 (Proposed Standard)*.
- Helmut, Hlavacs, Georges Da Costa et Jean-Marc Pierson. 2007. « Using Low-Power Modes for Energy Conservation in Ethernet LANs ». *In INFOCOM*, vol. 7, p. 2451-2455.
- Helmut, Hlavacs, Georges Da Costa et Jean-Marc Pierson. 2009. « Energy consumption of residential and professional switches ». *In Computational Science and Engineering*, vol. 1, n° 9, p. 240-246.
- hiaraviglio Luca, Marco Mellia et Fabio Neri. 2012. « Minimizing ISP network energy cost : formulation and solutions ». *IEEE/ACM Transactions on Networking (TON)*, vol. 20, n° 2, p. 463-476.
- Himanshu, Anand, Casey Reardon, Rajagopal Subramaniyan et Alan D. George. 2006. « Ethernet Adaptive Link Rate (ALR) : Analysis of a MAC Handshake Protocol ». *In LCN*, p. 533-534.
- Hiroki, Shirayanagi, Hiroshi Yamada et Kono Kenji. 2013. « Honeyguide : A vm migration-aware network topology for saving energy consumption in data center networks ». *IEICE TRANSACTIONS on Information and Systems*, vol. 96, n° 9, p. 2055-2064.
- Holland, J.H. 1975. « Adaptation in natural and artificial systems. An introductory analysis with application to biology, control, and artificial intelligence ». *Ann Arbor, MI : University of Michigan Press*, p. 439-444.
- HP. 2012a. « Deliver HP Virtual Application Networks ». En Ligne. <<http://h17007.www1.hp.com/docs/interopny/4AA4-3872ENW.pdf>>. Consulté le 10 Juin 2015.
- HP. 2012b. « Realizing the Power of SDN with HP Virtual Application Networks ». En Ligne. <<http://h17007.www1.hp.com/docs/interopny/4AA4-3871ENW.pdf>>. Consulté le 10 Juin 2015.
- Hussain, M. F. et K. S. Al-Sultan. 1997. « A hybrid genetic algorithm for nonconvex function minimization ». *Journal of Global Optimization 11*, , p. 313-324.
- Hyojoon, Kim et Nick Feamster. 2013. « Improving network management with software defined networking ». *IEEE Communications Magazine*, vol. 51, n° 2, p. 114-119.
- IBM. 2009. « V12. 1 : Users manual for CPLEX ». *Journal of Green Engineering*.
- IEEE. 2010. « IEEE P802.3az Energy Efficient Ethernet Task Force ». En Ligne. <<http://www.ieee802.org/3/az/index.html>>. Consulté le 12 février 2013.
- Jalali, Amin, Farzin Piltan, Maziyar Keshtgar et Meysam Jalali. 2013. « Colonial Competitive Optimization Sliding Mode Controller with Application to Robot Manipulator ». *International Journal of Intelligent Systems and Applications 5*, , p. 50.

- Jim, Guichard, François Le Faucheur et Jean-Philippe Vasseur. 2005. « Definitive MPLS network designs ». In *Definitive MPLS network design*.
- Joseph, Chabarek, Joel Sommers, Paul Barford, Cristian Estan, David Tsiang, et Steve Wright. 2008a. « Power Awareness in Network Design and Routing ». in *Proc. 27th IEEE Annual Conference on Computer Communications. (INFOCOM 2008)*, vol. 4, n° 27.
- Joseph, Chabarek, Joel Sommers, Paul Barford, Cristian Estan, David Tsiang et Steve Wright. 2008b. « Power awareness in network design and routing ». In *INFOCOM, IEEE Conference on Computer Communications*, vol. 27.
- Kannan, Kalapriya et Banerjee Subhasis. 2013. « Compact TCAM : Flow entry compaction in TCAM for power aware SDN ». In *International Conference on Distributed Computing and Networking*, p. 439-444.
- Khan, Shafiullah et Jaime Lloret Mauri. 2013. « Green Networking and communications : ICT for sustainability ». In *CRC Press*.
- Konstantinos, Samdanis, Peter Rost, Andreas Maeder, Michela Meo et Christos Verikoukis. 2015. « Green Communications : Principles, Concepts and Practice ». *John Wiley & Sons, NY*.
- Lin, Wang, Fa Zhang, Chenying Hou, Jordi Arjona Aroca et Zhiyong Liu. 2013. « Incorporating rate adaptation into green networking for future data centers ». In *Network Computing and Applications (NCA)*, , p. 106-109.
- Luca, Chiaraviglio, Marco Mellia et Fabio Neri. 2009. « Reducing power consumption in backbone networks ». In *2009 IEEE International Conference on Communications*, p. 1-6.
- Luo, Shouxi, Hongfang Yu et al. 2014. « Fast incremental flow table aggregation in SDN ». In *Computer Communication and Networks (ICCCN), 2014 23rd International Conference on*. p. 1–8. IEEE.
- Mario, Baldi et Yoram Ofek. 2009. « Time for a "greener" internet ». In *2009 IEEE International Conference on Communications Workshops*, p. 1-6.
- Mark, Allman, Ken Christensen, Bruce Nordman et Vern Paxson. 2007. « Enabling an Energy-Efficient Future Internet Through Selectively Connected End Systems ». In *HotNets*.
- Mark, Reitblatt, Nate Foster, Jennifer Rexford et David Walker. 2011. « Consistent updates for software-defined networks : Change you can believe in ! ». In *in Proceedings of the 10th ACM Workshop on Hot Topics in Networks*. p. 7-18.
- Mark, Weiser, Brent Welch, Alan Demers et Scott Shenker. 1994. « Scheduling for reduced CPU energy ». In *Mobile Computing*, p. 449-471.

- Maruti, Gupta et Suresh Singh. 2003. « Greening of the Internet ». *In Proc. ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM 2003)*, p. 19-26.
- Maruti, Gupta, Satyajit Grover et Suresh Singh. 2004. « A feasibility study for power management in LAN switches ». *In Proc. 12th IEEE International Conference on Network Protocols (ICNP 2004)*, , p. 361-371.
- MATLAB. 2010. *version 7.10.0 (R2010a)*. Natick, Massachusetts : The MathWorks Inc.
- McKeown,Nick, Tom Anderson Hari Balakrishnan-Guru Parulkar Larry Peterson Jennifer Rexford Scott Shenker et Jonathan Turner. 2008. « OpenFlow : enabling innovation in campus networks ». *IACM SIGCOMM Computer Communication Review*, , p. 69-74.
- Meiners, Chad R, Alex X. Liu et Eric Torng. 2007. « TCAM Razor : A systematic approach towards minimizing packet classifiers in TCAMs ». *IEEE International Conference on Network Protocols*, p. 266-275.
- Meiners, Chad R., Alex X. Liu, et Eric Torng. 2012. « Bit weaving : A non-prefix approach to compressing packet classifiers in TCAMs ». *IEEE/ACM Transactions on Networking (ToN)*, vol. 20, n° 2, p. 488-500.
- Mijumbi, Rashid, Joan Serrat, Juan-Luis Gorricho, Niels Bouten, Filip De Turck et Raouf Boutaba. 2016. « Network function virtualization : State-of-the-art and research challenges ». *IEEE Communications Surveys & Tutorials*, vol. 18, n° 1, p. 236–262.
- Mingui, Zhang, Cheng Yi, Bin Liu et Beichuan Zhang. 2010. « GreenTE : Power-aware traffic engineering ». *Network Protocols (ICNP), IEEE International Conference on*, , p. 21-30.
- Mininet, Team. 2014. « Mininet ».
- Nanxi, Kang, Zhenming Liu, Jennifer Rexford et David Walker. 2013. « Optimizing the one big switch abstraction in software-defined networks ». *In Proceedings of the ninth ACM conference on Emerging networking experiments and technologies*, vol. 9, p. 13-24.
- Nedeljko, Vasić, Prateek Bhurat, Dejan Novaković, Marco Canini, Satyam Shekhar et Dejan Kostić. 2011. « Identifying and using energy-critical paths ». *In Proceedings of the Seventh Conference on emerging Networking Experiments and Technologies (ACM)*, , p. 18.
- NFV. 2016. « What is NFV ? ». En Ligne. <<https://www.sdxcentral.com/nfv/definitions/whats-network-functions-virtualization-nfv>>. Consulté le 15 Mars 2016.
- nsf.gov. 2008. « NSF Topology ». En Ligne. <https://www.dit.upm.es/vnumlwiki/index.php/Example-NSF-14_1.8>. Consulté le 23 février 2016.
- Ochoa-Aday, Leonardo, Cristina Cervello-Pastor et Adriana Fernandez-Fernandez. 2015. « Current Trends of Topology Discovery in OpenFlow-based Software Defined Networks ».

- Olson, Eric G. 2008. « Creating an enterprise-level “green” strategy ». *Journal of business strategy*, vol. 29, n° 2, p. 22–30.
- ONF. June 2014. « SDN architecture ». En Ligne. <www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR_SDN_ARCH_1.0_06062014.pdf>. Consulté le 12 Janvier 2016.
- opendaylight.org (2017). 2017. « OpenDaylight : Open Source SDN Platform ». En Ligne. <<https://www.opendaylight.org>>. Consulté le 20 Mars 2017.
- Pfaff, Ben, Justin Pettit, Teemu Koponen, Ethan J Jackson, Andy Zhou, Jarno Rajahalme, Jesse Gross, Alex Wang, Joe Stringer, Pravin Shelar et al. 2015. « The Design and Implementation of Open vSwitch. ». In *NSDI*. p. 117–130.
- Pradeep, Purushothaman, Mukund Navada, Rajagopal Subramaniyan, Casey Rendon et Alan D. George. 2006. « Power-proxying on the NIC : a case study with the Gnutella file-sharing protocol ». In *Proceedings.IEEE Conference on Local Computer Networks*, , p. 519-520.
- Prem, Bianzino Aruna, Claude Chaudet, Federico Larroca, Dario Rossi, et Jean-Louis Rougier. 2010. « Energy-aware routing : a reality check ». In *2010 IEEE Globecom Workshops*, p. 1422-1427.
- Priya, Mahadevan, Puneet Sharma, Sujata Banerjee et Parthasarathy Ranganathan. 2009. « A power benchmarking framework for network devices ». In *International Conference on Research in Networking*, p. 795-808.
- Priya, Mahadevan, Sujata Banerjee et Puneet Sharma. 2010. « Energy proportionality of an enterprise network ». In *Proceedings of the first ACM SIGCOMM workshop on Green networking*, p. 53-60.
- Raffaele, Bolla, Roberto Bruschi, Franco Davoli et Flavio Cucchietti. 2011. « Energy efficiency in the future internet : a survey of existing approaches and trends in energy-aware fixed network infrastructures ». *IEEE Communications Surveys*, vol. 2, n° 2, p. 223-244.
- Reviriego, Pedro, HERNADEZ Jose-Alberto et LARRABEITI David. 2010. « Burst transmission for energy-efficient ethernet ». *IEEE Internet Computing*,, vol. 14, n° 4, p. 50-57.
- Richard, Brown. 2008. « Report to congress on server and data center energy efficiency : Public law 109-431 ». *Lawrence Berkeley National Laboratory*.
- Sandhu, Sukhdeep Singh, Arushi Rawal, Prabhjot Kaur et Niyati Gupta. 2012. « Major components associated with green networking in information communication technology systems ». In *Computing, Communication and Applications (ICCCA), 2012 International Conference on*. p. 1–6. IEEE.
- SDN. 2013a. « Software Defined Networking ». En Ligne. <<https://www.opennetworking.org/sdn-resources/sdn-definition>>. Consulté le 10 Mars 2016.

- SDN. 2013b. « Software-Defined Networking (SDN) Definition ». En Ligne. <<https://www.opennetworking.org/sdn-resources/sdn-definition>>. Consulté le 10 Mars 2016.
- Sergiu, Nedevschi, Lucian Popa, Gianluca Iannaccone, Sylvia Ratnasamy et David Wetherall. 2008. « Reducing Network Energy Consumption via Sleeping and Rate-Adaptation ». *In NSDI*, vol. 8, n° 2, p. 323-336.
- Sergiu, Nedevschi, Jaideep Chandrashekar, Junda Liu, Bruce Nordman, Sylvia Ratnasamy et Nina Taft. 2009. « Skilled in the Art of Being Idle : Reducing Energy Waste in Networked Systems ». *In NSDI*, vol. 9, p. 381-394.
- sflow.org. 2003. « Traffic Monitoring using sFlow ». En Ligne. <www.sflow.org>. Consulté le 19 Mars 2016.
- Shao-Heng, Wang, Patrick P-W. Huang, Charles H-P. Wen et Li-Chun Wang. 2014. « EQVMP : Energy-efficient with QoS-aware VM Placement algorithm ». *In The International Conference on Information Networking(ICOIN)*, p. 220-225.
- Shayan, E. et A. Chittilappilly. 2004. « Genetic algorithm for facilities layout problems based on slicing tree structure ». *International Journal of Production Research* 42, , p. 4055-4067.
- Sherali, Seadally, Samee Ullah Khan et Naveen Chilamkurti. 2012. « Energy-efficient networking : past, present, and future ». *The Journal of Supercomputing*, vol. 62, n° 3, p. 1093–1118.
- Shimon, Even, Alon Itai et Adi Shamir. 1975. « On the complexity of time table and multi-commodity flow problems ». *In Foundations of Computer Science, 16th Annual Symposium on*, , p. 184-193.
- Szpiro, George G. 1997. « A search for hidden relationships : data mining with genetic algorithms ». *Computational Economics* 10, , p. 267-277.
- Tang Jian, Brendan Mumey, Yun Xing-Andy Johnson. 2012. « On exploiting flow allocation with rate adaptation for green networking ». *In INFOCOM*, p. 1683-1691.
- Uhlig, Steve, Bruno Quoitin, Jean Lepropre et Simon Balon. 2006. « Providing Public Intra-domain Traffic Matrices to the Research Community ». *SIGCOMM Comput. Commun. Rev.*, vol. 36, n° 1, p. 83–86.
- Vakilinia, Shahin, Behdad Heidarpour et Mohamed Cheriet. 2016. « Energy Efficient Resource Allocation in Cloud Computing Environments ». *IEEE Access*.
- Van Heddeghem, Ward, Bart Lannoo, Didier Colle, Mario Pickavet et Piet Demeester. 2016. « A quantitative survey of the power saving potential in IP-over-WDM backbone networks ». *IEEE Communications Surveys & Tutorials*, vol. 18, n° 1, p. 706–731.

- Vela, Alba P, Anna Vía, Fernando Morales, Marc Ruiz et Luis Velasco. 2016. « Traffic generation for telecom cloud-based simulation ». In *Transparent Optical Networks (ICTON), 2016 18th International Conference on*. p. 1–4. IEEE.
- Vetter, P, Tolga Ayhan, K Kanonakis, Bart Lannoo, KL Lee, Laurent Lefèvre, C Monney, F Saliou et Xin Yin. 2013. « Towards energy efficient wireline networks, an update from greentouch ». In *OptoElectronics and Communications Conference held jointly with 2013 International Conference on Photonics in Switching (OECC/PS), 2013 18th*. p. 1–2. IEEE.
- Will, Fisher, Martin Suchara et Jennifer Rexford. 2010. « reining backbone networks : reducing energy consumption by shutting off cables in bundled links ». In *Proceedings of the first ACM SIGCOMM workshop on Green networking*, p. 29-34.
- Wu, Jingjing, Zhaolong Ning et Lei Guo. 2017. « Energy-Efficient Survivable Grooming in Software-Defined Elastic Optical Networks ». *IEEE Access*.
- Xiaodong, Wang, Yanjun Yao, Xiaorui Wang, Kefa Lu et Qing Cao. 2012. « Carpo : Correlation-aware power optimization in data center networks ». In *INFOCOM*, p. 1125-1133.
- Yen, Jin Y. 1971. « Finding the k shortest loopless paths in a network ». *management Science*, vol. 17, n° 11, p. 712–716.
- Yetginer, Emre et George N Rouskas. 2009. « Power efficient traffic grooming in optical WDM networks ». In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*. p. 1–6. IEEE.
- Yossi, CKanizo, David Hay, et Isaac Keslassy. 2013. « Palette : Distributing tables in software-defined networks ». In *INFOCOM, 2013 Proceedings IEEE*, p. 545-549.
- Zemmouri, Samy, Shahin Vakilinia et Mohamed Cheriet. 2016. « Let's adapt to network change : Towards energy saving with rate adaptation in SDN ». In *Network and Service Management (CNSM), 2016 12th International Conference on*. p. 272–276. IEEE.
- Zhao, Yimeng, Samantha Lo, Ellen Zegura, Niky Riga et Mostafa Ammar. 2017. « Virtual Network Migration on the GENI Wide-Area SDN-Enabled Infrastructure ». *arXiv preprint arXiv :1701.01702*.