

Republique De Cote D'ivoire
Union - Discipline - Travail

**MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE**



**Institut National Polytechnique
Félix HOUPHOUËT-BOIGNY de Yamoussoukro**

EDP
Ecole Doctorale Polytechnique
Ecole Doctorale Polytechnique

UMRI : 78

THESE

Pour l'obtention du grade de

DOCTEUR DE L'INSTITUT NATIONAL POLYTECHNIQUE FELIX HOUPHOUËT-BOIGNY

Mention : **Informatique et Télécommunications**

Spécialité : **Réseaux Informatiques**

Thème :

Contribution à l'optimisation de la consommation de l'énergie dans les réseaux SDN

Présentée et soutenue publiquement le ----- par

M. KRA LAGASANE OUATTARA

Devant le Jury composé de :

M. ZOUEU T. Jeremie	Professeur Titulaire, Institut National Polytechnique Houphouët-Boigny (Yamoussoukro)	Président
M. ASSEU Kouamé Olivier Pascal	Professeur Titulaire, Institut National Polytechnique Houphouët-Boigny (Yamoussoukro) Dept G.E.E	Directeur de Thèse
M. DIABATE Nabongo	Maître de Conférences, Université Alassane Ouattara (Bouaké)	Rapporteur
M. Roger Marcelin FAYE	Professeur Titulaire, Université Amadou Mathar MBOW (Dakar)	Rapporteur
M. AKA Boko	Professeur Titulaire, Université Nangui Abrogoua (Abidjan)	Examineur

DEDICACE

A ma famille !

REMERCIEMENTS

Au moment de soumettre la présente thèse à évaluation, qu'il nous soit permis de nous acquitter d'un devoir de reconnaissance aux personnes qui ont contribué à la réalisation de ce manuscrit.

Ainsi, nous remercions, tout d'abord, notre directeur de Thèse, Monsieur ASSEU Kouamé Olivier Pascal, Professeur Titulaire pour avoir dirigé ce travail.

Nous remercions aussi Monsieur GOORE Bi Tra, Maître de Conférences, pour avoir accepté de nous offrir son encadrement.

Nos remerciements vont également à l'endroit des responsables du LARIT (Laboratoire de Recherche en Informatique et Télécommunication) de Yamoussoukro, l'ensemble du personnel administratif et scientifique ainsi que doctorants, post doctorants avec qui nous avons eu le grand plaisir de partager des connaissances.

Nous manifestons notre reconnaissance à Monsieur KOFFI N'Guessan, Directeur Général de l'Institut National Polytechnique Félix Houphouët Boigny (INP-HB) de Yamoussoukro, Côte d'Ivoire et son équipe pour leur accueil cordial dans l'illustre école de formation qu'ils gèrent.

Cette reconnaissance va également à l'endroit de Monsieur YAO Kouassi Benjamin, Professeur Titulaire, Directeur de l'Ecole Doctorale Polytechnique (EDP) de Yamoussoukro, Côte d'Ivoire et son équipe pour leur accueil chaleureux dans la prestigieuse école doctorale dont ils assurent la direction.

Nos sincères remerciements à Monsieur ADAMA Konaté, Maître de Conférences, Directeur de l'ESATIC d'Abidjan, Côte d'Ivoire et son équipe pour ses conseils.

Nous ne pouvons oublier Monsieur ZOUEU Thouakessh Jérémie, Directeur du Laboratoire des Unités Mixtes de Recherche et d'Innovation (UMRI 78).

Nous adressons nos chaleureux remerciements à Monsieur MONSAN Vincent, Maître de Conférences pour sa contribution à cette Thèse.

Nous tenons à exprimer toute notre gratitude à Monsieur ANASSE Adja A. Augustin, Professeur Titulaire, Doyen de l'UFR des Sciences Economiques et Développement de l'Université Alassane Ouattara de Bouaké, pour ses précieux conseils.

Nous remercions très chaleureusement Monsieur FAYE Roger, Professeur Titulaire et Monsieur DIABATE Nabongo, Maître de Conférences d'avoir accepté de rapporter notre Thèse.

Nous disons également merci à Monsieur AKA Boko, Professeur Titulaire d'avoir accepté d'examiner notre Thèse.

Nous disons merci à Dr TANO H Tanoh Lambert pour son soutien.

Nous remercions Monsieur KONAN Lambert, Professeur Titulaire, pour sa contribution à la correction des fautes d'orthographe, de grammaire et de syntaxe.

Nous remercions les collègues du service informatique de l'UFR-SED de l'Université Alassane Ouattara de Bouaké, Gballou Yao Théophile, Dr N'Guessan Koffi, Samassi Adama, Manzan Kabran Didier et Bamba Djiba pour leur sens de collaboration.

Et enfin, à tous nos proches pour leur soutien tout au long des trois années d'intenses recherches et tous ceux qui nous ont aidé à la réalisation de cette Thèse.

A tous les membres du jury, nous adressons notre reconnaissance pour l'honneur qu'ils nous font d'être présents pour l'évaluation de ce travail. Ce sont :

- **M. ZOUEU T. Jérémie** (Professeur Titulaire), Président
- **M. ASSEU Kouamé Olivier Pascal** (Professeur Titulaire), Directeur
- **M. DIABATE Nabongo** (Maître de Conférences), Rapporteur

- **M. Roger Marcelin FAYE** (Professeur Titulaire), Rapporteur
- **M. AKA Boko** (Professeur Titulaire), Examineur

RESUME

Avec l'avènement des nouvelles technologies tels que l'IoT, le Big Data, l'augmentation des architectures des réseaux de Télécommunications est en pleine expansion. Leur satisfaction en énergie est devenue un défi important à relever tant sur le plan économique qu'écologique.

Plusieurs approches ont, cependant, été proposées pour sa réduction. Nous utilisons dans cette Thèse, un nouveau paradigme de réseautage appelé Software Defined Networking (SDN). C'est une architecture dans laquelle le plan de données, responsable de transfert de paquets est découplé du plan de contrôle, responsable de la prise de décision.

C'est donc un contrôleur centralisé qui va gérer tout le réseau. Le SDN permet une grande flexibilité dans la gestion du réseau en offrant la programmabilité des équipements réseaux et permet de rediriger sur d'autres ports tout en éteignant ceux qui ne sont pas utilisés.

Dans cette Thèse, nous proposons une solution basée sur un nouveau modèle de minimisation de la consommation d'énergie sous contrainte de délai, perte de paquets et gigue.

C'est un modèle mathématique et une nouvelle stratégie d'économie d'énergie. Il consiste à minimiser la consommation du routeur "i" à l'instant "t" dont $\ell(t)$ ports ont été désactivés. Une fonction linéaire et strictement croissante est obtenue par la nouvelle stratégie d'économie d'énergie (mode désactivation progressif des ports) à chaque instant déterminant ainsi le "**gain d'énergie**". Cette contribution a fait l'objet de la première et deuxième publication.

Notre proposition consistera à introduire un paramètre qui caractérise le nombre de ports désactivés à l'instant t ($\ell(t)$) dans la fonction objectif de Gelenbe et al., 2009 sous contrainte de délai, perte de paquets et gigue.

Le maximum de ports éteints constitue notre "gain d'énergie". Nous utilisons la technologie SDN pour éteindre les ports ou liens. Les ports éteints seront ceux des routeurs adjacents. Notre stratégie pour faire une économie d'énergie reposera sur le mode de désactivation des ports.

Notre solution algorithmique s'appuyant sur le modèle mathématique proposé, va d'abord déterminer les plus courts chemins par l'algorithme de Kruskal en désactivant les liens qui créent les cycles, bien avant l'utilisation du réseau formé.

Le contrôleur SDN est chargé d'allumer les routeurs dans le réseau. Une fois que les routeurs sont allumés, un seul port est activé et les autres sont éteints.

Notre troisième contribution concerne la simulation du modèle mathématique sous contrainte de délai, perte de paquets et gigue. Nous avons déterminé le délai de transmission des paquets en fonction du nombre de ports désactivés. Les résultats graphiques obtenus, ont montré une bonne réduction de délai avec un taux de perte de paquets et gigue appréciables. Cette contribution a été acceptée et publiée. Le principe de résolution est celui des auteurs Radu et al., 2015 à la seule différence que nous avons travaillé dans les ports.

Les résultats de la simulation montrent que notre modèle donne de bonne valeur d'économie d'énergie 87,5% d'économie d'énergie dans les ports et 33,33% d'économie d'énergie dans les liens avec une bonne qualité de service.

Mots clés: Internet des objets, Big Data, Réseau défini par application, désactivation des ports, modèle, Optimisation, algorithme de Kruskal

ABSTRACT

With the advent of new technologies such as IoT, Big Data, the increase in telecommunications network architectures is booming. Their energy satisfaction has become an important challenge to take up both economically and ecologically.

Several approaches have, however, been proposed for its reduction. In this Thesis, we use a new networking paradigm called Software Defined Networking (SDN). It is an architecture in which the data plane, responsible for packet transfer is decoupled from the control plane, responsible for decision-making.

It is therefore a centralized controller that will manage the entire network. SDN allows great flexibility in network management by providing the programmability of network equipment and allows to redirect to other ports while switching off those that are not in use.

In this Thesis, we propose a solution based on a new model for minimizing energy consumption under time constraints, packet loss and jitter. It is a mathematical model and a new energy saving strategy. It consists in minimizing the consumption of the router "i" at the instant "t" whose $\ell(t)$ ports have been disabled. A linear and strictly increasing function is obtained by the new energy saving strategy (progressive port deactivation mode) at each instant thus determining the "energy saving". This contribution was the subject of the first and second publication.

Our proposal will consist in introducing a parameter which characterizes the number of ports deactivated at time t ($\ell(t)$) in the objective function of Gelenbe and al., 2009 under time constraint, packet loss and jitter.

The maximum number of ports off constitutes our "energy saving". We use SDN technology to shut down ports or links. The ports that are off will be those of adjacent routers. Our strategy to save energy will be based on how to deactivate the ports.

Our algorithmic solution based on the proposed mathematical model, will first determine the shortest paths by the Kruskal algorithm by deactivating the links that create the cycle, well before the use of the formed network. The SDN controller is responsible for turning on the routers in the network. Once the routers are turned on, only one port is enabled and the rest are turned off.

Our third contribution concerns the simulation of the mathematical model under delay constraint, packet loss and jitter. We have determined the packet transmission delay based on the number of disabled ports. The graphical results obtained, showed a good reduction of delay with an appreciable rate of packet loss and jitter. This contribution has been accepted and published. The principle of resolution is that of the authors Radu and al., 2015 with the only difference that we worked in ports.

The results of the simulation show that our model gives a good energy saving value 87.5% energy saving in ports and 33.33% energy saving in links with good quality on duty.

Keywords: Internet of Things, Big Data, SDN, port deactivation model, Optimization, Kruskal algorithm.

TABLE DES MATIERES

	Pages
DEDICACE	II
REMERCIEMENTS	III
RESUME.....	VI
ABSTRACT.....	VIII
TABLE DES MATIERES	X
LISTE DES FIGURES.....	XV
LISTE DES TABLEAUX.....	XVII
LISTE DES ABREVIATIONS ET SYMBOLES.....	XVIII
ABREVIATIONS.....	XVIII
SYMBOLES.....	XXII
INTRODUCTION GENERALE.....	1
CHAPITRE 1 : GENERALITES SUR LES METHODES D'OPTIMISATION DANS LES RESEAUX DE COMMUNICATION ET DANS LES RESEAUX SDN.....	7
1.1. Introduction	8
1.2 Architecture des réseaux de capteurs sans fil et réseaux maillés.....	9
1.2.1. Les réseaux de capteurs.....	9
1.2.2 Les réseaux maillés sans fil.....	11
1.2.3. Les limites.....	12
1.3 Concept et architecture de base des réseaux classiques.....	13
1.3.1 Fonctionnement de l'architecture classique	13
1.3.2 Limites des réseaux classiques.....	14
1.3 Concept des SDN.....	14
1.4.1 Définitions des termes du sujet.....	14

1.4.1.1 SDN.....	14
1.4.1.2 La congestion.....	15
1.4.1.3 La Consommation d'énergie.....	15
1.4.1.4 Optimisation de la consommation de l'énergie	15
1.4.2. Rôle des différentes couches du SDN.....	16
1.4.2.1 Couche application.....	16
1.4.2.2 Couche contrôleur	16
1.4.2.2 Couche infrastructure	17
1.5. Techniques d'optimisation utilisées dans les réseaux.....	19
1.5.1. Les réseaux de capteurs.....	19
1.5.1.1 durée de vie du réseau de capteur sans fil.....	19
1.5.1.2 Duty-cycling.....	20
1.5.2. Les réseaux maillés.....	21
1.6. Evolution vers les réseaux mobiles 5G.....	22
1.6.1. Les réseaux 2G.....	22
1.6.2. Les réseaux 3G.....	23
1.6.3. Les réseaux LTE/4G.....	24
1.7. Les réseaux mobiles 5G.....	25
1.8. Les réseaux SDN.....	27
1.9. Historique des réseaux SDN.....	28
1.10. Importance du SDN pour l'optimisation de la consommation.....	
de l'énergie dans les autres types de réseaux.....	30
1.11. CONCLUSION.....	31
CHAPITRE 2 : ETAT DE L'ART SUR L'OPTIMISATION DE LA	
CONSOMMATION DE L'ENERGIE DANS LES RESEAUX SDN.....	33
2.1. Introduction.....	34
2.2. Approches utilisées dans l'optimisation de la consommation	
d'énergie dans les réseaux de communication	35

2.2.1. Approche matérielle.....	35
2.2.2. Approche logicielle.....	37
2.2.2.1. Virtualisation des ressources dans le réseau.....	38
2.2.2.2. Honeyguide.....	38
2.2.3. Approche hybride.....	39
2.2.3.1. Explication de la Démarche expérimentale	41
2.2.3.2. Adaptation dynamique à la charge de trafic.....	45
2.2.3.3. Optimisation du placement des règles de flux.....	46
2.2.3.4. Compression des données de la table de routage SDN	46
2.2.3.5. Autres types de compression de données sur TCAM.....	48
2.3. Synthèse bibliographique pour les réseaux classiques et nouvelles générations (SDN).....	49
2.4. Conclusion	51
CHAPITRE 3 : CONTRIBUTION A L'OPTIMISATION DE LA CONSOMMATION DE L'ENERGIE DANS LES RESEAUX SDN.....	53
3.1. Introduction.....	54
3.2. Recherche de modèle d'économie d'énergie dans les réseaux SDN	57
3.2.1. Hypothèses de travail.....	58
3.2.2. Modélisation du problème.....	58
3.3. Recherche d'approche de résolution du modèle.....	62
3.3.1. La programmation linéaire en nombres entiers (PLNE).....	62
3.3.2. Les heuristiques algorithmiques.....	63
3.3.3. Le graphe de flot.....	63
3.4. Critère et Choix de la méthode de résolution.....	64
3.5. Représentation du modèle via OMNET ++.....	66
3.6. Description de l'algorithme d'économie d'énergie	71
3.6.1. Evaluation numérique du modèle.....	74
3.6.2. Performance de l'approche	74

3.7. Conclusion.....	75
CHAPITRE 4 : APPROCHE ALGORITHMIQUE DE LA	102
MINIMISATION DE LA CONSOMMATION UTILISANT LA	
FONCTION EXPONENTIELLE COMME CHOIX D'ECONOMIE	
D'ENERGIE DANS LES RESEAUX SDN.....	76
4.1. Introduction.....	77
4.2. Fonction exponentielle comme modèle mathématique pour réduire la	
consommation de l'énergie.....	78
4.2.1. Nouvelle formulation du problème	79
4.2.2. Données du problème.....	82
4.3. Approche algorithmique	82
4.3.1. Principe de l'approche	83
4.3.2. Méthode d'utilisation de l'approche	84
4.4. Stratégie d'économie d'énergie par extinction des ports	84
4.5. Evaluation et Performance de l'approche	85
4.6. Comparaison des résultats.....	85
4.7. Conclusion.....	86
CHAPITRE 5 : SIMULATION DE DELAI EN FONCTION DES PORTS	
DESACTIVES DANS LES RESEAUX SDN.....	88
5.1. Introduction.....	89
5.2. Synthèse sur quelques simulations	90
5.3. Présentation du modèle.....	95
5.3.1. Algorithme et Simulation de délai en fonction de ports	
désactivés.....	97
5.3.2. Evaluation de la performance.....	104
5.4. Conclusion	105
CONCLUSION GENERALE ET PERSPECTIVES.....	106
REFERENCES BIBLIOGRAPHIQUES.....	111

ANNEXES.....	121
ANNEXE A : Le simulateur OMNET++	122
ANNEXE B : Installation de OMNET++ 5.2.1.....	124
ANNEXE C : Code et différents algorithmes.....	127
ANNEXE D : Liste des publications tirées de la thèse.....	136

LISTE DES FIGURES

Figure 1.1	Architecture de réseaux de capteurs sans fil [16]	10
Figure 1.2	Architecture d'un réseau de capteur sans fil [17]	11
Figure 1.3	Architecture d'un réseau maillé sans fil [18]	12
Figure 1.4	Architecture comparée des réseaux SDN et traditionnels de [19]	13
Figure 1.5	Processus de communication entre les commutateurs et le contrôleur et un switch supportant le protocole Open flow18
Figure 1.6	Diagramme de transition des différents états d'activités d'un nœud RCSF	20
Figure 1.7	Modèle de consommation d'énergie [36, 37].....	22
Figure 1.8	Réseau d'accès [40]	24
Figure 1.9	Réseau de cœur [40]	25
Figure 1.10	Architecture C-RAN [41]	26
Figure 1.11	.Architecture du SDN avec Open flow	27
Figure 1.12	Environnement virtualisé	31
Figure 2.1	Etat sommeil et réveil d'un équipement	39
Figure 2.2	Topologie du réseau [1].....	41
Figure 2.3	Délai – première expérience [1].....	42
Figure 2.4	Perte de paquets – première.....	43
Figure 2.5	Gigue – première expérience [1]	43
Figure 3.1	Consommation énergétique des réseaux de communication [89]	55
Figure 3.2	Architecture du SDN [89]	56
Figure 3.3	Exemple d'algorithme de résolution ILP [59]	64
Figure 3.4	Architecture du réseau sur OMNET++	68
Figure 3.5	Architecture du réseau obtenu par l'algorithme Kruskal sur OMNET++	70
Figure 3.6	Le protocole SPRING [2]	72
Figure 4.1	Jour typique d'utilisation du réseau [92]	77
Figure 4.2	Réseau avec capacités [94]	83

Figure 4.3	Architecture obtenue par l'algorithme de Kruskal [94].....	83
Figure 5.1	Delai- Première experience [1]	91
Figure 5.2	Perte de parquets – Première experience [1].....	91
Figure 5.3	Gigue – Première experience [1]	93
Figure 5.4	Simulations Results in Germany50 (top) and Giant (bottom) networks [2]	94
Figure 5.5a	Délai -première experience [1]	100
Figure 5.5b	Délai avec ports désactivés Première expérience	101
Figure 5.6a	Perte de paquet- Première expérience [1]	101
Figure 5.6b	Perte de paquet avec ports désactivés - Première expérience	102
Figure 5.7a	Gigue -Première expérience [1].....	103
Figure 5.7b	Gigue avec ports désactivés Première expérience	103

LISTE DES TABLEAUX

Tableau 1.1	Classe des services en 3G [13]	23
Table 2.2	Résultats de MINNIE sur différentes topologies avec environ 1000 serveurs	48
Table 3.1	Estimation de la répartition de la puissance consommée des équipements des TIC dans le monde [89].....	54
Table 3.2.	Résumé des notations	61
Table 3.3	Arbre maximal de plus faible poids	71
Table 3.4	Evaluation de l'approche	74
Table 4.1	Résumé des notations	81
Table 4.2	Evaluation de la fonction exponentielle	85
Table 4.3	Evaluation de la fonction linéaire [94]	85
Table 4.4	Tableau récapitulatif des deux approches.....	86
Table 5.1	Résumé des notations	96

LISTE DES ABREVIATIONS ET SYMBOLES

ABREVIATIONS

2G	2 nd Generation of Mobile Network
3G	3 rd Generation Mobile of
3GPP	3 rd Generation Project Patnership
4G	4 th Generation of Mobile Network
ALR	Adaptive Link Rate
API	Application Programming Interface
ACK	Acknowledgment
AN	Active Network
ATM	Asynchrone Transfer Mode
BBU	Base Band Unit
CPN	Cognitive Packet Network
C-RAN	Cloud Radio Access Network
DVFS	Dynamic Voltage and Frequency Scaling
DCAN	Devolved Control of ATM Network
EDP	Ecole Doctorale Polytechnique
EPC	Evolved Packet Core

EPS	Evolved Packet System
eUTRAN	Evolved Universal Territorial Access Network
ForCES	Forwarding and Control Element Separation
FSs /SP	Fournisseurs des Services/Service Provider
GE	Gigabit Ethernet
GSM	Global System for Mobile communications
GSMP	General Switch Management Protocol
IBM	International Business Machine
ICMP	Internet Control Message Protocol
IF	Impact Factor
IEFT	Internet Engineering Task Force
ILP	Integer Linear Programming
IGP-WO	Integer Gateway Protocol Weight Optimization
IP	Internet Protocol
IoT	Internet of Things
JST	Scientific Technology Journal
KA	Kruskal Algorithm
LISP	Locator/Identifier Separation Protocol

LFB	Logical Function Blocks
LPI	Low Power Idle
LTE	Long Term Evolution of 3G
MAC	Medium Access Control
MILP-EWO	Mixed Integer Linear Programming based algorithm for Energy aware Weight Optimization
MPLS	Multiprotocol Label Switching
MSP/PSM	Modified SPRING Protocol / Protocole SPRING Modifié
NFV	Networking Function Virtualization
OCE	Optimisation de la Consommation Energétique
OMNET++	Objective Modular Network testbed in C++
ONF	Open Network Foundation
OnePK	Open Network Environnement Platform Kit
OSPF	Open Shortest Path First
PN	Programmable Network
PCE	Path Computation Element
PLNE	Programme Linéaire en Nombres Entiers
QoS	Quality of Service
RCSF	Réseau de Capteur Sans Fil

RDL	Réseau Défini par Logiciel
RMSF	Réseau Mobile Sans Fil
RNC	Radio Network Controller
RRH	Remote Radio Head
RSVP	Reservation Protocol
SAE	System Architecture Evolution
SDN	Software Defined Network
SDWN	Software Defined Wireless Network
STR	Suspend to Ram
STREETE	SegmenT Routing based Energy Efficient Traffic Engineering
TCAM	Ternary Content Addressable Memory
TE	Traffic Engineering
TICs/ICTs	Technologies de l'Information et de la Communication
UMRI	Unités Mixtes de Recherche et d'Innovation
UMTS	Universal Mobile Telecommunications System
UTRAN	Universal Territorial Access Network
VN	Virtual Networking
WNV	Wireless Network Virtualization

SYMBOLES MATHÉMATIQUES

N	Réseau
n	Sous -réseau
i	Routeur i
j	Routeur j
$\ell(t)$	Nombre de ports à désactiver à l'instant t
λ_{max}	Nombre total de ports sur un routeur
$P(n(t), t)$	Consommation du sous réseau à l'instant t
$P(i, t, \ell(t))$	Consommation du routeur i à l'instant t dont le nombre de ports $\ell(t)$ ont été désactivés
$P(k(i, j, t))$	Etat des liens dans un réseau
$C(i, j)$	Capacité des liens i, j
g	Fonction gain d'énergie du routeur i
$\max_t \ell(t)$	Maximum de ports désactivés à l'instant t
$g_i(t, \ell(t))$	Gain d'énergie du routeur i

$a\ell(t) + b$	Fonction linéaire caractérisant la fonction gain d'énergie
C_i	Consommation globale du routeur i
C_{ch}	Consommation du châssis
C_p	Consommation du port
$q(f(t))$	Qualité de service observée par demande
$k \in \{0, 1\}$	Etat du routeur. 0 : état éteint; 1: état actif
$F(t)$	Ensemble des demandes dans le réseau
$Q(f(t))$	Qualité de service fixé à chaque demande
$\sum f(t)$	Somme des demandes observées
$\sum f(t) \leq C_{i,j}$	Somme des demandes ne doit pas dépasser la capacité du lien
$D_{\ell(t)}$	Délai en fonction du nombre de ports désactivés
$D_{\text{délai}_n}$	Délai normal
$P_{0(u)}$	Puissance fixe

$\Delta_{p(u)}$

Facteur multiplicateur

$P_{t(u)}$

Puissance de Transmission

INTRODUCTION GENERALE

Contexte et motivations

Nous définissons le réseau comme étant un ensemble de matériels et logiciels mis en œuvre pour assurer les communications entre ordinateurs, station de travail et terminaux informatiques.

Depuis plusieurs décennies, l'énergie est devenue un enjeu très important dans les Technologies de l'Information et de la Communication (TIC). L'augmentation du nombre d'utilisateurs et de leurs différentes communications a entraîné une augmentation rapide du trafic de l'internet mobile, une évolution accrue des équipements réseaux. C'est le cas de parler des prévisions de Cisco [3]. Selon les auteurs, la croissance du trafic IP mondial pourrait être 1000 fois plus élevée en 2020 par rapport à 2010 [3]. Une étude du Cabinet Gartner a aussi montré que le nombre des objets connectés pourrait atteindre 80 milliards en 2020. Toutes ces données justifient, ainsi, l'augmentation de la consommation de l'énergie électrique dans les réseaux de communication qui pointent à l'horizon.

Les fournisseurs réseaux réfléchissent afin de trouver des solutions pour faciliter la gestion du trafic dans le réseau. Dans les réseaux mobiles, certains auteurs ont proposé l'idée d'extension du réseau d'accès radio pour les 5G. Concernant les réseaux informatiques, d'autres proposent des approches logicielles et virtuelles par la programmabilité des équipements à distance. Pour ces derniers, un nouveau paradigme de réseautage est né. On l'appelle SDN (Software Defined Network) qui vise à faciliter la gestion de réseau. Le SDN est le découplage du réseau en plan de contrôle et en plan de données. C'est un ou plusieurs contrôleurs centralisés ayant des interfaces API (Application Programming Interface) et des protocoles standards tels qu'Open flow, Netconf etc., qui vont permettre la gestion de tout le réseau. De par son fonctionnement, cette nouvelle technologie offre des possibilités d'améliorer les performances du réseau et d'optimiser la consommation de l'énergie. Des algorithmes sont utilisés pour agir à distance sur les équipements, afin d'éteindre ceux qui ne sont pas utilisés tout en redirigeant le trafic sur d'autres ports ou liens.

L'intelligence des routeurs leur sont retiré et donc ils deviennent que de simples dispositifs de transmissions des demandes ou des paquets dans le réseau. Les auteurs [4] ont montré que le SDN n'est pas une proposition révolutionnaire, mais surtout un mélange des propositions antérieures, notamment les réseaux programmables et la séparation du plan de contrôle et du plan de données. Ceux-ci ont donc affirmé la popularité actuelle du SDN dans les milieux industriels et de recherches.

Des auteurs [5, 6] ont montré dans une étude que les Technologies de l'Information et de Communication (TIC), représentent 2 à 10% de la consommation mondiale d'énergie [5]. Aux Etats-Unis, la consommation de l'énergie est de 6 Twh /an selon les auteurs [6].

La recherche dans ce domaine est devenue une préoccupation pour les chercheurs.

L'architecture ancienne présente plusieurs limites liées à la non programmabilité des routeurs déployés. Les limites sont d'ordre fonctionnels en terme de gestion de trafic et agissant de façon statique. Les routeurs effectuent seulement les tâches pour lesquelles ils sont conçus. Or, avec les prévisions actuelles dues aux objets connectés, le trafic dans le réseau s'avère dense et la congestion est plus que préoccupante. Nous pensons que le SDN apparaît, alors, comme une alternative pour agir à distance et de façon dynamique en vue de la minimisation de l'énergie en éteignant les équipements non utilisés.

La minimisation de la consommation de l'énergie dans les réseaux de communication a réduit la consommation de l'énergie plus ou moins, mais avec dégradation du QoS en terme de délai, de perte de paquets et gigue [1, 7, 8].

Dans cette Thèse, nous proposons ainsi une approche d'optimisation de la consommation énergétique basée sur un modèle mathématique et une nouvelle stratégie d'économie d'énergie dans les réseaux types SDN.

Problématique de la consommation de l'énergie dans les réseaux

Les réseaux de communication évoluent de manière progressive en taille et en puissance (performance). Cette évolution implique absolument la croissance du nombre d'utilisateurs, l'augmentation des équipements réseaux (routeurs, Switch, commutateurs), ainsi que le nombre des informations échangées. Ces équipements sont devenus de plus en plus volumineux avec des puissances de calculs élevées et un nombre important d'internautes dans le monde [9].

Cette augmentation des équipements et de leur performance, accroissent la consommation de l'énergie dont ils ont besoin pour leur fonctionnement. A titre d'exemple, selon Cisco Visual Networking index [10], plus de 563 millions d'appareils mobiles ont été ajoutés en 2015 et il est prévu 1.5 autres par personnes d'ici 2020. Dans la suite des exemples, nous pouvons citer les auteurs des revues [11, 12]. Les travaux des auteurs de la revue [11] montrent l'impact qu'ont les réseaux de communications sur l'environnement (augmentent le gaz carbonique). Dans un rapport publié, ils ont montré que les équipements réseaux dans leur ensemble contribuent à environ 1.25% de l'émission de gaz carbonique totale en 2002 (correspondant à 0.53 Gtons d'émission de CO₂) et 2% en 2007 (correspondant à 0.83 Gtons d'émission CO₂). Une prévision du même organisme a été publiée et estime que la contribution des infrastructures réseaux pour l'émission de gaz carbonique atteindra 2.7% en 2020 dans le monde (correspondant à 1.43 Gtons d'émission CO₂) [11]. Au vu de cette forte consommation énergétique des réseaux de communications, il apparaît clairement, où les travaux de recherche sont attendus [12]. L'architecture traditionnelle étant statique et non programmable, constitue une des difficultés dans l'acheminement des informations. Car le plan de contrôle et de données sont intégrés, ce qui rend les opérations de configuration complexes. Les administrateurs configurent équipement par équipement. Ce qui prend assez de temps tout en augmentant le risque d'erreur et de perte de temps. La congestion due aux retards de livraison de paquets est plus que préoccupante. L'architecture

nouvelle SDN apparaît comme une solution pour faciliter l'innovation dans les réseaux, simplifier la gestion et la configuration des équipements réseaux.

Objectifs

Le sujet de nos travaux de recherches s'intitule : << **Contribution à l'optimisation de la consommation de l'énergie dans les réseaux SDN**>>. Il s'agira de trouver dans cette étude, un modèle d'optimisation et une nouvelle stratégie d'économie d'énergie et valider ce modèle dans les réseaux SDN compte tenu des prévisions citées plus haut.

Quel modèle d'optimisation et stratégie trouver pour minimiser la consommation d'énergie sous contrainte de la qualité de service en termes de délai, de perte de paquets et gigue ? Quel pourra être l'apport du SDN dans la minimisation de la consommation de l'énergie dans les réseaux ?

Organisation de la thèse

Ce mémoire de Thèse est structuré autour de cinq (5) chapitres. Ainsi,

- **Le chapitre 1** traite les généralités sur les méthodes d'optimisation de la consommation de l'énergie dans les réseaux de communication ; notamment dans les réseaux filaires ou sans fil et les réseaux SDN. Par la suite, nous avons présenté des techniques d'optimisation pour la réduction de la consommation d'énergie. L'objectif est de montrer la nécessité de proposer une architecture nouvelle capable de faire face aux gros volumes de différentes communications dans le réseau. C'est le cas de parler des travaux de collaboration des auteurs Zamblé et al. [13]. Nous montrons l'importance du SDN dans l'optimisation de la consommation de l'énergie dans les 5G, sous contrainte de la qualité de service dans ce chapitre, en ayant présenté les différents types de réseaux et les réseaux SDN, puis suivra la conclusion.
- **Le chapitre 2** fait l'état de l'art sur les approches d'optimisation utilisées dans la consommation de l'énergie dans les réseaux de communication et notamment les SDN. Nous les avons classé en trois grands groupes : approche matérielle, approche

logicielle et approche hybride, c'est-à-dire combinaison des deux premières approches. Notre procédure de classification a permis de dégager les forces et limites des techniques utilisées dans l'optimisation de la consommation de l'énergie sous contrainte de qualité de service (QoS).

- **Le chapitre 3** est consacré à notre première contribution pour améliorer la consommation de l'énergie dans les réseaux SDN. Nous avons proposé un modèle mathématique et une nouvelle stratégie d'économie d'énergie. Nous, nous sommes focalisé sur une approche dynamique en nous appuyant sur les travaux de Gelenbe et al. [1]. Les chercheurs ont, dans leur contribution proposé un modèle d'extinction des liens sous contrainte de QoS.

- **Le chapitre 4** est une suite de la première contribution. Dans ce chapitre, la fonction "gain d'énergie" a été approximée à la fonction exponentielle. Le résultat obtenu est le même que dans le précédent.

- **Le chapitre 5** présente la simulation de notre modèle mathématique et la stratégie nouvelle d'économie d'énergie. Nous avons définis une fonction "délai en fonction du nombre de ports désactivés". Cette fonction, nous a permis d'obtenir de meilleurs résultats que dans la littérature.

Nous terminons nos travaux par une conclusion générale qui fait la synthèse des travaux effectués où un modèle de minimisation de l'énergie comme contribution d'économie d'énergie a été proposé. Des perspectives qui serviront de base à des travaux futurs ont été énumérées.

**CHAPITRE 1 : GENERALITES SUR LES
METHODES D'OPTIMISATION DANS LES
RESEAUX DE COMMUNICATION ET
DANS LES RESEAUX SDN**

1.1. INTRODUCTION

Face à l'augmentation de la demande de trafic due à l'avènement de l'IoT (Smartphones, Tablettes, Objets connectés etc.) et le Big Data, l'architecture réseau classique ne satisfait plus aux attentes des utilisateurs et clients. Le nombre d'utilisateur croît et le besoin en bande passante permettant aux demandeurs des réseaux mobiles d'échanger des messages ou informations est nécessaire. Le nombre d'équipements a aussi grossi pour suivre l'évolution de l'utilisation de ces réseaux. Nous constatons beaucoup de plaintes des usagers des réseaux de communication et aussi des administrateurs des réseaux informatiques. Ces plaintes concernent, entre autres, les problèmes de congestion, le délai long de transmission des messages dans les réseaux de communication [1, 14].

Dans le cas des autres réseaux tels les réseaux mobiles, capteurs, SDN etc., le délai long de mise sur le marché de nouveaux services, la disponibilité limitée du spectre et l'adoption massive du haut débit [13, 15], limitent ces types de réseau. Le problème d'interférence est préoccupant. Ces problèmes ont donc entraîné une consommation élevée de l'énergie électrique.

Face à ces problèmes, Les opérateurs de téléphonie mobile, réseaux informatiques, télécommunications sont unanimes que l'architecture traditionnelle soit revue afin de répondre aux besoins des utilisateurs pour satisfaire la qualité de service. La naissance du SDN et l'idée du futur réseau mobile 5G justifient bien cette volonté de changer les architectures anciennes qui sont statiques. A titre d'exemple, les auteurs [13] pensent que la virtualisation des fonctions de réseau (Networking Function Virtualization ou NFV) est une solution pour accroître la flexibilité du réseau afin de réduire le délai de mise sur le marché de nouveaux services. Ils trouvent de même que l'utilisation d'interfaces d'applications ouvertes et le partage de centres de données permettent de fournir de façon rentable un grand nombre d'applications et de services sur des réseaux haut débit fixes et mobiles et sur des plateformes de terminaux.

Dans ce chapitre, nous allons parler des généralités concernant les différentes méthodes utilisées dans l'optimisation de la consommation d'énergie dans tous types de réseaux ; notamment, les réseaux classiques (capteurs sans fil, maillés etc.) et de nouvelles générations (SDN et CRAN) appelés aussi, réseaux intelligents. Nous présentons d'abord, l'architecture de ces réseaux et ses limites, ensuite les raisons pour lesquels, des chercheurs comme Zamblé et al. [13] et bien d'autres ont orienté leurs travaux vers les réseaux intelligents et enfin nous présentons notre domaine de recherche c'est à dire le SDN.

1.2. Architecture des réseaux de capteurs sans fil et maillés

Dans cette partie, nous présentons les différentes architectures des réseaux (capteurs sans fil et réseaux maillés). Nous dégagons les insuffisances de ces réseaux puis suivront les techniques d'optimisation utilisées dans la littérature. L'évaluation des résultats pourrait nous guider dans le choix d'autres réseaux futurs.

1.2.1. Les réseaux de capteurs

Plusieurs travaux ont traité la problématique de la consommation de l'énergie dans les réseaux de capteurs. C'est le cas des articles [16, 17]. Des architectures ont donc été proposées en fonction de différentes définitions de chaque auteur.

Un réseau de capteur sans fil, comme le montre la **figure 1.1**, est formé de nœuds capteurs. Il est envoyé dans un environnement donné pour exécuter une tâche précise comme contrôle, surveillance, détection d'éléments intrus, etc. Le réseau de capteur peut être défini comme étant un réseau Ad hoc regroupant plusieurs micro-capteurs. Ces micro capteurs sont déployés de façon aléatoire dans un champ de détection donné. Les routeurs tout comme les capteurs communiquent entre eux en vue de transmettre des informations d'un nœud à une destination "Sink" (puits). Les données des capteurs sont envoyées vers une station de base afin d'être utilisée (**figure 1.2**).

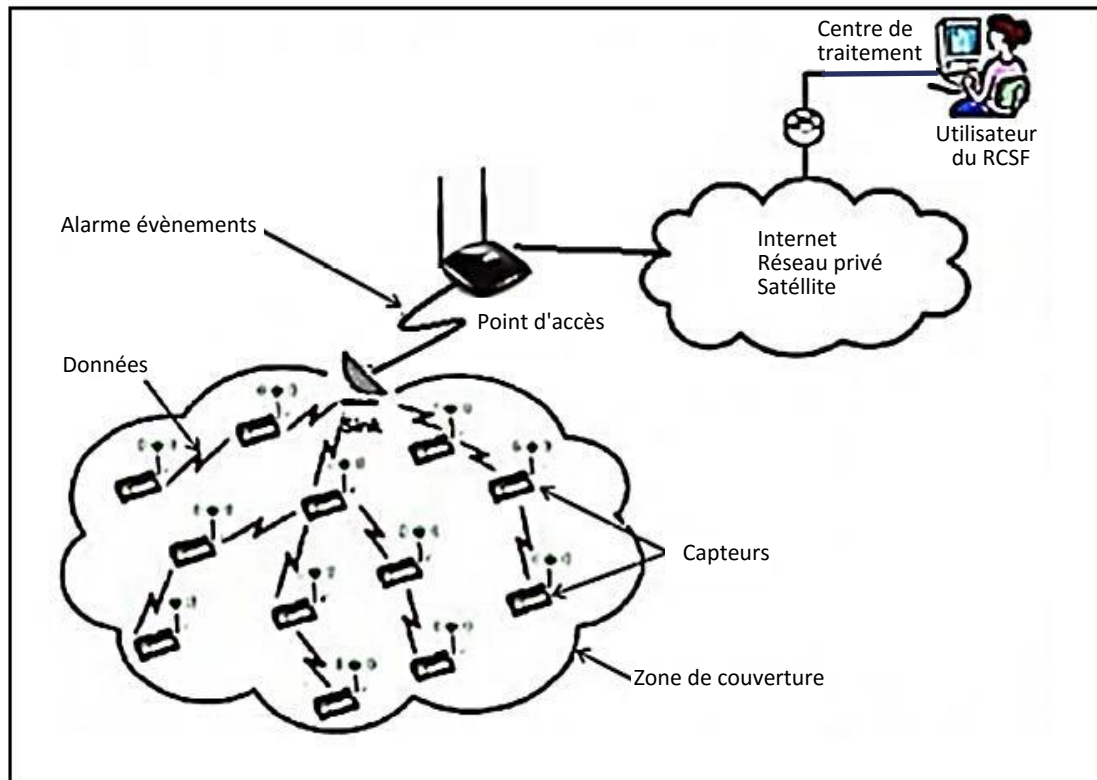


Figure 1.1 Architecture de réseaux de capteurs sans fil [16]

Un réseau de capteur sans fil (RCSF) ou WSN (Wireless Sensor Network) est aussi un ensemble de capteurs autonomes qui collectent des données, échangent des informations afin de véhiculer des messages d'une zone donnée. La position d'un RCSF est souvent aléatoire dans une zone de couverture ou "champ de détection".

Les données des capteurs sans fil sont transmises grâce à un routage mono ou multi-sauts vers la destination. Par exemple, suivant la (**figure 1.1**), l'utilisateur du RCSF, depuis un centre de traitement, se connecte à la destination "Sink" à partir des services du réseau internet ou un moyen satellitaire. Le comportement des capteurs sans fil est similaire aux réseaux ad hoc. Les réseaux ad hoc jouent le même rôle que les protocoles de routage à la différence que ceux-ci développent une intelligence individuelle. L'utilisateur utilise des requêtes, lorsque celui-ci a besoin d'informations dans la zone de couverture. Il envoie donc les éléments nécessaires (types de données) aux autres capteurs du réseau grâce à une requête d'inondation.

Elle consiste à envoyer une requête à tous les nœuds capteurs dans le réseau. Les captures de mesure sur des événements sont envoyées à la station de base depuis une zone de captage grâce à un routage multi-saut (voir Figure 1.2)

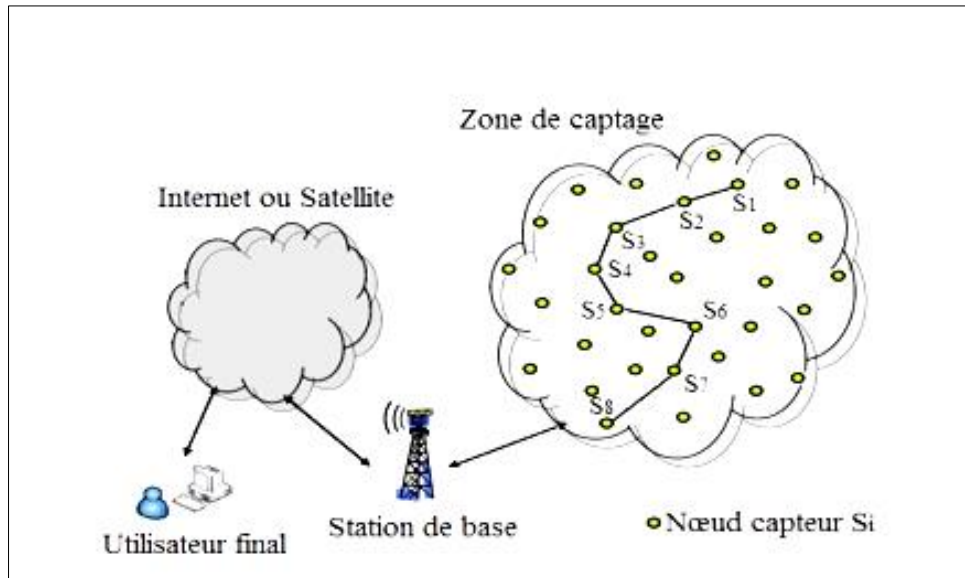


Figure 1.2 Architecture d'un réseau de capteur sans fil [17]

1.2.2. Les réseaux maillés sans fil

Les travaux de Ounis et al. [18] ont traité l'optimisation de la consommation de l'énergie dans les réseaux maillés. La similitude qu'il y'a entre les réseaux maillés et les réseaux de capteurs est que les deux types de réseaux fonctionnent comme les réseaux Ad hoc. Ces auteurs ont défini les RMSF (Réseau Maillé Sans Fil), comme étant un réseau qui permet d'interconnecter plusieurs équipements sans fil de sortes à constituer un maillage sans câble. La communication dans les réseaux maillés sans fil est possible à distance grâce aux relais radio. Dans les réseaux maillés, il s'agit des routeurs sans fil connectés par des liens radio appelé points d'accès. L'idée de tels réseaux fait penser aux réseaux mobiles, qui utilisent les stations de base pour relayer des communications à distance. La figure 1.3 ci-dessous présente l'architecture d'un réseau maillé. Les points d'accès sont chargés de collecter les demandes des clients mobiles/fixes. Ces données sont envoyées vers les passerelles (gateway) qui sont connectés aux réseaux externes fournissant différents services.

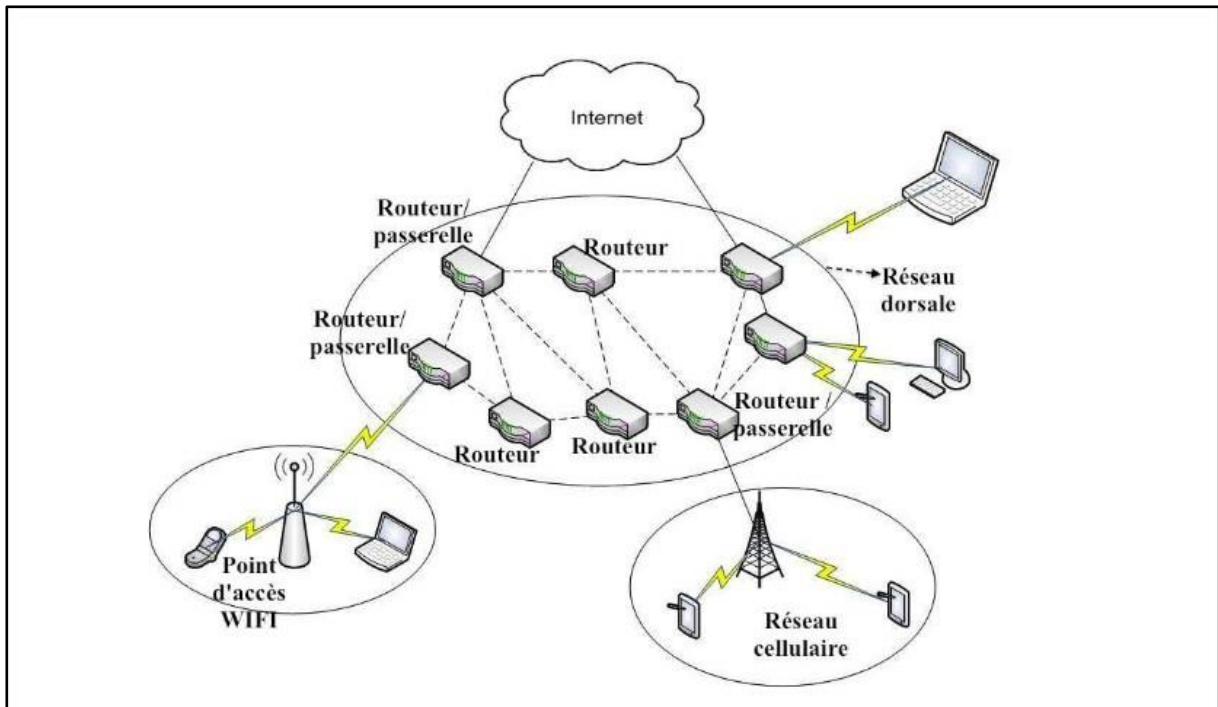


Figure 1.3 Architecture d'un réseau maillé sans fil [18]

1.2.3. Les limites

La consommation de l'énergie, l'inadaptabilité des protocoles et architectures aux réalités actuelles sont des problèmes que l'on rencontre sur les réseaux de communication (capteurs sans fil, maillés réseaux sans fil etc.). Les protocoles sont anciens et doivent être améliorés.

Les batteries des capteurs sans fil ne sont pas rechargeables, donc leur durée de vie est limitée. Le problème d'encombrement limite la capacité du réseau maillé sans fil. La durée de vie du réseau limité est une faiblesse pour les RCSF. Nous notons aussi, la bande passante faible et le délai long d'acheminement de paquets entraîne la perte de plusieurs données ou informations.

Dans la suite de notre travail, nous présentons le concept de l'architecture de base des réseaux classiques et celle des SDN.

1.3. Concepts et architecture de base des réseaux classiques

Dans cette partie, nous présentons l'ancienne architecture. Le schéma de l'architecture se présente comme suit (voir figure 1.4 ci-dessous). Nous présentons l'architecture traditionnelle et SDN de sorte à les comparer.

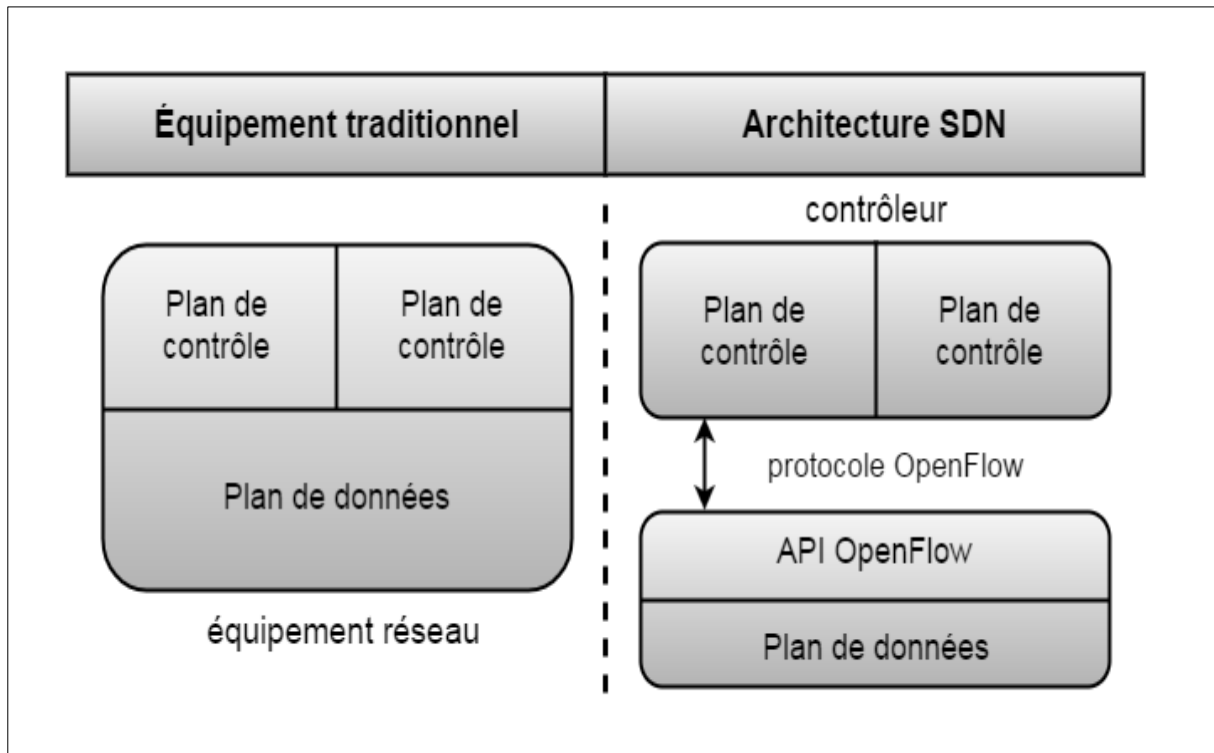


Figure 1.4 Architecture comparée des réseaux SDN et traditionnels de [19]

1.3.1. Fonctionnement de l'architecture classique

Traditionnellement, un réseau informatique est composé d'équipements réseaux interconnectés tels que les routeurs (voir figure 1.4). Nous trouvons dans chaque équipement un mécanisme d'acheminement dans la couche de transmission de paquets et un plan de contrôle qui incorpore le système d'exploitation et les applications. Les entrées et sorties des équipements sont accessibles mais leur fonctionnement interne est masqué d'où l'idée de "boîtes noires" c'est-à-dire fermé.

1.3.2. Limites des réseaux classiques

Nous notons comme limites aux réseaux classiques, que les interfaces de contrôle sont beaucoup limitées et sont spécifiques aux fournisseurs. Les codes des applications permettant de configurer les équipements réseaux ne sont pas accessibles. Tout Ceci rend l'évolution difficile ainsi que le déploiement de nouveaux protocoles. Afin d'ouvrir les équipements réseaux et de séparer les applications de la couche de transmission, le SDN a vu le jour.

1.4. Concept des SDN

Le concept de SDN selon ONF [Open Network Foundation] (un consortium d'entreprise à but non lucratif fondé en 2011 consacrée à la promotion et l'adoption du réseau défini par logiciel (SDN) à travers des standards ouverts de développement) est une architecture qui sépare le plan de contrôle du plan de données [20]. ONF est constituée de plus de 100 entreprises (les géants du web, les opérateurs de télécom et les fabricants) d'où son intérêt pour les industriels.

Pour la compréhension de notre sujet, nous définissons de façon détaillée ses différents termes.

1.4.1. Définitions des termes du sujet

1.4.1.1. SDN

Les réseaux SDN, "Software Defined Network" désigne un réseau défini ou plus exactement piloté par logiciel. En d'autres termes le SDN est une interface logicielle qui doit permettre d'orchestrer un réseau physique, c'est-à-dire diriger les flux, modifier la bande passante, les règles d'un Switch etc. sans devoir accéder au matériel et de façon physique [21].

1.4.1.2. La congestion

Selon les auteurs [22], la congestion se manifeste lorsqu'un dispositif reçoit plus d'informations que ce qu'il ne peut traiter, plus précisément, dès lors que le débit de données entrant est supérieur à celui sortant. Autrement dit, lorsque le flot entrant est supérieur à la capacité du lien.

Selon le dictionnaire Larousse 2018 en ligne, la congestion se définit comme état d'un lieu de passage, d'un réseau qui est fortement encombré [23].

Nous pouvons même définir la congestion du réseau comme la condition dans laquelle une augmentation du trafic (flux) provoque un ralentissement global de celui-ci (le goulet d'étranglement).

1.4.1.3. La consommation d'énergie

La consommation de l'énergie du réseau peut résulter de plusieurs facteurs, par exemple le délai de communication, la dissipation énergétique, le protocole de communication etc. Ces raisons peuvent augmenter la consommation du réseau de manière significative [22].

D'après la définition du CNRTL (Centre National de Ressource Textuelles et Lexicales), la consommation de l'énergie est l'action d'amener l'énergie à son maximum [24].

1.4.1.4. L'optimisation de la consommation de l'énergie

L'optimisation de la consommation énergétique consiste à minimiser l'énergie sous des contraintes ou réduire l'augmentation de l'énergie. Nous proposons un modèle mathématique et une nouvelle stratégie de minimisation de la consommation de l'énergie dans les réseaux informatiques.

En effet, les problèmes liés à la qualité de service sont souvent la congestion et la consommation. Plus le délai de transmission des paquets augmente, plus il y'a possibilité de perte de paquet due à la congestion et donc consommation d'énergie dans le réseau. Sachant qu'on ne peut pas optimiser sans dégradation de la qualité de

service, notre travail serait de proposer notre approche d'économie d'énergie. Cependant, le fonctionnement des différentes parties du SDN est nécessaire à connaître.

1.4.2. Rôle des différentes couches du SDN

L'architecture du SDN se compose de trois (03) couches :

- la couche application;
- la couche contrôleur;
- la couche infrastructure.

1.4.2.1. Couche application

Elle concerne celle des développeurs. Au niveau de cette couche, des applications nécessaires au contrôleur SDN sont développés via les interfaces de programmation d'application (API). Il existe deux interfaces:

L'interface nord ou North-bound, dirigée vers la couche supérieure qui favorise la communication entre le contrôleur et les applications.

L'interface sud ou South-bound, est dirigé vers la partie sud (les équipements) et qui permet la communication entre le contrôleur et les infrastructures grâce à l'open flow (protocole standard de l'ONF). Les applications développées peuvent s'agir de la gestion du réseau, l'équilibrage des charges, le routage, la qualité de services.

1.4.2.2. Couche contrôleur

Le contrôleur SDN est l'organe intelligent qui récolte des informations sur tout le réseau. Il offre une vue centralisée globale et envoie des instructions aux infrastructures. Le contrôleur reçoit des instructions ou exigences de la couche application et les transmet aux équipements vice-versa. Cette couche facilite le contrôle du trafic. Il peut ajouter, modifier ou supprimer des règles dans la table de routage via l'interface API Open flow, de manière réactive c'est-à-dire, lorsque le routeur reçoit un paquet, il envoie un signal au contrôleur pour demander les règles qu'il lui faut ou proactive (les règles sont d'avance installées dans le commutateur).

1.4.2.3. Couche infrastructure

Cette couche exécute les comportements traduits en termes d'instructions envoyées par le contrôleur. Lorsqu'un équipement reçoit un paquet, il examine son entête et en fonction de la logique du plan de contrôle exécute les actions. Le protocole, le plus utilisé et déployé vers les données est Open flow. C'est un composant fondamental pour apporter des solutions SDN. La mise à jour de l'Open flow est du ressort de l'ONF. Sa dernière version est 1.5.0 [25]. Les tables contenues dans le protocole sont spéciales appelées "Tables des flux" ou "Flow Table (FT)". Elles sont similaires à la table MAC, à la seule différence qu'elles stockent et gèrent des règles correspondantes à chaque flux au lieu des adresses MAC. Nous donnons un peu de détails sur Open flow et son mode de fonctionnement.

La figure 1.4 ci-dessous, illustre le processus de communication entre un contrôleur et un commutateur qui supporte le protocole Open flow pendant l'installation de nouvelles règles.

D'autres API existent à part Open flow. C'est le cas par exemple du Framework ForCES (Forwarding and Control Element Separation) [26]. Il a été proposé par l'Internet Engineering Task Force en 2004, c'est-à-dire avant l'arrivée du SDN dans le cadre de développement d'un nouveau protocole pour rendre standard l'échange des informations entre le plan de contrôle et celui de transmission. Le comportement des données (éléments de la transmission) sont modifiés par les données de contrôle en utilisant le protocole ForCES via les LFB (Logical Function Blocks). Ces données ou éléments peuvent s'organiser par module pour former des moyens de transmission plus complexe. Une architecture appelée SoftRouter [27] sépare aussi les fonctions du plan de contrôle et de celles du plan de données de façon dynamique. Ces deux API sont apparus avant SDN, mais ils peuvent cependant jouer le rôle d'Open flow. Nous pouvons aussi citer d'autres catégories dans la technologie réseau telles que PCE(Path Computation Element) qui sont des éléments de calculs de chemins et le protocole LISP (Locator/Identifier Separation Protocol) [28]. Il existe encore une

variante de protocoles sud utilisés par des industriels tels que Cisco et Junifer comme Open Network Environnement Platform Kit (OnePK) et Contrail [29].

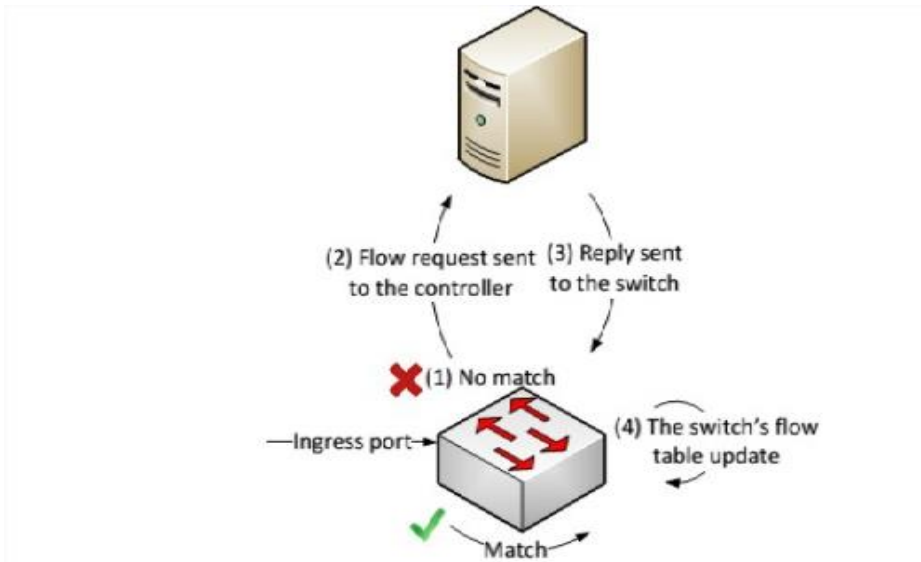


Figure 1.5 Processus de communication entre les commutateurs et le contrôleur et un switch supportant le protocole Open flow

D'après les travaux de [30], le processus de communication entre les commutateurs et le contrôleur a été mentionné. Quand un paquet est reçu par le commutateur, le champ d'entête du dit paquet est examiné et comparé avec le champ 'Match' dans les entrées de la table de flux. Une fois le Match identifié, l'action correspondante est exécutée dans la table de flux. Mais, s'il n'y a pas de Match (1), une demande est envoyée au contrôleur (2) sous forme de 'paquet in', puis le contrôleur décide selon sa configuration une action pour ce paquet et envoie une nouvelle règle de transmission sous forme d'un 'packet out' et 'Flow_mod' au commutateur (3). Enfin, la mise à jour est faite dans la table de flux (4).

La section précédente a permis de connaître les causes de la consommation de l'énergie dans les réseaux voire les concepts et différentes architectures, quelles sont les techniques d'optimisation proposées pour son économie par les chercheurs ?

1.5. Techniques d'optimisation utilisées dans les réseaux

1.5.1. Les réseaux de capteurs

Tout objet consommateur d'énergie, a une durée de vie dans son environnement. Les RCSF sont des objets utilisant des piles non rechargeables dans le réseau. Dans les limites, la durée de vie et l'inadaptabilité des protocoles de communication aux réalités actuelles sont à la base de la consommation de l'énergie dans les réseaux de capteurs sans fil. Nous citons par exemple les travaux d'Ibrahima et al. [16], qui ont dans la littérature, insisté sur la durée comme métrique la plus importante dans l'évaluation de la performance d'un réseau de capteur. Dans le même ordre d'idée, Diery et al. [17] ont montré que l'optimisation de la durée de vie dans un capteur sans fil, doit tenir compte de la couverture de la zone d'intérêt et de connectivité du réseau. La plupart des difficultés soulevées par les auteurs dans l'utilisation de RCSF, sont entre autres, la performance de l'énergie des surfaces (plates-formes) et des protocoles de communication.

1.5.1.1. La durée de vie du réseau de capteur sans fil

Selon Giridhar et al. [31], la problématique de la durée de vie d'un réseau de capteur dépend des protocoles utilisés et l'application. Une réalité dans la littérature est que, la vie du réseau de capteurs est la période pendant laquelle, le réseau peut maintenir plus de connectivité, couvrir toute la zone de captage et aussi minimiser le taux de perte suivant un seuil donné. Dans les travaux de [8], l'origine de la consommation de l'énergie se situe pendant la détection, la communication et le traitement des données. Parmi les différentes définitions de la littérature, nous constatons une convergence de certaines d'entre elles. Par exemple, les auteurs des articles [32], [33] et [34] définissent la durée de vie comme celle pendant laquelle le réseau répond de façon continue aux exigences de l'application. Elle est définie comme étant l'activité du nœud c'est-à-dire le temps début de mise en service du réseau jusqu'à épuisement du premier nœud. A présent, d'autres définitions doivent être compléter par les

chercheurs, puisque aucune d'entre elles ne prennent en compte les difficultés des moyens de son utilisation et les problèmes environnementaux. D'autres techniques ont été proposées dans la réduction de l'énergie dans les RCSF.

1.5.1.2. Duty-cycling

Cette technique est définie comme étant l'intervalle de temps où les nœuds sont actifs. Un moyen plus efficace de faire des économies d'énergie est de mettre la radio de l'émetteur en mode veille, c'est-à-dire (low-power idle) lorsque la communication n'est pas importante. Dans cette technique, chaque fois que le réseau est en attente de données, il peut être en état "sommeil" jusqu'à réception de paquet où il passe en état "actif". Les capteurs font des travaux collaboratifs. Un algorithme d'ordonnancement (actif/sommeil) est mis en place pour assurer le Duty-cycling. Cela a donc conduit Younis et al. [35] à affirmer que le module radio peut être dans quatre états différents qui sont "actif, réception, transmission et sommeil". La consommation de l'énergie dépend de chaque état :

- Actif : le module radio est allumé mais inutilisé. Pas de transmission, ni réception. Mais l'écoute du canal de transmission fait consommer de l'énergie.
- Réception et Transmission : le module radio reçoit et transmet des paquets.
- Sommeil : le module radio est off. L'énergie à cet état est faible (énergie de transition) (voir figure 1.6).

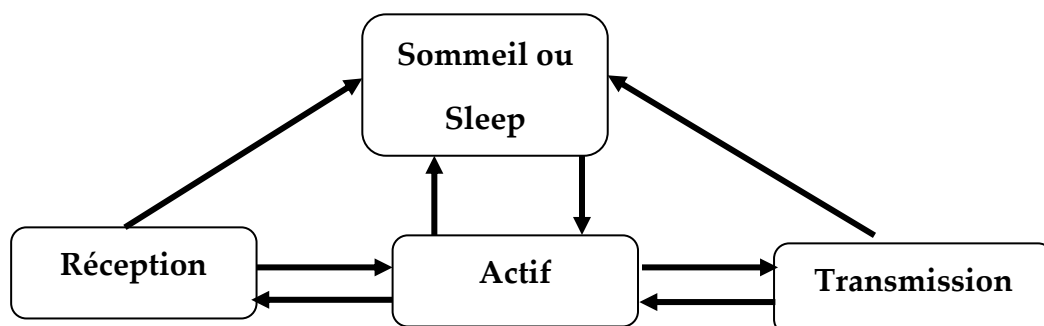


Figure 1.6 Diagramme de transition des différents états d'activités d'un nœud RCSF

1.5.2. Les réseaux maillés

Nous présentons quelques techniques utilisées dans la littérature pour l'optimisation de la consommation de l'énergie dans les réseaux maillés sans fil.

Dans les réseaux de capteurs, l'interface radio est celle qui consomme le plus de l'énergie avec plus de 95% représentant la consommation de la batterie [36].

A travers la figure 1.4 des auteurs [36, 37], ont présenté les différents états de consommation de l'énergie de la radio (transmission, réception, écoute passive et sommeil). Ceux-ci ont trouvé une similitude avec les réseaux Ad hoc qui, lorsque le nœud est en état de réception et transmission, consomme le plus.

Les auteurs [37, 38], ont proposé un modèle d'économie d'énergie dans les réseaux cellulaires en considérant les états d'activités des types de station (macro, micro, pico etc.). Ceux-ci considèrent une puissance $P_0(u)$ fixe, lorsque la station n'est pas active. Si elle est en état transmission, la consommation serait égale à:

$$P_0(u) + \Delta p (u) * P_t(u) \quad 1.1$$

Cette consommation va dépendre de P_t et de $\Delta p (u)$ (facteur multiplicateur) qui optimise en fonction de la puissance de transmission la consommation de l'amplificateur (voir figure 1.7).

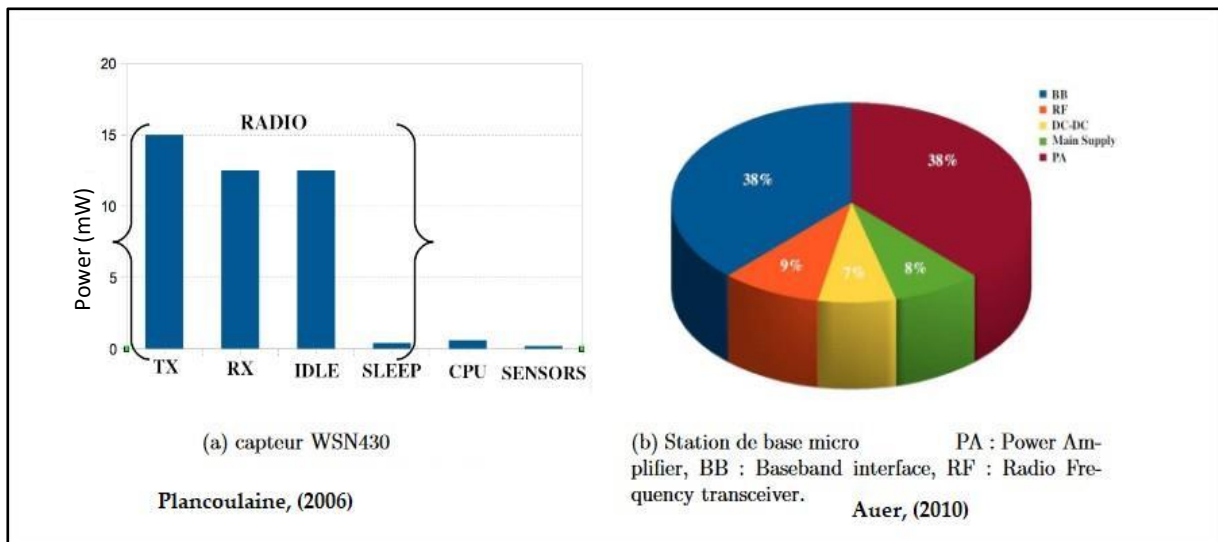


Figure 1.7 Modèle de consommation d'énergie [36, 37]

Parmi les réseaux déjà cités, des auteurs ont orienté leur recherche également vers d'autres types de réseaux appelés réseaux futurs 5G. Nous présentons d'abord les réseaux 2G, 3G, 4G actuellement utilisés puis suivra les réseaux mobiles 5G.

1.6. L'évolution vers les réseaux mobiles 5G

Dans cette partie, nous présentons brièvement les réseaux mobiles de deuxième génération 2G aux réseaux 4G en passant par les 3G. Nous montrons comment la zone des fréquences radio ou spectre pourrait être utilisé dans chacun d'eux.

1.6.1. Les réseaux 2G

Notons que les réseaux 2G sont nés dans les années 1990. Cette époque est appelée l'ère du GSM (Global System For Mobile Communications). Le rôle de ces réseaux était de passer du mode voix au mode commutation de circuits, de données à un débit réduit [39]. La fréquence à laquelle les communications se font varie entre 900 et 1800 MHz avec une largeur de bande de 25MHz ou 75 MHz. Cette bande est découpée en sous canaux espacé de 200kHz (bande de réserve) pour éviter les interférences. Un sous canal est partagé en 8IT (Intervalle de Temps). Pour faire face à la demande de nouveaux clients, les opérateurs mobiles construisent de nouvelles stations de base (BS). Ceci va accroître la couverture du réseau. Comme l'objectif du

GSM n'étant pas le transfert des données à haut débit, cela est un problème lié aux réseaux 2G. La mauvaise utilisation de ces stations de base la rende limitée. Car au moment où la demande des clients est élevée dans certaines zones, elle est sous exploitée ailleurs. La limite des réseaux 2G a conduit à la naissance de la 3G

1.6.2. Les réseaux 3G

Le réseau 3G est aussi appelé UTRAN (Universal Territorial Access Network). Il est constitué de station de base Node B et de RNC (Radio Network Controller) [40, 41, 42]. Mais dans la littérature, parmi les normes 3G utilisées, la norme européenne est celle qui fait mention dans notre étude : UMTS (Universal Mobile Télécommunications System). L'avantage du 3G par rapport au 2G est qu'il offre des débits très élevés allant de 2Mbps pour l'utilisation fixe, 384 Kbps dans le cas d'un piéton et 144Kbps en cas d'utilisation mobile. La 3G, à la différence de la 2G, introduit la qualité de service par la définition de quatre (04) classes de service résumé dans le tableau ci-dessous.

Tableau 1.1 Classe des services en 3G [13]

	Classe	Caractéristiques	Exemples
Applications à contraintes de temps réel	Classe A (Conversational)	Garantie des délais de transfert < 200ms	❖ Voix ❖ Vidéophonie
	Classe B (Streaming)	Délai plus long mais QoS garantie	Streaming Multimédia
Applications sensibles aux erreurs de transmission	Classe C (Interactive)	Garantie de l'absence d'erreur	❖ Navigation web ❖ E-commerce
	Classe D (Background)	Informations transmises avec la plus faible priorité	❖ Fax ❖ Notification d'email de sms

Le mauvais compromis entre l'efficacité spectrale et la réutilisation de fréquences limite les réseaux 3G. Tout comme les réseaux 2G, certaines BS sont sous exploitées à

des moments de la journée. Aussi, la multiplicité des objets connectés, la 3G et ses évolutions limitent la demande de ces réseaux en haut débit.

1.6.3. Les réseaux LTE/4G

Le réseau d'accès 4G est appelé LTE (Long Term Evolution of 3G) ou eUTRAN (Evolved UTRAN) [40], désigne l'évolution du réseau d'accès radio de la 3G pour résoudre efficacement la problématique des demandes à haut débit, qui limitaient les réseaux 3G et leurs évolutions. En plus du réseau d'accès, les réseaux de cœur connaissent aussi des modifications. Le réseau est appelé EPC (Evolved Packet Core) ou SAE (System Architecture Evolution). L'ensemble formé par LTE+EPC = EPS (Evolved Packet System) (voir figure 1.6 et figure 1.8).

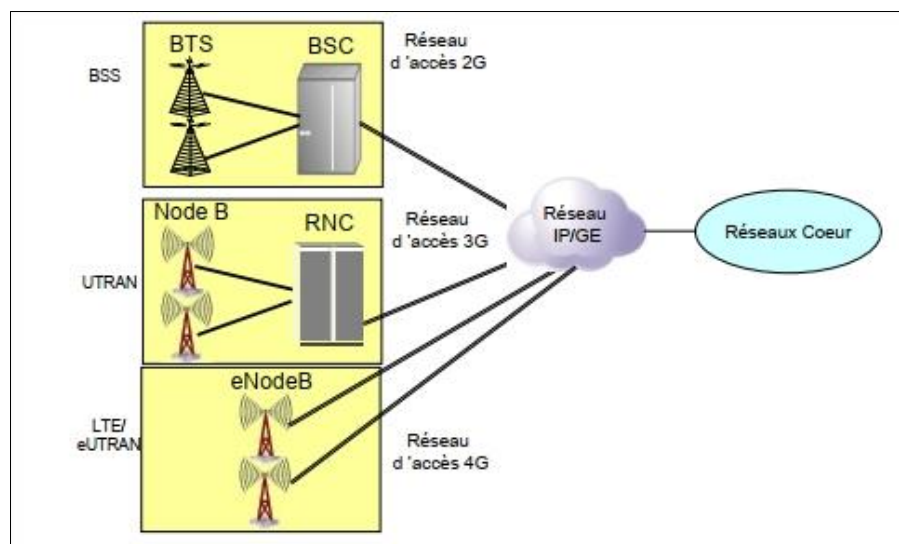


Figure 1.8 Réseau d'accès [40]

Le réseau cœur 2G/3G est formé des domaines de circuit et paquet. Le domaine de circuit est le cas par exemple de services de téléphonie. Lors d'une communication, la ligne est occupée jusqu'à la fin de l'appel. Le trafic entre le réseau d'accès et le réseau de cœur est transporté via un backbone IP sur Gigabit Ethernet (GE) (voir figure 1.9).

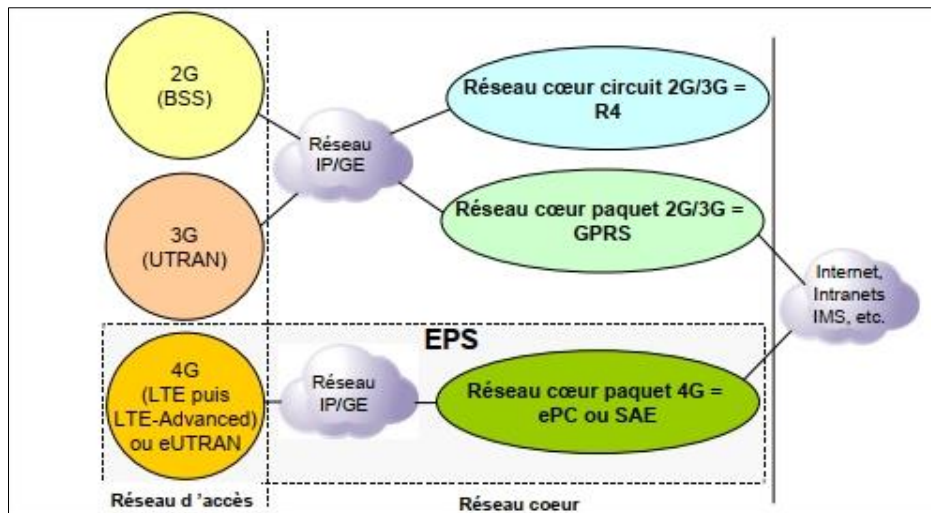


Figure 1.9 Réseau de cœur [40]

Bien que la LTE/4G ait apporté une forte amélioration à la 2G et 3G, les opérateurs n'entendent pas se limiter aux 4G. Car la problématique de l'efficacité spectrale et les interférences demeurent. Des moyens de l'amélioration de la capacité du système afin de réduire la consommation de l'énergie pour mieux gérer les interférences ont été abordées dans la littérature. Parmi ces moyens, la construction de nouvelles stations de base. Etant donné que l'extension est chère et difficile pour sa construction, Bouguen et al. [43] ont montré que la diminution de la taille des cellules peut réduire les puissances d'émissions des équipements et augmenter la durée de vie des batteries. L'amélioration de l'efficacité spectrale et l'efficacité énergétique des LTE/4G va dépendre de la réduction de la distance émetteurs et récepteurs. Ceci conduit donc à un autre type de réseaux hyper rapide avec des débits de l'ordre de 5Go/s grâce à une bande passante de 28 GHz. Ce type de réseaux doit pouvoir fonctionner avec différents types d'équipements ou objets de fonctions variées.

1.7. Les réseaux mobiles 5G

Les limites des réseaux 2G, 3G, ont poussé les opérateurs réseaux à mettre en place la LTE/4G afin de résoudre la problématique spectrale. Cependant, les interférences demeurent lors des différentes communications. Plusieurs moyens se mettent peu à peu en place pour faire face aux défis futurs de la consommation de l'énergie. C'est le

cas de la 5G déjà déployé dans certains pays et dans d'autres d'ici 2020, avec des débits allant jusqu'à 5Go grâce à une bande passante de 28 Ghz [44]. Cette technologie de télécommunication sans fil promet d'apporter une amélioration dans la manière dont le monde communique. Les auteurs Mugen Peng et al. [45, 46], ont donc montré que le futur réseau devrait fournir des débits binaires, 10 à 100 fois supérieurs à ceux de la LTE/4G et à réduire la consommation de l'énergie d'environ 90 %. Avec l'arrivée des IoT, véhicules autonomes, robots chirurgicaux, toutes les communications mobiles seront couvertes par une nouvelle tranche de bande passante située entre 6 Ghz et 300 Ghz. Une partie de cette coupe sera sous licence, mais une grande partie libre. Une architecture physique sera séparée de celle logique ou "virtuelle". En fonction de la demande mise en place, la gestion complexe en réseau sera effectuée par programmation (voir figure 1.10) : le C-RAN (Cloud Radio Access Network).

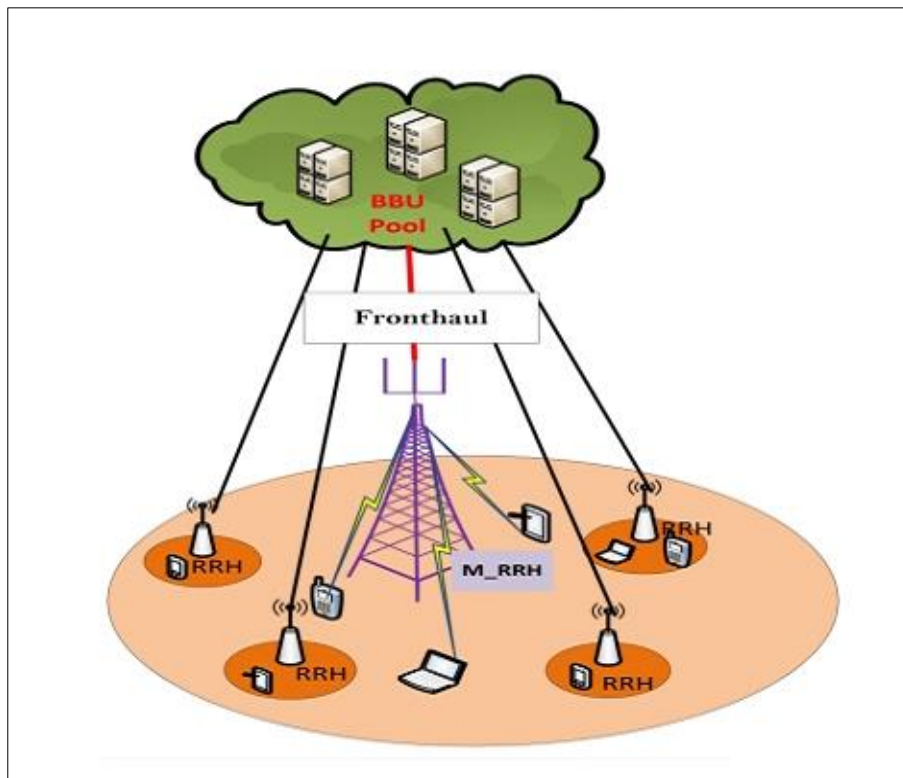


Figure 1.10 Architecture C-RAN [41]

1.8. Les réseaux SDN

Ces dernières années ont montré une augmentation rapide de données [3]. Cela est dû à la multiplicité des Smartphones, tablettes, objets connectés etc. Le nombre d'utilisateurs les utilisant a aussi augmenté. Une fois, ces équipements sont interconnectés, il y'a consommation d'énergie et congestion dans le réseau qui entraîne un délai élevé de transmission de paquets ou informations. L'architecture ancienne ne dispose pas de ressources suffisantes pour son traitement. Des chercheurs ont trouvé l'idée d'envisager une nouvelle architecture basée sur des structures dotées d'expériences (intelligence) où le plan de contrôle du routeur est séparé du plan de données (**voir figure 1.11**): SDN (Software Defined Network ou Réseau Défini par Logiciel (RDL)). Dans la thèse [30], les auteurs ont fait mention de certains prédécesseurs du SDN qui prônent la séparation des plans contrôle-données : 4D [47] et Ethane [48]. Dans notre thèse, nous rappelons l'historique des réseaux SDN comme ce fut le cas de ceux de [30]. Cependant, il est important de parler de ses modèles de programmabilité.

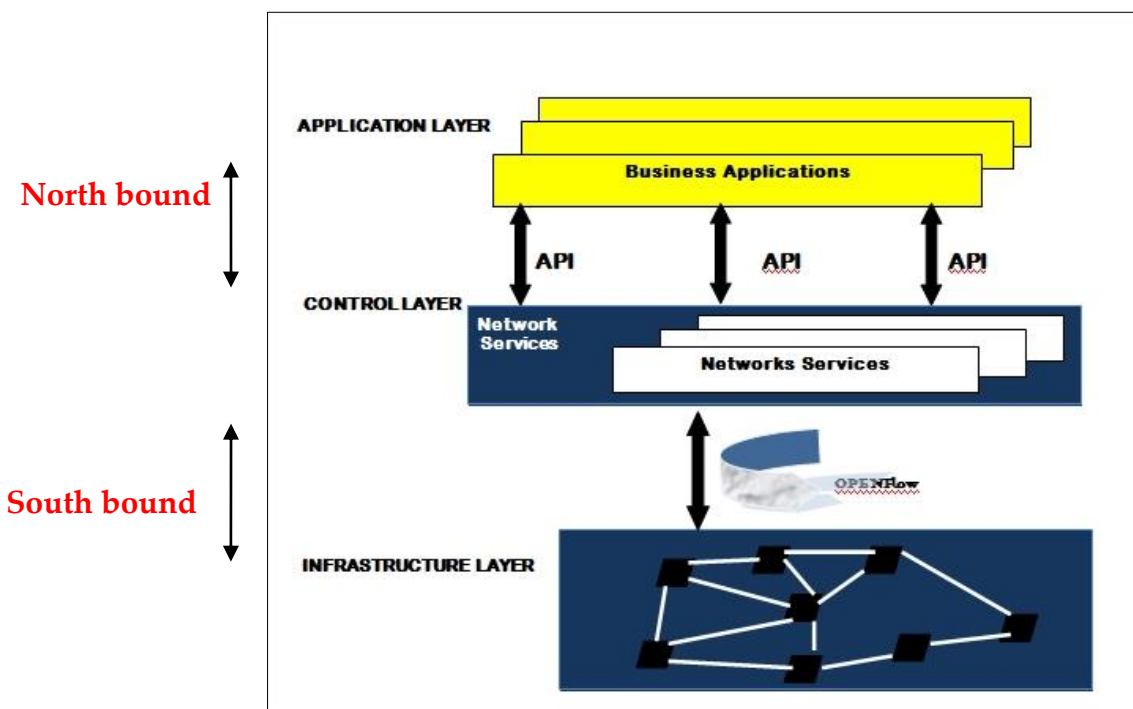


Figure 1.11 Architecture du SDN avec Open flow

Le SDN utilise trois différents modèles de programmabilité :

1. La programmabilité individuelle de chaque équipement. Dans ce cas, une application interagit directement sur un équipement via les interfaces API ou localement pour réaliser des tâches.
2. La programmabilité via un contrôleur. Un ordre abstrait et global est donné à un contrôleur. Le contrôleur traduit ces ordres en une suite d'instructions auprès des équipements du réseau concerné. Ce type de modèle est le plus utilisé puisqu'il simplifie le réseau. La complexité du réseau est surtout masquée par le contrôleur.
3. La Création d'un réseau virtuel au-dessus du réseau physique. Les applications créent leur propre réseau "overlay". On parle de virtualisation des fonctions réseau.

1.9. Historique des réseaux SDN

Dans cette section, nous présentons les idées et travaux déjà proposés dans la programmation du réseau et la séparation des plans de contrôle et celui de données. En 1996, la première idée a vu le jour nommé <<réseaux actifs>> AN (Active Network) [49]. Le rôle de ces réseaux est de développer des programmes à partir des données du paquet. Les routeurs sont chargés d'exécuter ces programmes afin de procéder à leur acheminement, modification ou la suppression de ce paquet. De nombreuses études [50, 51, 52], ont aussi été menés sur les réseaux actifs précisément sur des paquets intelligents. La limite des paquets nommés "ANs" due aux transports possibles de programmes malveillants, une solution aux réseaux actifs a été proposé <<réseaux programmes>> PN (Program Network) [53]. Leur rôle est d'injecter des programmes à l'intérieur des nœuds du réseau. Ces nœuds assurent la sécurité par exécution de programmes lorsque la phase d'identification et de vérification a été observée. Les deux réseaux actifs et programmables respectivement ANs et PN malgré la programmabilité introduite dans les réseaux dans les paquets et commutateurs programmables n'ont pu réduire la complexité de l'équipement réseau.

Certains projets ont tenté de séparer le plan de contrôle du plan de données afin de simplifier l'architecture réseau et fournir une abstraction sur l'équipement physique. C'est le cas par exemple du projet de recherche appelé DCAN (Devolved Control of ATM Networks) [54] de l'université de Cambridge qui a pour objectif de développer des fonctions de contrôle et de gestion entre plusieurs équipements de sortes à découpler l'équipement physique et permettre à une entité externe comme gestionnaire. Un problème de protocole pour faciliter la communication entre la partie physique et logique était nécessaire. Un atelier organisé par le groupe de travail OPENSIGN a permis de rendre les réseaux ATM, Internet et mobiles plus ouverts, extensibles et programmable [55]. OPENSIGN permet de noter une différence entre l'équipement physique qui achemine les demandes (paquets), le contrôle et la gestion du réseau. L'objectif de cette proposition était de faciliter l'accès aux équipements physiques à travers des interfaces (API) programmables et ouvertes en laissant aux fournisseurs de services de faire le choix des applications intermédiaires. Ces idées ont donc permis au groupe de travail de l'IETF (Internet Engineering Task Force) de proposer un protocole général de gestion des commutateurs GSMP (General Switch Management Protocol) [56]. En 2005, un projet nommé 4D a été lancé [47] pour la séparation des décisions de routage et les protocoles qui permettent la communication de façon interactive entre les équipements réseaux. Les plans de diffusion et de découvertes sont utilisés pour collecter les informations sur le réseau et les envoient à la partie décisionnelle du réseau qui a une vue globale sur l'ensemble du réseau afin de contrôler la transmission du trafic circulant dans le plan de données. C'est donc à l'université de Stanford que démarre les réseaux SDN à travers le projet Ethane en 2006. Ethane (projet) [48] a défini une architecture nouvelle pour les réseaux d'entreprises. Son objectif était d'avoir un contrôleur centralisé pour gérer les règles et la sécurité dans le réseau. Les composants utilisés par Ethane sont : un contrôleur pour la décision de transmissions et un switch Ethane ayant une table et un canal de communication entre les deux. Les réseaux Nox [57] ont vu le jour grâce à l'idée de Ethane. Cette idée

est devenue un concept aujourd'hui appelé <<réseaux programmés par logiciel>> (SDN).

1.10. Importance du SDN pour l'Optimisation de la Consommation de l'énergie dans les autres types de réseaux.

Selon les auteurs de [15], les réseaux définis par logiciels (Software Defined Networking, (SDN)) permettent d'introduire l'intelligence dans les réseaux de communication. Elle offre (l'intelligence) aux FSs (Fournisseurs des Services), la possibilité de contrôle efficace des différentes composantes de leur réseau. Elle facilite l'augmentation de l'efficacité du réseau physique en vue de faire fonctionner l'infrastructure logique. Le SDN permet ainsi de programmer les différentes composantes du réseau. Dans le SDN, le plan de contrôle et le plan de données d'un routeur ou nœud physique sont découplés. Cette flexibilité permet aux FSs, la possibilité de programmer et rendre intelligent le contrôle du réseau. Ils répondent ainsi aux besoins changeant en qualité de services des services déployés sur un VN (Virtual Networking) (voir figure 1.12). L'importance du SDN a été montrée par l'article [13]. Selon les auteurs, les réseaux C-RAN et SDN sont très souvent associés au concept de virtualisation de réseau mobile (WNV, Wireless Network Virtualization). Des travaux ont aussi montré que la combinaison SDN-NFV peut offrir des avantages énormes aux réseaux sans fil, voire les réseaux mobiles 5G. Les réflexions seront plus orientées vers le problème de fonctionnement de ces différentes technologies. Selon le rapport d'Ericsson [58], le trafic mondial a augmenté de 60% entre 2015 et le premier trimestre de 2016. Pour éviter un problème d'interférence entre les différents signaux sans fils qui dégraderait la QoS, le SDWN (Software Defined Wireless Network) a été proposé comme solution d'intégration du SDN. Et cela, pour faciliter la gestion et supporter le déploiement de nouvelles applications dans l'infrastructure réseau.

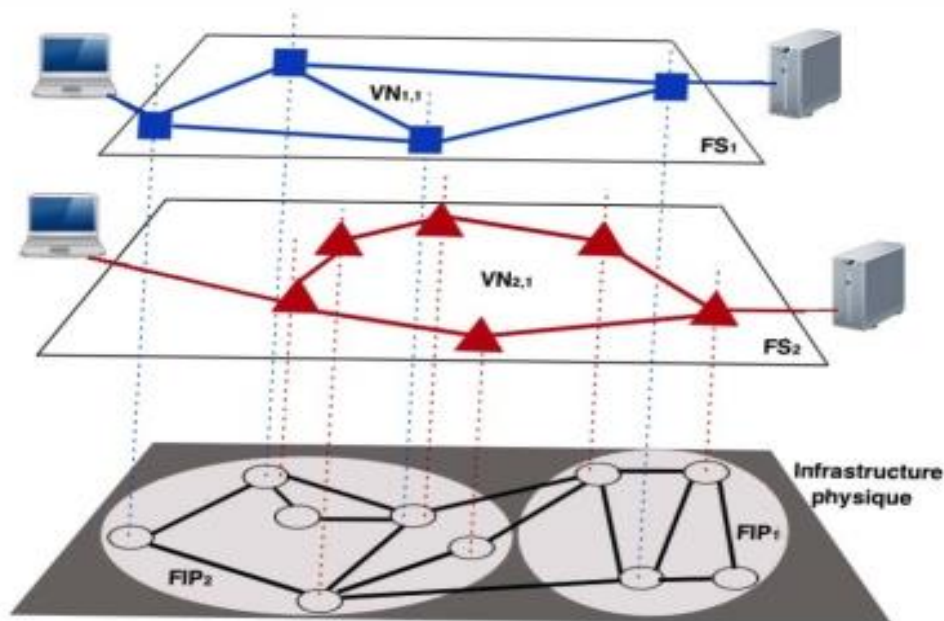


Figure 1.12 Environnement virtualisé

1.11. Conclusion

Dans ce chapitre, nous avons présenté les techniques d'optimisation d'énergie utilisées dans les réseaux ; notamment les réseaux de capteur sans fil, réseaux maillés, les réseaux 5G (en passant par 2G jusqu'à 4G) et réseaux SDN. Il ressort des analyses que ce sont les interfaces réseaux (routeur, commutateur etc.) qui sont à la base de la consommation de l'énergie. Plusieurs solutions ont été proposées comme méthodes d'optimisation de la consommation énergétique. La solution concernant les réseaux mobiles 5G, est la proposition d'une architecture H-CRAN à trois composantes. Dans la suite, le cas du SDN a été abordé. C'est une technologie basée sur la programmabilité des équipements réseaux agissant sur les fonctions réseau de façon virtuelle afin de mettre en veille ou éteindre des équipements lorsque ceux-ci ne sont pas utilisés. Cette technologie est importante parce que grâce à ses interfaces programmables, il peut être considéré comme étant la solution des réseaux du futur. Le SDN occupe une place importante dans les réseaux sans fils 5G, capteurs, maillés etc. d'où le SDWN (Software Defined Wireless Network). La virtualisation des

réseaux et le SDN sont des solutions efficaces gérant aux mieux la qualité de service tout en diminuant la consommation énergétique des réseaux mobiles 5G. Ces deux technologies permettent aux FSs d'offrir de nouveaux services aux utilisateurs avec une QoS meilleure que les services offerts par l'internet d'aujourd'hui "best effort" [15].

Le chapitre suivant est consacré entièrement à l'état de l'art des approches utilisées.

**CHAPITRE 2 : ETAT DE L'ART SUR
L'OPTIMISATION DE LA
CONSOMMATION DE L'ENERGIE DANS
LES RESEAUX SDN**

2.1. INTRODUCTION

Dans ces dernières années, les Technologies de l'Information et de la Communication (TICs) ont gagné du terrain dans notre vie de chaque jour avec les nombreux avantages qu'elles offrent sur le plan économique et humain. Ces technologies ont cependant contribué à la croissance de la consommation globale de l'énergie électrique et aussi à la forte progression des émissions de gaz à effet de serre dans le monde [22].

Notre Thèse porte sur la contribution à l'optimisation de la consommation de l'énergie dans les réseaux définis par logiciels (SDN). La motivation de notre problématique se justifie dans les articles des auteurs [5, 6, 7]. Ces auteurs ont montré que les TIC (Technologies de l'Information et de la Communication) consomment 2 à 10% de la consommation de l'énergie mondiale. Cette consommation énergétique est de 6Twh/an aux USA. Dans les réseaux mobiles, les prévisions [3], donnent 80 à 90% de communications vidéo du trafic Internet. Des auteurs estiment qu'environ 80 milliards d'objets seront connectés en 2020. Il est donc urgent que des mesures nouvelles soient prises pour faire face aux nombreux problèmes (débit, disponibilité en bande passante) à venir. Plusieurs approches technologiques pour réduire l'énergie ont été utilisées à cet effet. Nous pensons utiliser le nouveau paradigme qu'est le SDN pour apporter notre contribution. Cette architecture est nouvelle. Elle sépare le plan de données de celui de contrôle contrairement à l'architecture ancienne dont les plans de contrôle et données sont intégrés. Pour la compréhension et la maîtrise du paradigme qu'est le SDN, nous avons commencé nos travaux par une étude bibliographique. Si certains tels que les auteurs [59] et [60], nous ont permis de comprendre et définir ce nouveau paradigme, d'autres par contre ont énumérés quelques travaux qui ont traité notre problématique. Nous pouvons les classer en trois grandes classes d'approches : les approches matérielles d'abord, ensuite les approches logicielles et hybrides enfin. Les réseaux SDN concernent surtout tous les réseaux informatiques filaires, sans fils etc...

Les chapitres précédents ont permis de connaître les causes de la consommation de l'énergie dans les réseaux voire les concepts et différentes architectures. Dans la suite de ce chapitre, nous présentons les différentes approches utilisées par les différents chercheurs dans la littérature.

2.2. Différentes approches utilisées dans l'optimisation de la consommation d'énergie dans les réseaux de communication

Dans le souci de rendre le réseau éco énergétique, plusieurs recherches ont été menées afin de mettre au point des stratégies efficaces. Ces recherches s'étendent à différents niveaux du réseau tels que les routeurs, Switch, interfaces de communication et protocoles de routage etc. Cette section consacrée aux différentes approches ou techniques écoénergétiques utilisées concernent les réseaux traditionnels et la nouvelle génération des réseaux (SDN). Nous les avons classés en trois grands groupes : approche matérielle, approche logicielle et approche hybride comme déjà annoncé en introduction.

2.2.1. Approche matérielle

Notons que l'approche matérielle concerne toute modification matérielle. Elle est connue comme étant la première des solutions proposées pour minimiser la consommation énergétique dans les réseaux de communication. Nous pouvons citer par exemple les boutons marche/arrêt sur les matériels, du mode On/Off pour éteindre les équipements lorsque ceux-ci ne sont pas utiliser [61], [62] et [63].

Une approche matérielle plus simple pour réduire la consommation énergétique est décrite en [63]. C'est celle d'utiliser les interrupteurs (commutateurs), le mode On/Off permettant d'éteindre ou d'allumer les équipements réseaux à distance (par exemple le script Wakeonlan 3).

Les auteurs de [64], ont proposé deux techniques pour réduire la consommation d'énergie. Elle est basée sur l'amélioration du niveau bas des équipements réseaux les utilisant (disk spin down et DVFS). La stratégie des auteurs est de détecter les périodes d'inactivités des liens et les éteindre pour en faire une économie d'énergie sans dégradation de la qualité de service en termes de pertes et retards de paquets [62]. C'est le cas de certaines entreprises qui ont développés des équipements réseaux "green" avec les solutions des auteurs [64] tels que :

- Cisco (Energy Wise) qui réduit la consommation des équipements connectés à l'ensemble du réseau.
- IBM (Big Green) qui réduit la consommation de l'énergie dans les Data Centers.

L'ensemble des propositions de solutions des auteurs ont donné une idée à [63] qui ont proposé une approche matérielle basée sur la conception des équipements réseaux (routeurs, commutateurs...) tout en intégrant des technologies d'économie d'énergie importante et rendant souple l'utilisation des dispositifs actuels par la modification de la fréquence d'utilisation des équipements par exemple.

Limites : L'utilisation de l'approche matérielle dans la littérature a augmenté la charge du trafic dans le réseau en période de pic. Cette approche est beaucoup complexe dans son installation et dans sa maintenance puis elle est coûteuse.

Ces insuffisances ont donc conduit certains chercheurs a proposé des approches logicielles. Nous allons présenter quelques-unes.

2.2.2. Approche logicielle

Les approches logicielles concernent les conceptions algorithmiques en vue de stopper ou éteindre les liens inactifs, les mettre en veille, les faire changer de routes, réduire/augmenter la taille de la bande passante, comprimer les tables de routage sous contrainte de la qualité de service.

Les auteurs [64], [65] et [1] décrivent tous, les approches qui minimisent la consommation de l'énergie dans les réseaux. Dans [64], les auteurs ont proposé une carte réseau de façon automatique qui réponde à la demande du réseau au cas où l'équipement est à l'état "STR" (Suspend To RAM). Une autre approche illustrée dans [65] a utilisé une fonction automatique On/Off. Les auteurs ont donné deux solutions pour réduire la consommation énergétique :

- Eteindre l'équipement réseau entre la réception de deux paquets consécutifs. Cette solution est efficace en cas de bas trafic et cela dans un réseau local.
- Modifier les protocoles de routage afin de regrouper tous les trafics passant par les chemins parallèles dans un seul.

C'est le cas de citer les auteurs [1] qui ont présenté un modèle de routage géré par une commande automatique On/Off pour éteindre des liens ou routeurs en fonction de la charge du réseau. Quant aux auteurs [68], ils ont proposé une solution MILP-EWO. Cette solution est basée sur le modèle du protocole de routage OSPF. Ce modèle a pour objectif de minimiser la consommation énergétique dans les réseaux de communication tout en évitant en même temps la congestion des liens. L'heuristique utilise l'algorithme d'IGP-WO.

Les auteurs [67], [68] et [65] ont tous décrit des techniques qui minimisent l'énergie dans le réseau. Dans [67], les auteurs ont proposé un programme linéaire en nombres entiers tout en analysant de façon expérimentale et par simulation la quantité d'énergie qui peut être sauvée. Dans les travaux des auteurs [65], différentes techniques sont proposées pour économiser de l'énergie (par exemple adapter le débit) en fonction du trafic. L'impact de la technologie sur l'efficacité en énergie du routage a été mené par les auteurs [68]. C'est dans cette vision que les auteurs [69] ont proposé une approche basée sur le routage efficace en énergie par l'extinction des interfaces des routeurs sous contrainte de QoS. Une solution de réduction de l'énergie est la virtualisation du réseau au niveau traditionnel comme moderne. Nous présentons dans la dernière section cette technique et ses avantages.

2.2.2.1. Virtualisation des ressources dans le réseau

Cette technique consiste à masquer les détails des équipements c'est-à-dire les fonctions des différents services dans le réseau. Elle est un moyen de réduction efficace de la consommation de l'énergie. La virtualisation peut donc être appliquée à différentes sortes de ressources, y compris les équipements passifs (câbles, fibre optique), les dispositifs de stockage, les ressources logicielles. A titre d'exemple, en regroupant plusieurs ressources physiques pour obtenir un commutateur virtuel plus puissant avec assez de fonctionnement donne une efficacité énergétique [70]. Un autre exemple d'autres auteurs est que la virtualisation par le passé a eu un impact positif. Elle a permis de faire fonctionner simultanément plusieurs systèmes d'exploitation (Linux, Windows, Mac...) en partageant les ressources d'une seule machine. Le constat est la réduction de coût du matériel, de l'énergie et de l'empreinte carbone des centres de données de façon générale [71].

2.2.2.2. Honeyguide

Honeyguide, selon les auteurs en [72], est une topologie de migration de machines virtuelles pour l'efficacité énergétique dans les réseaux des centres de données où deux techniques sont combinées : optimisation du placement de machines virtuelles (VM) et consolidation de trafic.

Dans la littérature, il a été montré qu'un équipement en état de veille consomme 10% de la puissance en état actif. Ceci pour dire, lorsqu'un équipement ne travaille pas, il peut être mis en veille. Cette politique permet de diminuer sa consommation. Elle est appelée **état de sommeil**. Dans les travaux de [73] [64], le réveil d'un équipement à l'état de sommeil n'est pas systématique. L'équipement peut être réveillé, soit par l'arrivée d'un paquet qui se perd par la suite, soit l'interface dormant est réveillé mais n'est pas actif pour le transfert de paquets. Le paquet est mis en attente dans une mémoire tampon au niveau du routeur qui le gère voir (figure 2.1 - ci-dessous).

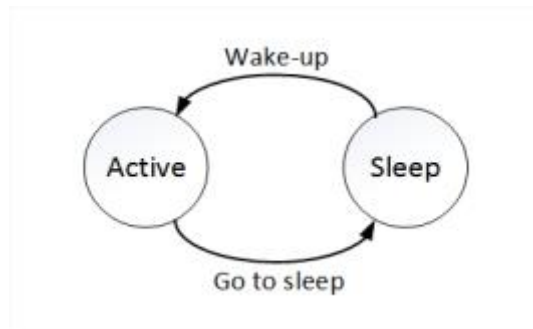


Figure 2.1 Etat sommeil et réveil d'un équipement

Limites : Les approches logicielles proposées ont minimisé la consommation de l'énergie soit par mise en veille des équipements réseaux soit par extinction des routeurs ou liens en fonction de la demande en période normale. En période de pic, il y'a eu une dégradation de la qualité de service (délai étendu, perte de paquets, variation de délai), surcharge sur le réseau, difficulté d'actualisation des tables de routage. Ceci a conduit d'autres chercheurs a proposé une approche hybride (combinaison de matérielles et logicielle).

2.2.3. Approche hybride

Plusieurs travaux des chercheurs ont traité la consommation de l'énergie dans les réseaux de communication avec l'approche hybride. Cette solution doit être évolutive, fiable, rapide, tolérante aux pannes, efficace et sécurisée. C'est le cas de citer les auteurs [1, 7, 14, 73, 74]. Les auteurs [1] ont basé leur approche d'extinction de liens ou routeurs autour du protocole de routage CPN (Cognitive Packet Network) en utilisant les boutons de commande On/Off dans les réseaux classiques. Une bonne économie d'énergie avec dégradation du QoS a été trouvée. Les auteurs ont travaillé dans un réseau de 44 routeurs et le protocole utilisé est "Cognitive Packet Network (CPN)". C'est un protocole intelligent. Le but du test-bed est de voir le comportement du réseau en termes de qualité de service (délai, perte de paquet et gigue) à chaque extinction des liens. Trois types de paquets sont utilisés pour les expériences à savoir : paquets dotés d'expériences ayant pour entête CPN, paquets

muets obtenus à la suite de l'encapsulation de son entête IP qui utilise la charge utile pour le transport des paquets et le troisième qui est un paquet généré chaque fois que celui-ci arrive à destination (ACK). Sept routeurs sont à chaque fois utilisés pour faire les expériences et 10 à 20% de réduction est observé. La fonction objectif utilisée est la suivante en plus de la topologie du réseau utilisé [1] :

$$P(n(t), t) = \sum_{i \in n(t)} P(i, t) + \sum_{i, j \in n(t)} P(k(i, j, t)) \quad (2.1)$$

Si et Seulement si $q(f(t), R(F(t), n(t), k(t), t)) \leq Q(f(t)) \quad \forall f(t) \in F(t)$

Avec

$P(n(t), t)$	<i>Consommation d'énergie dans le sous réseau</i>
$P(k(i, j, t))$	<i>Consommation des liens connaissant l'état des liens à l'instant t.</i>
q	<i>Qualité de service observée par les utilisateurs</i>
$F(t)$	<i>L'ensemble de toutes les demandes à l'instant t.</i>
$f(t) \in F(t)$	<i>Demande observée d'un utilisateur l'instant t</i>
$R(F(t))$	<i>Routage de toutes les demandes à l'instant t.</i>
$n(t) \subseteq N(t)$	<i>Le sous réseau est inclus dans le réseau à l'instant t.</i>
$Q(f(t))$	<i>Qualité de service fixée à ne pas excédée pour toutes demandes observées</i>
i, j	<i>Routeurs i et j</i>

La fonction objectif des auteurs [1] traduit que la consommation dans le sous réseau à l'instant "t" ($P(n(t), t)$) est égale à la somme de la consommation du routeur "i" à l'instant "t" pour tout "i" appartenant au sous réseau plus la somme de la consommation des états de liens formé par les routeurs "i" et "j" dans le sous réseau.

L'optimisation de la consommation de l'énergie dans le sous réseau doit respecter les sous contraintes. Les contraintes expriment simplement que, **la qualité de service (q) observée** d'une demande $f(t)$ avec le routage de l'ensemble des demandes ($R(F(t))$), dans le sous réseau ($n(t)$), en fonction de "l'état" des liens ($k(t)$) aux instants "t" ne doit pas dépasser **la qualité de service fixée ($Q(f(t))$)** à chaque demande. Nous notons

qu'une demande est prise dans un ensemble de demande. Les auteurs ont minimisé l'énergie (2.2) sous trois paramètres (**délai, perte de paquets et gigue**).

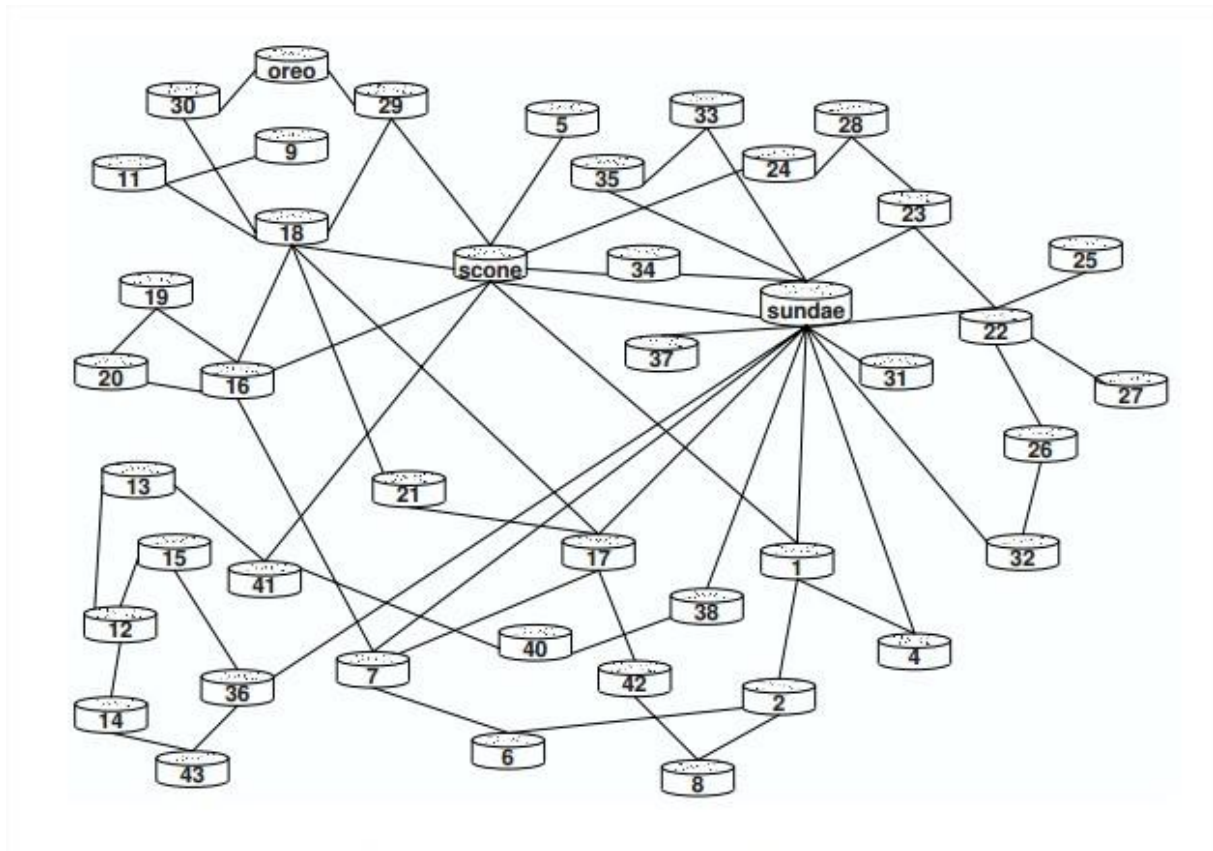


Figure 2.2 Topologie physique du réseau [1]

2.2.3.1. Explication de la Démarche expérimentale

Dans la topologie [1] de la figure 2.2, seulement que sept routeurs éteints sur 44 font une économie d'énergie, soit 15% d'extinctions possible dans les liens. Ils choisissent le routage par source qui consiste à utiliser parmi plusieurs routes source ayant une même destination, choisir une seule adresse afin de transmettre les paquets. Trois chemins sont choisis (8 à 1, 13 à 40 et 30 à 21) en fonction de leur débit. Les extinctions se font de manière progressive tout en observant la qualité de service. La dégradation du réseau survient aux instants 200s, 400s et 600s avec désactivation des routeurs 2, 41, 18 puisque le protocole a identifié ces chemins. Lorsque le chemin n'est pas pris en charge par un protocole c'est-à-dire, lorsque les routeurs sont éteints par la désactivation de tous ses liens, peu importe ce qui se passe sur ce chemin, il n'y

a pas de dégradation de la qualité de service. C'est le cas des instants 100s, 300s, 500s et 700s correspondants respectivement aux routeurs 28, 34, 16, et 26 de la figure 2.2. Nous pensons améliorer ces travaux en proposant une nouvelle technologie utilisant le protocole standard Open flow de l'ONF. Des captures de graphiques obtenus sont mentionnées ci-dessous [1].

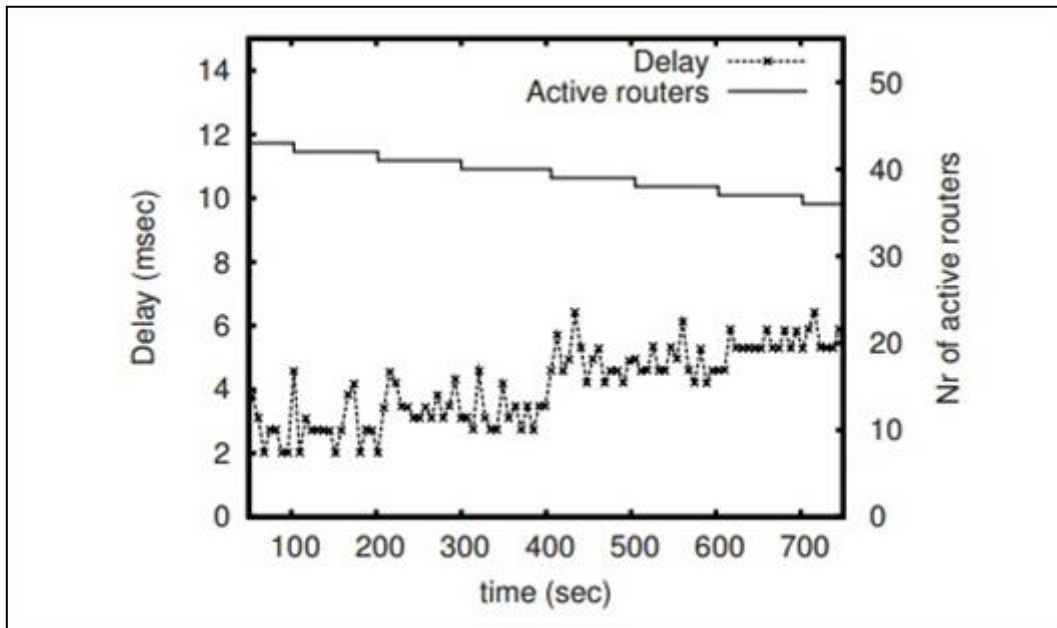


Figure 2.3 Délai – première expérience [1]

La figure 2.3 des auteurs [1], décrit le mode d'extinction progressive des routeurs par désactivation de leurs liens. Cette extinction entraîne une élévation du délai de transmission des paquets. Lorsque le routeur est éteint, les paquets mettent du temps à trouver de meilleurs chemins.

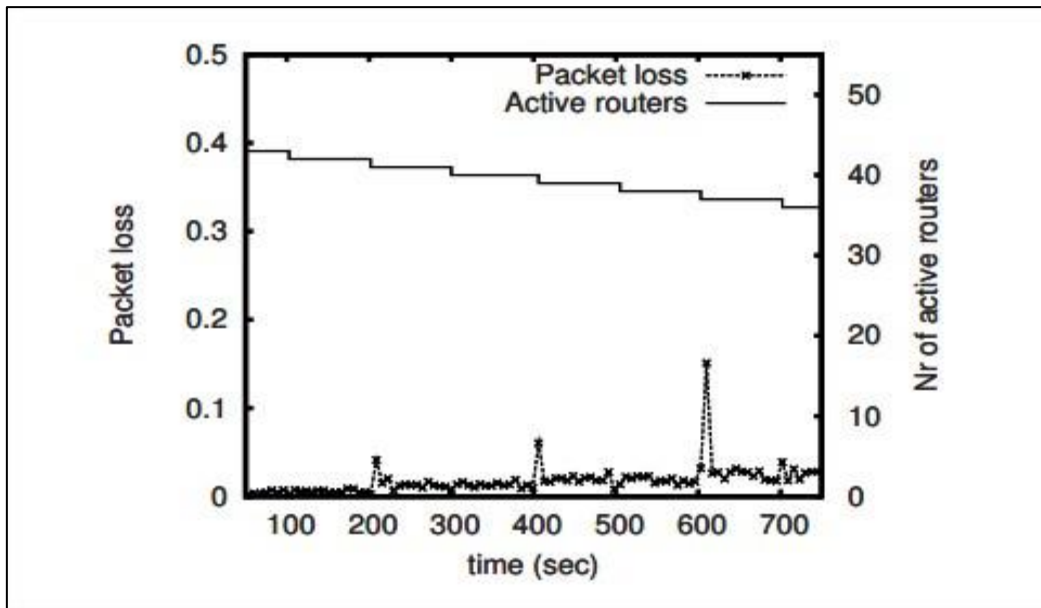


Figure 2.4 Perte de paquets – première expérience [1]

La figure 2.4 est la conséquence de la précédente c'est-à-dire, qu'au moment de l'extinction du routeur, certains paquets se trouvant dans la file d'attente sont automatiquement perdus. D'autres aussi, se perdent par le phénomène de congestion qui survient sur les chemins courts pendant l'extinction.

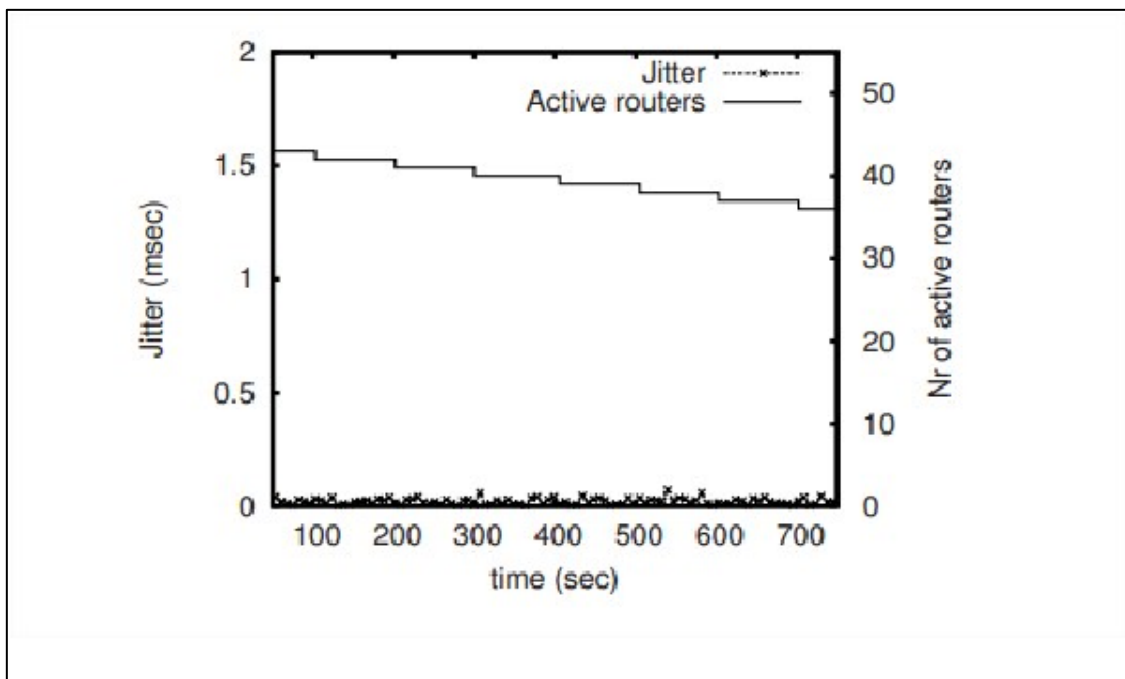


Figure 2.5 Gigue – première expérience [1]

Cette partie (figure 2.5) décrit l'évolution de la gigue. La gigue est la variation de la latence. Ici, la gigue suit le rythme du délai.

Dans les travaux intitulé << **Optimisation de la consommation de l'énergie dans les réseaux d'accès radio 5G sur la base d'un algorithme de commande d'admission d'appel**>>, Zamble et al. ont proposé un algorithme de contrôle de ressources (appels etc.) à chaque demande sous contrainte de la qualité de service et une nouvelle architecture afin de permettre aux utilisateurs d'échanger des informations tout en évitant les interférences [13]. Ces auteurs [13] ont minimisé la consommation de l'énergie en ayant déterminé un seuil de débits qui dégraderait la qualité de service de façon algorithmique. La stratégie est de mettre en veille les équipements RRH (équipement de transmission) sous-utilisé. Elle s'appuie sur ceux de Yongyu et al. [75], qui ont dans leur recherche, proposé des algorithmes minimisant à la fois la puissance totale et le nombre de commutations des modes actifs / veille dans un réseau H-CRAN avec garantie de la qualité de service.

Cette architecture doit tenir compte des challenges à venir et les mécanismes à mettre en place pour la gestion optimale du spectre. La station de base sera décomposée en RRH (Remote Radio Head) et BBU (Base Band Unit) regroupées dans un BBU pool relié par un lien appelé fronthaul, c'est-à-dire une architecture à trois composantes. C'est donc une C-RAN possédant une architecture centralisée dans laquelle la partie responsable du traitement bande de base est séparée du traitement radio fréquence [3, 39, 42, 76, 77]. Cette séparation va permettre à chaque opérateur d'envisager le mode de fonctionnement de son système (il fournit ses propres applications) afin de faire face aux préoccupations des clients et utilisateurs.

Cette gestion a eu pour avantages la réduction de la puissance du réseau, une utilisation efficace des ressources et une réduction des coûts d'exploitation et maintenance. La combinaison des réseaux C-RAN et hétérogènes (HetNets: Heterogeneous Network), selon les experts est une solution pour les réseaux mobiles futurs, pour la réduction de la consommation de l'énergie dans les réseaux [78, 79].

Sachant que les actifs réseaux sont énergétivores, d'autres chercheurs ont proposé des solutions intelligentes voire automatique pour agir sur les équipements actifs à distance et de façon dynamique pour réduire la consommation de l'énergie sans dégradation de la qualité de service. C'est un contrôleur centralisé ayant une vue globale du réseau qui est utilisé où le plan de contrôle et celui de données sont séparés. Les équipements éteints sont ceux qui ne sont pas utilisés dans le réseau. Cette architecture s'appelle SDN (Software Defined Network) ou réseau défini par logiciel. Giroire et al. [7] et Havet et al. [14], ont utilisé le SDN pour réduire l'énergie dans les réseaux. Cette nouvelle architecture apparaît comme la solution de l'augmentation des données exponentielles des défis qui attendent la cinquième génération du réseau mobile (5G).

D'autres techniques tels que l'adaptation du réseau à la charge du trafic, l'optimisation du placement des règles des flux et la compression de la mémoire TCAM ont été utilisés comme approche hybride dans l'économie de l'énergie.

2.2.3.2. Adaptation dynamique à la charge du trafic

Jin et al. ont proposé un modèle de réduction énergétique hybride dans les réseaux de communication en combinant des approches "Low Power Idle" (LPI) et Adaptive Link Rate "ALR) [74]. La consommation de l'énergie dans les réseaux a motivé la thèse de Wael Zouaoui, 2016 "*Synthèse de commande pour des réseaux de communication énergétiquement performants*" soutenu le 15 janvier 2016 [22]. Ces auteurs ont rappelé les différentes approches utilisées et ont proposé une nouvelle solution hybride composée par une application originale et alternative pour réguler le flux de sortie du nœud contrôlé, couplé avec l'approche ALR pour minimiser l'énergie sous contrainte de bonne qualité de service. Une proposition de solution de mise en veille des liens et routeurs est décrite en [80]. Une réduction de 35% dans les nœuds et 25% dans les liens a été obtenue.

Les équipements réseaux ont fonctionné avec un système intégré (plan de données et de contrôle). Ce qui rend l'évolution du déploiement de nouveaux protocoles

difficiles. L'apparition du paradigme SDN semble intéresser les chercheurs. Il a une souplesse dans le développement et simulation de nouveaux protocoles et politiques dans les réseaux.

2.2.3.3. Optimisation du placement des règles de flux

Nous citons les travaux de [2] qui ont utilisé le protocole de routage SPRING tout en présentant l'architecture logicielle STREETE (Segment Routing based Energy Efficient Traffic Engineering) qui adapte dynamiquement le nombre de liens des routeurs allumés en fonction de la charge du réseau avec le SDN. L'optimisation du placement des règles de flux a été aussi utilisé en [7, 81, 82]. Ils ont proposé un programme linéaire en nombres entiers et des heuristiques algorithmiques pour minimiser l'énergie avec la technologie SDN. Ils ont défini des règles afin de rediriger le trafic et éteindre les liens non utilisés sous contrainte de capacité de liens dans le réseau et l'espace des règles. Cette technique est beaucoup utilisée dans les réseaux SDN comme le mentionne le rapport de Zemmouri et al. Intitulé "**Modèle de routage éco énergétique dans les réseaux définis par logiciel**" [83]. Dans cette approche, il s'agit de savoir comment et où placer les règles de flux dans la mémoire TCAM. TCAM est une mémoire gourmande en consommation énergétique et coûteuse ayant une capacité de stockage limitée. L'optimisation des règles de flux réduit l'espace mémoire TCAM utilisée, donnant ainsi une économie d'énergie. La solution de cette technique n'est pas optimale mais approximatif. Dans le souci d'apporter une amélioration aux travaux préliminaires de [7], les auteurs [14] ont étendu leur idée par une approche de compression de table de routage.

2.2.3.4. Compression des données de la table de routage SDN

Plusieurs chercheurs ont utilisé cette approche comme modèle d'économie d'énergie [14] et [84]. Dans [14], les auteurs ont étendu les travaux de la revue [7] par la compression de la table de routage utilisant une mémoire TCAM. Cette solution minimise le besoin de mémoire des informations stockées dans les commutateurs. La

mémoire TCAM a une grande vitesse et elle fait les recherches dans un seul cycle d'horloge. La solution proposée par les chercheurs, c'est la compression des données se trouvant dans la mémoire (règles et contenu). Il en est de même dans l'article [84]. Les auteurs [85] ont quant à eux proposé un algorithme de routage nommé MINNIE se fondant sur la technique de compression de règles pour réduire la taille des tables et celle de la consommation de l'énergie. Car la compression réduit le délai (temps de contact avec le contrôleur et le temps d'installation des règles pour le flot). Les auteurs ont montré qu'avec 1000 règles par table, MINNIE supporte aux alentours d'un million de flots avec des temps de calcul négligeables. La qualité de service n'est pas dégradée en termes de perte de paquets et de délai avec MINNIE et le taux de compression est entre 70 et 96%. Ce qui traduit qu'elle est une approche efficace (voir table 2.2). Ce tableau fait la comparaison du comportement du simulateur MINNIE sur différentes topologies des centres de données avec environ 1000 règles par tables. Comme, nous pouvons le voir dans le tableau, ces topologies, quatre au total (Flat tree, VL2, BCube et DCell) sont divisées en deux groupes. Le premier formé, est constitué de flat tree et de VL2. Le second formé, est composé de BCube et DCell. La comparaison a montré que sans absence compression, seuls 128 serveurs qui peuvent être déployés. En cas de compression, 256 serveurs sont déployables. Et l'utilisation complète de MINNIE, donne jusqu'à 1458 serveurs avec une limite de 1000 règles pouvant être installées.

Table 2.2 Résultats de MINNIE sur différentes topologies avec environ 1000 serveurs

Topologie	#de serveurs	# de commutateurs	# de liens	Degré moyen	# de flots par commutateurs			# de règles avec comp.			Moyen Comp. Ratio	Temps de calcul en moyenne (ms)	
					Total	Max	Moyen	Total	Max	Moyen		Chemin	Comp.
Group 1													
4- Flat tree (64)	1024	20	1056	54.4	917504	454244	216268	8923	999	446	~ 99.60	0.17	13
8- Flat tree (8)	1024	80	1280	19.2	1015808	649044	61030	25853	999	323	~ 99.61	0.21	7
16- Flat tree (1)	1024	320	3072	16	1040384	630998	15897	97173	999	303	~ 98.42	0.30	5
VL2 (16,16,14)	896	88	384	16	790272	261266	42906	59237	1000	673	~ 97.90	0.15	4
VL2 (8,8,64)	1024	28	612	~ 41.1	983040	423752	161499	22394	1000	799	~ 99.45	0.19	11
VL2 (16,16,16)	1024	88	1152	~ 17,5	1032192	276575	56040	57078	1000	648	~ 98.39	0.18	4
Group 2													
Dcell (32,1)	1056	33	1584	~ 2.91	1114080	63787	4893	123655	1000	113	~ 97.23	0.09	2
Dcell (5,2)	930	186	1860	~ 3.33	863970	11995	5716	717018	994	642	~ 87.84	0.19	2
BCube 32,1)	1024	64	2048	~ 3.77	1047552	37738	3734	358204	999	329	~ 86.04	0.19	2
BCube (10,2)	1000	300	3000	~ 4.62	999000	10683	4153	849316	998	653	~ 80.85	0.25	2
BCube (6,3)	1296	864	5184	~ 4.8	1678320	7852	5184	1795400	991	831	~ 83.18	0.49	4

2.2.3.5. Autres types de compression de données sur TCAM

- **Rectilinear**

D'après les travaux des auteurs [83], "Rectilinear" est définie comme une technique qui exploite les fonctionnalités de l'architecture SDN. La compression réduit la taille des bits pour stocker l'essentielle des informations afin de classer les paquets en flux. Un identifiant est donné à chaque demande pour identifier de façon unique les paquets dans le flux correspondant [86].

La modification des entêtes de paquets se situe au niveau des commutateurs d'acheminement. Cela permet de transporter l'identifiant de flux qui peut être utilisé par d'autres équipements sur le chemin en vue de faire le classement des paquets. Les auteurs ont montré que la compression réduit la consommation d'énergie de la mémoire TCAM de 80% en moyenne.

- **Bit Weaving**

Cette technique comme l'indique son nom, utilise le schéma de compression sans préfixe. En citant les travaux des auteurs, la technique "bit weaving" [87] consiste en l'observation des entrées TCAM ayant la même décision. Les prédicats qui diffèrent

d'un seul bit, peuvent être fusionnés en une entrée en remplaçant le bit par *. Deux types d'approches sont utilisées par "bit weaving" : l'échange de bits et la fusion de bits [87].

Cette technique présente un avantage dans la rapidité d'exécution et l'efficacité plus ou moins complémentaire à d'autres méthodes d'optimisation TCAM comme routine de pré/post-traitement.

Nous notons que la désactivation des ports ou liens des routeurs est une forme de réduction des interfaces des routeurs. Mais comment désactiver des ports ou liens sans dégrader la QoS ? Ce problème est NP-complet. Selon une étude menée par Brandon et al, le temps nécessaire pour allumer/éteindre les liens et les équipements réseaux varie de 30 secondes pour le commutateur " Quanta" à environ 3 minutes pour celui de "HP" [48]. Par contre l'activation et la désactivation des ports prennent 1 à 3 secondes et remplit la table de flux en de moins de 5 secondes. Ceci dit, il convient de dire que la désactivation des ports ou liens des routeurs est une solution éco énergétique. Une ingénierie doit cependant être développée pour éviter les délais longs de transmission de paquets dans le réseau. Plus le délai de transmission est long, plus il y'a consommation d'énergie dans le réseau.

Limites : Comme inconvénient, une dégradation de la qualité de service due à une surcharge dans le réseau en période de pic, délai étendu, perte de paquets, contraintes limitées dans les fonctions objectifs définis dans certains travaux des auteurs. La problématique de la mémoire utilisée dans les dispositifs SDN est préoccupante étant donné que leur taille est limitée.

2.3. Synthèse bibliographique pour les réseaux classiques et nouvelles générations (SDN)

Nous exposons dans cette section, une synthèse des approches proposées pour réduire la consommation de l'énergie dans les réseaux de communication (réseaux

classiques et SDN). Des atouts et limites de chacune de ces approches ont été dégagés.

Nous remarquons que l'ensemble des approches minimise la consommation de l'énergie pendant les périodes où le trafic est faible, mais en période de pic (période dense), la qualité de service dans le réseau est dégradée.

Malgré les bons résultats, l'approche matérielle demeure très coûteuse et complexe dans son installation et sa maintenance face à une topologie de réseau étendue. Cette complexité n'encourage surtout pas les fournisseurs des équipements réseaux à la proposer, sur le plan commercial.

Pour les autres approches c'est-à-dire logicielles et hybrides, présentent entre autres, des résultats satisfaisants comme solution pour réduire l'énergie dynamiquement.

L'utilisation d'autres techniques dans le réseau, et sur les routeurs par des chercheurs, est due à la complexité qu'il y a dans l'approche matérielle. Ces méthodes sont entre autres : l'extinction des liens, la mise en veille des liens à l'état d'inactivité du réseau. Les résultats de cette approche réduisent efficacement l'énergie, et présente des insuffisances en période de pic (activité dense). Cette limite est liée aux délais de transmission des paquets longs, la perte de paquets et la gigue due à un goulot d'étranglement dans le réseau. Une autre limite est de déterminer le moment idéal pour appliquer l'extinction ou la mise en veille des équipements sans dégrader la QoS.

La technique d'adaptation dynamique à la charge du trafic, est un moyen efficace pour les autres solutions. Elle est logicielle et nécessite des équipements informatiques adéquats afin de rendre possible son exploitation. L'avantage de cette technique, est qu'il peut être couplé avec d'autres solutions de gestion énergétique tout en respectant une certaine condition de satisfaction mutuelle.

Un autre point concernant les limites, est lié à la dégradation de la qualité de service en termes de délais de transmission. Les paquets mettent du temps dans le "buffer" afin d'adapter le taux de transmission dans les liens.

Plusieurs auteurs ont quant à eux, utilisés la technique de l'optimisation de placement de règles de flux. Cette technique consiste à, définir des règles en vue de ré-diriger le trafic et éteindre les liens non utilisés sous condition de capacité de liens et d'espace des règles. Mais la limite de cette technique, est de déterminer comment et où placer les règles de flux dans la mémoire TCAM. Cette mémoire est coûteuse et gourmande en consommation de l'énergie, ayant une capacité de stockage limitée.

Un des points importants est que, l'optimisation des règles de flux réduit l'espace mémoire TCAM utilisée pour réduire l'énergie. D'autres techniques dans la littérature, telles que la compression de la table de routage SDN, "Rectinear", Bit Weaving etc. ont été utilisé. Les résultats de ces techniques pour la plupart, sont obtenus par approximation utilisant la Programmation Linéaire en Nombres Entiers (PLNE) et des heuristiques. L'ensemble de ces techniques sont regroupées dans les solutions hybrides et sont utilisées dans les réseaux de nouvelles générations (SDN). Parmi toutes les solutions proposées, le SDN offre plus d'avantages en termes de gestion de trafic, routage et qualité de service etc. C'est un nouveau paradigme dans lequel le plan de contrôle est séparé de celui de données. Ce découplage plan de contrôle, plan de données rend souple et flexible la gestion du réseau. La partie contrôle est centralisée pour agir sur le plan de données c'est-à-dire sur les équipements réseaux à distance de façon automatique et dynamiquement via le protocole open flow pour éteindre ou mettre en veille les liens non utilisés. C'est un axe nouveau dans lequel plusieurs chercheurs travaillent et proposent leurs solutions. Dans la suite, nous expliquons notre proposition de contribution avant de conclure.

2.4. Conclusion

Avant de conclure ce chapitre, nous remarquons que toutes les approches réduisent la consommation de l'énergie avec plus ou moins de dégradation dans les réseaux classiques et SDN. Plusieurs travaux de recherches ont traité ce type de problème et

plusieurs solutions en ont été proposées afin de réduire la consommation de l'énergie avec une qualité de service relative.

Des auteurs ont montré que ce n'est pas la charge du réseau qui fait consommer mais c'est le nombre d'équipements. Une fois que ces équipements connectés consomment de l'énergie [67], [69] et [88]. Les auteurs en [1], [2] et [22] ont utilisé l'approche hybride pour proposer leur solution qui consiste à éteindre des liens lorsque ceux-ci ne sont pas utilisés. En [2], une solution logicielle STREETE (Segment Routing based Energy Efficient Traffic Engineering) a été proposée. Cette solution (STREETE) adapte le nombre de liens du réseau qui est allumé en fonction de la demande et dynamiquement dans les réseaux cœurs.

Le chapitre suivant présentera ainsi, le détail de notre solution proposée.

**CHAPITRE 3 : CONTRIBUTION A
L'OPTIMISATION DE LA
CONSOMMATION DE L'ENERGIE DANS
LES RESEAUX SDN**

3.1. INTRODUCTION

En général, il existe deux types d'équipements dans les réseaux de communication. Ce sont les équipements actifs tels que les routeurs, les commutateurs, etc (dispositifs de transmission des informations) et les équipements passifs tels que les câbles, les fibres optiques, etc. (support d'interconnexion des équipements). Parmi les deux types d'équipements, seuls les actifs sont énergivores.

L'évolution de ces réseaux entraîne une augmentation de ces équipements en quantité et en performance. L'augmentation de ceux-ci et de leur performance accroissent la consommation de l'énergie électrique dont ils ont besoin pour leur fonctionnement.

Une étude réalisée en 2009 [5] montre que les Technologies de l'Information et de Télécommunication (TIC) consomment à elles seules 2 à 10% de la consommation mondiale d'énergie. Les hubs, Switch et routeurs consomment 6 Twh/an aux USA [6]. Dès lors la recherche d'un modèle mathématique pour la consommation de l'énergie dans les réseaux de communication devient une préoccupation réelle pour les entreprises du moment. D'autres travaux comme ceux des auteurs [89] ont donné une estimation de la répartition de la puissance consommée des équipements des TIC dans le monde (**voir table 3.1**).

Table 3.1 Estimation de la répartition de la puissance consommée des équipements des TIC dans le monde [89]

Équipement	Puissance de pointe
Data Center	29 GW
PCs	30 GW
Équipements réseau	25 GW
TVs	44 GW
Autres équipements	40 GW
Total	168 GW

Selon les auteurs, l'ensemble de la consommation énergétique des équipements cités représentent plus de 8% de la consommation globale de l'électricité mondiale. Les auteurs montrent de même que d'environ 160 Gw en 2008, cette consommation atteindra 470 Gw d'ici 2020 (figure 3.1). Nous constatons qu'il est plus urgent de proposer des solutions dans les recherches scientifiques comme proposition de réduction efficace de l'énergie. Avec cette estimation, nous concluons que le réseau classique peinera à satisfaire les attentes des utilisateurs.

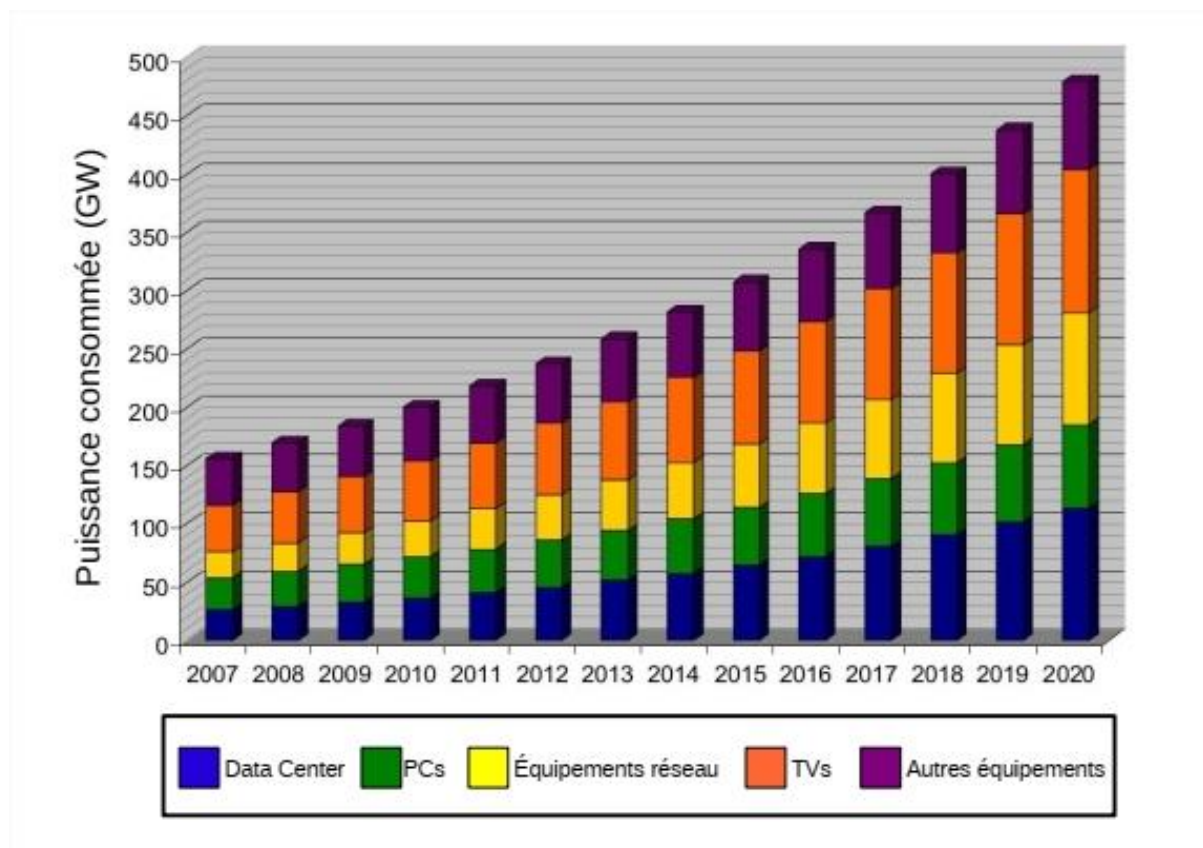


Figure 3.1 Consommation énergétique des réseaux de communication [89]

Dans les réseaux classiques, lorsqu'un paquet arrive sur un port d'un commutateur ou d'un routeur, celui-ci applique les règles de routage ou de commutation qui sont inscrits dans un système d'exploitation. Généralement, tous les paquets qui ont la même destination suivent le même chemin. Dans les modèles haut de gamme, les matériels sont capables de reconnaître le type d'application et de lui appliquer les

règles spécifiques. Mais cette programmation est rigide. Elle ne peut être changée que manuellement, par l'administrateur, ce qui prend évidemment du temps [93].

L'avènement des technologies de réseaux SDN (**Software Defined Network**) apparaît comme une bonne alternative pour agir sur les équipements à distance et de manière dynamique en vue de contribuer à la réduction de la consommation de l'énergie.

Dans la technologie SDN, c'est un contrôleur centralisé qui va se charger de faire router les paquets dans le réseau via un protocole SDN (openflow) par la programmabilité en injectant des règles de routage fournie par la couche application (**Figure 3.2**). Dans cette Thèse, nous utilisons ce paradigme pour agir sur les équipements réseaux en activant ou désactivant les ports des routeurs lorsque ceux-ci ne travaillent pas. Une nouvelle stratégie a été développée et prend en compte les périodes de pic (trafic dense) et normale (trafic bas) minimisant l'énergie tout en évitant la congestion. Les auteurs [1] ont dans leur approche minimisé l'énergie avec des délais élevés entraînant des pertes de paquets.

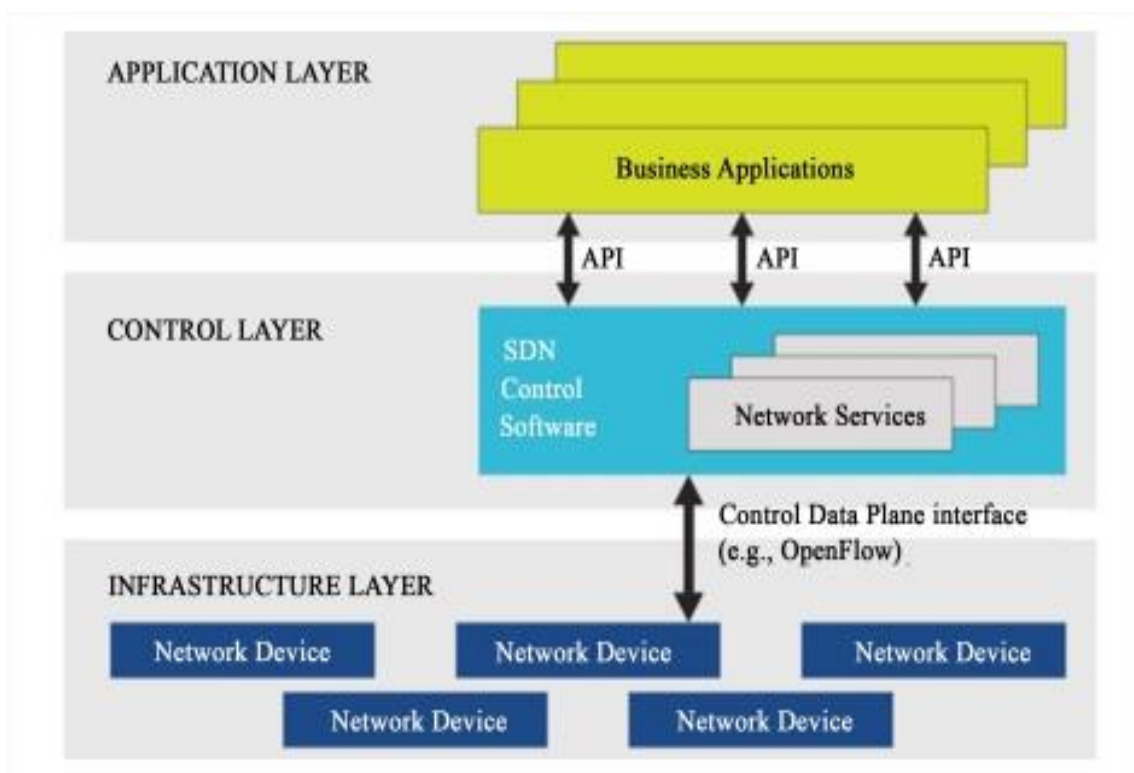


Figure 3.2 Architecture du SDN [89]

Notre modèle est basé sur celui des travaux de Gelenbe et al. [1]. Nous avons mis en place un modèle de minimisation de l'énergie basé sur une nouvelle stratégie. Notre démarche va s'articuler autour des points suivants :

- 1- Recherche de modèle d'économie d'énergie dans les réseaux de communication et SDN
- 2- Recherche d'approche de résolution du modèle
- 3- Critère et choix de la méthode de résolution
- 4- Simulation du modèle via OMNET++
- 5- Discussion et validation

3.2. Recherche d'un modèle d'économie d'énergie dans les réseaux SDN

La consommation d'énergie a été traitée par certains chercheurs [1, 7, 14, 67 et 88]. Dans [1], les auteurs ont minimisé la consommation de l'énergie sous trois contraintes (délai, perte de paquet et gigue), l'utilisation de la technologie on/off dans les réseaux câblés classiques, a réduit la consommation d'énergie dans les réseaux. Une économie de 10 à 20% a été obtenue. Mais à chaque extinction, une variation de délai entraînait des pertes de paquets en cas de trafic important. Les auteurs de [7, 14] ont minimisé l'énergie dans les réseaux SDN en utilisant l'approche de « compression » des données de la table de routage. Selon les auteurs, les dispositifs capables d'implémenter les règles SDN utilisent MTAC (Mémoire Ternaire Adressable par Contenu). Cette mémoire dans laquelle sont enregistrées les règles, est chère et gourmande en énergie. La compression des données de la table de routage va créer de l'espace mémoire afin d'augmenter le nombre chemin par la définition d'autres règles. La compression maximise l'espace de règle en augmentant le nombre de chemins. La consommation d'un lien à haute capacité et celle d'un lien non sollicité étant faibles, certains flux peuvent être redirigés vers d'autres liens. Comme solution de retard, les auteurs ont défini la règle par défaut pour le transfert

de paquets vers le port par défaut sans contacter le contrôleur SDN. Une économie d'énergie a été observée, mais avec une dégradation de la qualité de service en termes d'encombrement du réseau. En ce qui concerne les travaux de [69] et [88], les auteurs ont montré que c'est le nombre de ports des routeurs sur le réseau qui consomme. Notre approche va consister à désactiver les interfaces des routeurs lorsqu'ils ne sont pas utilisés. Nous orientons notre travail de cette manière tout en proposant un modèle mathématique de minimisation d'énergie dans le réseau SDN qui satisfasse à la « QoS » en termes de délai de transmission et de perte de paquets.

3.2.1. Hypothèses de travail

- ✓ Le futur modèle mathématique proposé à la section suivante et qui utilise l'algorithme de kruskal pour réduire les ports, minimise la consommation de l'énergie tout en préservant une bonne qualité de service.
- ✓ L'extinction des ports pourrait être une solution d'économie d'énergie dans les réseaux SDN.
- ✓ Chaque port a le même taux de consommation d'énergie.
- ✓ On peut avoir plusieurs interfaces sur un seul port. Chaque port peut donc gérer plusieurs services afin de rediriger le trafic.
- ✓ Utiliser l'exemple de Switch Cisco Catalyst 3560E ayant une puissance de 0.32w par port.

3.2.2. Modélisation du problème

Cette section consistera à faire la description de notre modèle mathématique. Nous formulons notre problème sous la forme d'un programme linéaire en nombres entiers contribuant à l'optimisation de la consommation d'énergie dans les réseaux SDN. Nous définissons N comme étant le nom du réseau et $n(t)$ celui du sous-réseau de N avec ($n(t) \subseteq N$) à un instant t .

Notre approche est dynamique basée sur deux hypothèses :

h1) L'activation et la désactivation des ports des routeurs à l'instant t et on note l'évolution de la consommation d'énergie.

h2) Nous supposons qu'un routeur ne peut pas être éteint et doit donc rester allumer. Une fois allumé, à chaque instant donné t , un lien est établi entre un routeur i et un routeur j . Il s'agira donc de maximiser un nombre de ports à désactiver pour réduire la consommation de l'énergie sous contrainte de délai, perte de paquets et gigue. Dans notre approche, nous activons ou désactivons les ports de routeur en utilisant une approche logicielle.

Le nombre de ports à désactiver $\ell(t)$ est défini comme suit :

$$\begin{aligned} \ell: [0, +\infty[&\rightarrow \mathbb{N} \\ t &\mapsto \ell(t) \end{aligned} \quad (3.1)$$

La fonction ℓ caractérise le nombre de ports désactivés à chaque instant " t ".

Nous considérons λ_{max} le nombre total de ports sur un routeur. Ce paramètre λ_{max} est fixe et dépend de chaque type de routeurs. Si par exemple, le nombre de ports sur un routeur est de huit (08), combien de ports faudra-t-il désactiver pour faire une bonne économie d'énergie sans toutefois éteindre l'équipement ?

Soit P , la consommation de l'énergie dans les sous réseaux. Nous exprimons le modèle de notre approche mathématique à partir de la fonction objectif de la solution du problème c'est-à-dire (h1) et (h2). Ce qui revient à :

$$P(n(t), t) = \sum_{i \in n(t)} P(i, t, \ell(t)) + \sum_{i, j \in n(t)} P(k(i, j, t)) \quad (3.2)$$

Un lien est la jonction entre deux interfaces de routeurs. $i, j \in \mathbb{N}$ ayant une capacité de charge maximale de $C(i, j)$ trafic de paquets par seconde. Les liens peuvent être dans l'état suivant **0** ou **1**. Soit k la variable d'état du lien : **0** état où le port est désactivé et **1** état où le port est activé. La fonction objectif à minimiser que nous rappelons dans l'objectif de définir ses différents paramètres. Cette fonction traduit que la consommation d'énergie du sous réseau $(P(n(t), t))$ est égale à la somme de la

consommation du routeur "i" dont $\ell(t)$ ports ont été désactivés à l'instant t plus la somme de la consommation d'énergie dans les liens.

$$P(n(t), t) = \sum_{i \in n(t)} P(i, t, \ell(t)) + \sum_{i, j \in n(t)} P(k(i, j, t)) \quad (3.2)$$

$P(i, t, \ell(t))$ est la consommation d'énergie du routeur i dont $\ell(t)$ ports ont été désactivés à l'instant t et $P(k(i, j, t))$ représente l'énergie du lien i, j au temps t. $P(i, t, \ell(t))$ est la fonction à exprimer.

Soit $g_i(t, \ell(t))$: le gain d'énergie du routeur i tel que $\max_t \ell(t) < \lambda_{max}$, avec

$g_i(t, \ell(t)) = a_i \ell(t) + b_i$ où $a > 0$. La fonction $g_i(t, \ell(t))$, obéit au mode de désactivation des ports du routeur "i" à chaque instant t. Elle exprime notre stratégie d'économie d'énergie. La fonction "gain d'énergie" est une fonction linéaire et strictement croissante en fonction du nombre de ports éteints.

$$\text{Si } \max_t \ell(t) < \lambda_{max} \text{ alors } P_i(t, \ell(t)) = C_i - (a_i \ell(t) + b_i) \quad (3.3)$$

Sinon $P_i(t, \ell(t)) = 0$ où C_i est la consommation totale du routeur i et

$P(i, t, \ell(t)) = P_i(t, \ell(t))$. Nous déduisons que si tous les ports sont désactivés, $g_i(t, \ell(t)) \approx C_i$.

La fonction (3.3) est l'expression finale de notre fonction "consommation de l'énergie" du routeur.

Nous montrons ainsi, que notre fonction est linéaire en $\ell(t)$. Elle est de la forme :

$$P(n(t), t) = \sum_{i \in n(t)} (C_i - (a_i \ell(t) + b_i)) + \sum_{i, j \in n(t)} P(k(i, j, t)) \quad (3.4)$$

$$\min(P(n(t), t)) = \min \left(\sum_{i \in n(t)} (C_i - (a_i \ell(t) + b_i)) + \sum_{i, j \in n(t)} P(k(i, j, t)) \right) \quad (3.5)$$

Avec $C_i =$ Consommation du Châssis (C_{ch}) + Consommation des ports (C_p) allumés.

D'où : $C_i = C_{ch} + C_p$

$$\min(P(n(t), t)) = \min \left(\sum_{i \in n(t)} ((C_{ch} + C_p) - (a_i \ell(t) + b_i)) + \sum_{i, j \in n(t)} P(k(i, j, t)) \right) \quad (3.6)$$

S/C de :

$$q(f(t), R(F(t), n(t), k(t), t)) \leq Q(f(t)), \forall f(t) \in F(t) [5]$$

$$\text{et } \sum_k f(t) \leq C_{i,j}, \forall f(t) \in F(t)$$

$$k \in \{0, 1\}$$

Table 3.2. Résumé des notations

$P(n(t), t)$	Consommation d'énergie dans le sous réseau
λ_{max}	Nombre total des ports sur un routeur
C_i	Energie globale du routeur i
$a_i \ell(t) + b_i$	Gain d'énergie du routeur i obtenu en fonction des ports désactivés.
$P(k(i, j, t))$	Consommation des ports dans le sous réseau à l'instant t connaissant l'état des ports.
q	Qualité de service observée par les utilisateurs
$F(t)$	L'ensemble de toutes les demandes à l'instant t .
$f_{s,p}(t) \in F(t)$	Demande observée d'un utilisateur d'un point s à p à l'instant t
$R(F(t))$	Routing de toutes les demandes à l'instant t .
$n(t) \subseteq N(t)$	Le sous réseau est inclus dans le réseau à l'instant t .
$Q(f(t))$	Qualité de service fixée à ne pas excédée pour toutes demandes observées

Nous avons obtenu un modèle mathématique qui minimise l'énergie, donc une méthode de résolution s'avère nécessaire.

3.3. Recherche d'approche de résolution du modèle

Dans cette partie, nous proposons une méthode de résolution de notre modèle mathématique.

Dans la littérature, plusieurs méthodes de résolution de problèmes d'optimisation ont été utilisées. Parmi lesquels : le Programme Linéaire en Nombres Entiers (PLNE), les heuristiques algorithmiques, le graphe de flot etc. Après avoir donné ou expliqué chacune des méthodes, nous dégagerons notre choix en fonction de la complexité du problème. Le problème d'optimisation dans les réseaux est NP-difficile à cause de la taille du réseau, la définition des règles de routage etc. Il n'existe donc pas de solution mais des propositions de solutions proches de la réalité.

3.3.1. La Programmation Linéaire en Nombres Entiers (PLNE)

Un programme linéaire en nombres entiers selon les auteurs dans la littérature, est basé sur trois (03) critères :

1. Les variables : ce sont les paramètres dont les valeurs optimisent la fonction objectif tout en respectant les contraintes du problème. Les variables doivent être entières.
2. La définition des contraintes. Elles définissent les conditions que doivent vérifier les variables. Les contraintes peuvent être des équations ou inéquations.
3. La fonction objectif : elle concerne la fonction à optimiser.

Les équations ci-dessous constituent un exemple de programme Linéaire en Nombres Entiers (PLNE) [7] :

$$\min \sum_{u,v \in E} X_{uv} \quad (3.7)$$

$$S/C \sum_{v \in N(u)} (f_{vu}^{st} + g_{vu}^{st} - g_{uv}^{st} - f_{uv}^{st}) = \begin{cases} -1 & \text{if } u = s, \\ 1 & \text{if } u = t, \\ 0 & \text{else} \end{cases} \quad \forall u \in V, (s, t) \in \mathcal{D} \quad (3.8)$$

- \mathcal{D} : Ensemble de demande
- X_{uv} : Variable binaire indiquant si le lien (u, v) est actif ou pas.
- $N(u)$: Ensemble des routeurs adjacents de u dans le graphe G .
- f_{uv}^{st} : Le flux (dont la source est "s" et la destination est "t") qui est routé par des règles normales.
- g_{uv}^{st} : Le flux (dont "s" est la source et "t" la destination), routé par les règles par défaut.

La fonction objectif (3.7), minimise la consommation de l'énergie dans les liens actifs. Les contraintes des flux (3.8) indiquent que la somme des flux entrant et sortant d'un routeur est égale, sauf les nœuds sources et destination (Règle de cohérence ou loi de Kirchhoff). Il est à noter qu'un flux normal entrant dans un routeur peut devenir un flux par défaut sur le lien sortant et vice - versa.

3.3.2. Les heuristiques algorithmiques

Elles consistent à utiliser des moyens mathématiques ou algorithmiques pour proposer une solution proche de l'optimale d'un problème donné. Nous pouvons parler de notre algorithmique de Kruskal qui est une heuristique. Elle est basée sur deux principes :

- La désactivation des liens qui créent des circuits, pour obtenir le réseau de plus court chemin.
- Appliquer notre stratégie d'économie d'énergie en permettant au contrôleur d'activer un seul port de routeur afin de maximiser le nombre de ports désactivés.

3.3.3. Le graphe de flot

Le graphe de flot est défini en Recherche Opérationnelle, comme un graphe orienté

$G = (V, E)$ avec un nœud source "o" et un nœud puit "d". Chaque arc e a une capacité positive dénotée par $u(e)$ (voir figure 3.3).

Nous notons que P_1 et P_2 sont des sous programmes de P (P étant un PLNE). La figure 3.3 montre un exemple d'exécution d'algorithme de résolution de PLNE. Sur la figure, le sous-programme P_1 n'est pas faisable et donc, il est abandonné, mais P_2 est divisé en P_{21} et P_{22} . La solution de ces deux sous programmes est une variable entière et P_{21} est optimale, ayant pour fonction objectif inférieure à P_{22} (pour un problème de minimisation).

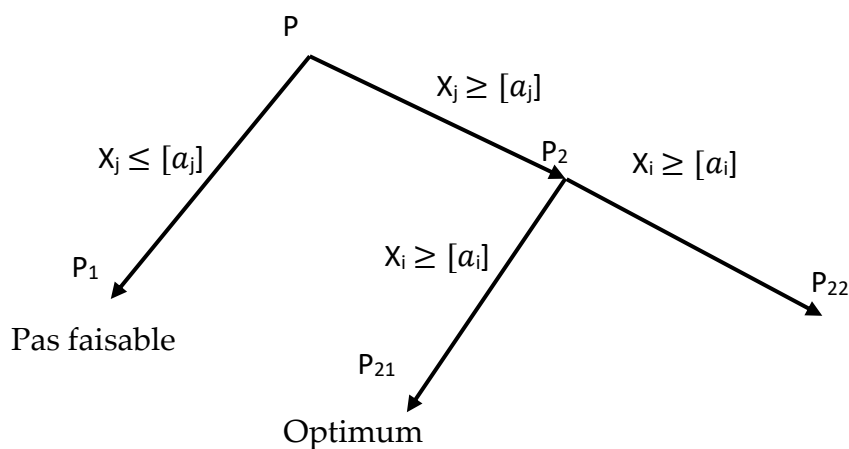


Figure 3.3 Exemple d'algorithme de résolution ILP [59]

Une des contraintes par exemple peut être : $f(e) \leq U(e)$ qui signifie que le flot qui passe sur un arc ne doit pas dépasser sa capacité $C(e)$.

3.4. Critère et choix de la méthode de résolution

La méthode de résolution devra tenir compte du dimensionnement du réseau et les ports des routeurs.

Notre proposition tient compte de la taille du réseau à travers le nombre de routeurs et autres équipements réseaux. Nous procédons par extinction progressive des routeurs tout en désactivant les ports des routeurs adjacents. Etant donné que le lien est la jonction entre deux interfaces de routeurs, l'extinction donc du port entraînerait celui de ses liens. Les variables utilisées sont de types entiers (détermination du

nombre de ports, liens, routeurs, la consommation du routeur...). Notre réseau est constitué de plusieurs nœuds formant ainsi un graphe. La fonction "gain d'énergie" est linéaire par rapport aux nombres de ports désactivés à chaque instant t .

Notre choix dans ces cas d'optimisation est le programme linéaire en nombres entiers avec des contraintes de qualité de service et une approche algorithmique pour décrire notre stratégie d'économie d'énergie.

La résolution du problème revient donc à :

- ✓ Résoudre par PLNE (Programme linéaire en nombres entiers), la fonction
$$P_i(t, \ell(t)) = C_i - (a_i \ell(t) + b_i)$$
- ✓ Résoudre par l'algorithme de Kruskal en cherchant l'arbre maximum, la fonction $P(k(i, j, t))$.

Plusieurs algorithmes existent pour résoudre le problème de la recherche de l'arbre maximum de poids le plus faible. Parmi ces algorithmes, nous citons par exemple l'algorithme de Kruskal, Prim et bien d'autres...

Le choix de l'algorithme de résolution de notre modèle doit tenir compte du dimensionnement du réseau. S'agissant du réseau ou sous réseau, nous voulons établir un minimum de liaisons à travers la recherche d'un arbre. Et, le coût global de cet arbre doit être le plus faible possible. Il faudra, pour cela, que l'arbre recherché soit de poids minimum.

L'algorithme de Kruskal (voir ci-dessous), est considéré comme l'algorithme le plus simple à comprendre et à implémenter. Contrairement à Prim, l'algorithme de Kruskal nécessite au préalable le tri des arêtes. Compte tenu du fait que notre réseau utilise 44 routeurs, inférieur à 100, l'algorithme de Kruskal est le meilleur pour ces types de réseaux (voir algorithme de Kruskal ci-dessous). Le temps d'exécution de l'algorithme est petit ayant un espace de stockage aussi petit que l'algorithme Prim. Ces différentes caractéristiques justifient notre choix dans cette thèse. L'utilisation de l'algorithme Prim serait beaucoup efficace si nos données étaient supérieures à 100.

Algorithme de Kruskal : Arbre recouvrant de poids minimal

Données : un graphe simple pondéré $G = (X, E, W)$ // E = Ensemble des arêtes; et
 W = Ensemble des poids des arêtes; X = Ensemble des sommets;

// La liste des arêtes dans l'ordre de leur poids croissant.

Résultat : un arbre $T = (X, U)$ recouvrant de poids minimal //

// Initialisation

1 $Coût \leftarrow 0$

2 $U \leftarrow \emptyset$

// Boucle principale

3 **Tant que** $|U| < N-1$ **faire**

4 Chercher la plus petite arête $[u, v]$ n'appartenant pas à U

5 **si** $U \cup [u, v]$ ne forment pas de cycle **alors**

6 | $U \leftarrow U \cup [u, v]$

7 | $Coût \leftarrow Coût + W_{u,v}$

8 **sinon**

9 | Eliminer l'arête $[u, v]$

10 **retourner** U

3.5. Représentation du modèle via OMNET++

Nous construisons un réseau (arbre maximal de poids minimal) à partir duquel nous obtiendrons nos économies d'énergie possibles. Les ports à éteindre sont ceux des routeurs adjacents. Pour notre simulation, nous utilisons un réseau de 7 routeurs comportant chacun huit (08) ports et neuf (09) liens (**figure 3.4**).

Notre arbre maximal de poids minimal est (**figure 3.5**) :

$12+12+12+4+4+5 = 49$. Cette écriture représente l'arbre recherché après exécution de l'algorithme d'extinction des liens qui créent les cycles.

Nous allons utiliser l'algorithme de Kruskal comme algorithme de plus court chemin. Ainsi, nous présentons notre réseau qui est un graphe, composé de sept routeurs et neuf liens parmi plusieurs. Ce graphe est construit avec le simulateur OMNET++ et le code source utilisé est décrit ci-dessous.

Dans la section (3.6), nous présentons la stratégie d'économie d'énergie par la description du mode de désactivation des ports.

Code Source permettant de construire le réseau sous OMNET++

Partie A : Processus de construction du réseau avec OMNET++ (Langage NED)

```
package org.omnetpp.queueing.routage7;
simple Router
{
    parameters:
        @group(Queueing);
        @display("i=block/routing");
        String routingAlgorithm
@enum("random", "roundRobin", "shortestQueue", "minDelay")= default("random");
        volatile int randomGateIndex = default(intuniform(0, sizeof(out)-1));
        // the destination gate in case of random routing
    gates:
        inout gate[]; // Définition des différents ports.
}
// Création des différents routeurs dans le réseau.
// Nous modélisons graphiquement notre réseau sur OMNET++
//
network Network
{
    @display("bgb=512,314;bgl=2");
    types:
        channel C extends ned.DatarateChannel
        {
            delay = 0;
            datarate = 100Mbps;
        }
    submodules:
        router: Router {
            @display("p=38,25");
        }
        router1: Router {
            @display("p=38,263");
        }
        router2: Router {
            @display("p=227,157");
        }
        router3: Router {
            @display("p=227,263");
        }
        router4: Router {
            @display("p=470,263");
        }
        router5: Router {
            @display("p=470,157");
        }
        router6: Router {
            @display("p=470,23");
        }
}
    connections:
        router.gate++ <--> { delay = 100ms; } <--> router1.gate++;
        router1.gate++ <--> { delay = 100ms; } <--> router.gate++;
        router1.gate++ <--> { delay = 100ms; } <--> router3.gate++;
        router3.gate++ <--> { delay = 100ms; } <--> router1.gate++;
}
```

ici, il s'agit des positions
des positions des routeurs
dans le réseau.

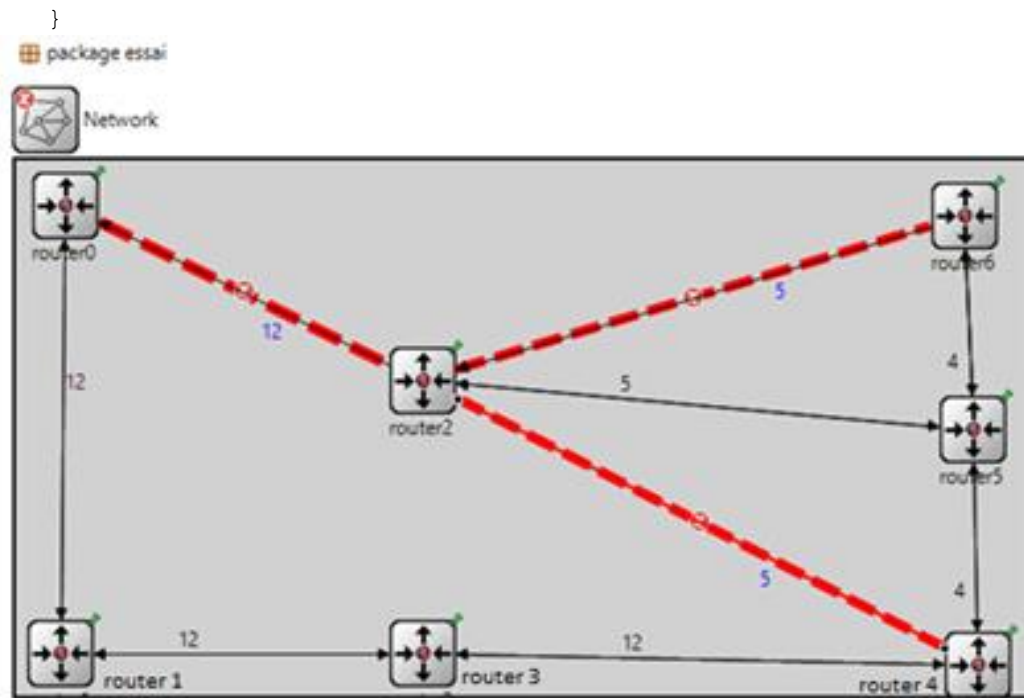


Figure 3.4 Architecture du réseau sur OMNET++

- **Commentaire**

Notre algorithme nous a permis de construire notre réseau. C'est un réseau de sept (07) routeurs avec huit (08) ports. Dans le réseau, tous les liens ou arêtes sont pondérés et le délai de transmission entre routeurs est de 100 ms. Les arêtes ou liens en pointillés sont susceptibles selon le choix de notre graphe connexe d'être désactivés. Et ce, de façon et dynamiquement. Notre algorithme (Algorithme de Kruskal) ci-dessous est chargé de désactiver le lien qui forme le cycle. Ce schéma a été obtenu sous OMNET++, Notre simulateur choisi (Voir Annexe C).

L'utilisation de l'algorithme Kruskal ci-dessous nous permet d'avoir le schéma suivant (voir Annexe C).

Partie B : Algorithme de Kruskal pour obtenir l'arbre maximal de plus faible poids.

***** Déclaration des variables*****

<pre>#include <iostream> #define MAX 30 int N; float Delais; float TD[100]; float delaiN=0; float Nber_port_ettiend=0; float Dlt=0; float Nber_router_etteind=0; float cont=0; float gainrecup=0; float gain_links_reup=0; int nber_router; int nber_links; int nber_port; float links=0; float links_off; float Nblinksr; float nberT; float link_off=0; float gain_port=0; float gain_links=0;</pre>	<pre>float Ci=0; typedef struct edge { int u,v,w; }edge; typedef struct edgelist { edge data[MAX]; int n; }edgelist; edgelist elist; int G[MAX][MAX],n; edgelist spanlist; void kruskal(); int find(int belongs[MAX],int vertexno); void union1(int belongs[MAX],int c1,int c2); void sort(); void print(); void sort1(); main() {</pre>
--	---

Les déclarations de l'algorithme suivant se trouvent en annexe C du manuscrit.

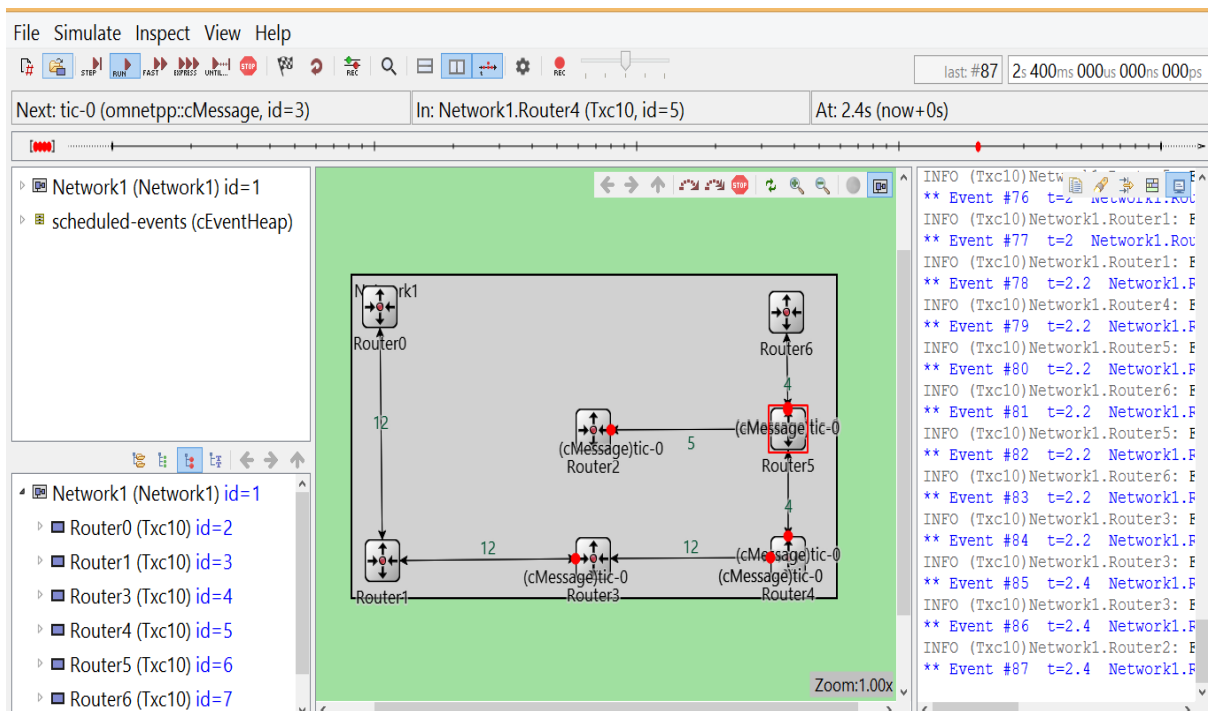


Figure 3.5 Architecture du réseau obtenu par l'algorithme Kruskal sur OMNET++

- **Commentaire**

Notre figure 3.5 est obtenu grâce à l'algorithme de Kruskal. Cet algorithme nous a permis de résoudre la problématique de la consommation dans les liens par la désactivation de ceux-ci, lorsqu'ils forment un cycle. Cette ingénierie développée nous permet ainsi d'éviter que les paquets se promènent trop dans le réseau (**voir fonction objectif** (3.6)). La table 3.3 ci-après montre le résultat de la simulation de cet algorithme d'extinction de liens ou ports. L'objectif était de trouver l'arbre recouvrant de plus faibles poids. Dans un graphe, il existe au moins une chaîne allant de la source à la destination. L'algorithme classe les poids des arêtes par ordre croissant. Pour trouver l'arbre recherché Z, il faut choisir les arêtes en fonction de leurs poids croissant. L'ajout d'une arête à la précédente est possible si celui-ci ne crée pas de cycles. Dès lors, il y'a cycle à chaque transmission de paquets vers une destination, l'algorithme ci - dessus en fonction du protocole de routage, désactive l'arête responsable de cycle jusqu'à la fin de tout le processus (n-1).

Table 3.3 Arbre maximal de plus faible poids

Arêtes	Poids croissant des arêtes	Ajouter à l'arbre recherché	Arbre recherché Z
[4,5]	4	Oui	4
[5,6]	4	Oui	4
[4,2]	5	Oui	5
[5,2]	5	Non(cycle)	-
[6,2]	5	Non(cycle)	-
[0,1]	12	Oui	12
[1,3]	12	Oui	12
[3,4]	12	Oui	12
[2,0]	12	Non(cycle)	-
Somme de Z = 49			

3.6. Description de l'algorithme de stratégie d'économie d'énergie

Notre stratégie d'extinction en fonction des demandes doit sélectionner tous les ports qui ne sont pas concernés par celles-ci et les éteindre dynamiquement.

Dans notre approche, lorsqu'un routeur est allumé, il établit un lien avec ses voisins et ce, dans tout le réseau. Notre réseau a sept (07) routeurs et huit (08) ports chacun. Seul un port est allumé par le contrôleur SDN et les $(n - 1)$ ports sont éteints avec leurs liens constituant ainsi notre gain d'énergie. Dans le choix des routes favorables, les cycles sont à éviter. Les chemins sont choisis en fonction de leurs plus faibles poids sans doublons. Cela éviterait que les paquets se promènent dans le réseau et donc évite de se perdre.

Le Protocole SPRING a été utilisé par les auteurs [2]. Ce protocole fonctionne avec le simulateur OMNET++ et est particulièrement adapté au SDN. Les auteurs ont éteint les liens dans le réseau. Nous avons orienté nos travaux vers les ports des liens (un lien est la jonction entre les interfaces) des routeurs. Etant donné que notre stratégie a

utilisé les ports alors nous avons modifié l'algorithme des auteurs [2] en proposant le nôtre (voir Algorithme Annexe C). Parlons un peu du Protocole SPRING.

❖ Protocole SPRING

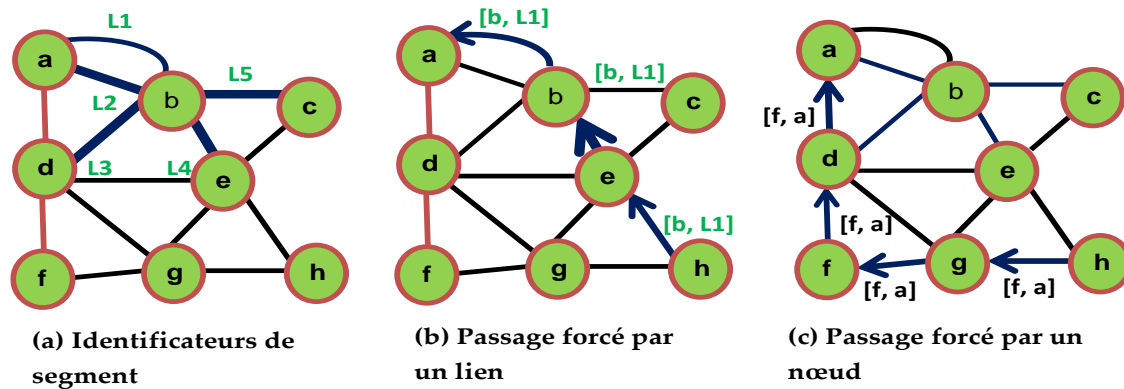


Figure 3.6 Le protocole SPRING [2]

Le protocole SPRING appelé Segment Routing (routage de segment) est un draft IEFT (Internet Engineering Task Force), un groupe de travail sur l'ingénierie Internet. Commencé en 2013, son but est de remplacer le tandem MPLS + RSVP - TE pour l'ingénierie du trafic. Il permet la flexibilité de la gestion du trafic en combinant la puissance du routage par la source et le routage par le plus court chemin. La réduction de son signal est meilleure que MPLS et celle de son entête mieux que Myrinet.

Myrinet est un protocole réseau haute-vitesse conçu par **Myricom**, afin d'être utilisé comme système d'interconnexion de plusieurs machines formant une grappe (ou cluster) [91]. Nous expliquons le mode de fonctionnement à travers les figures 3.6. Nous, nous intéressons à 2 catégories d'identifications : par nœud et par voisinage.

- L'identificateur du routeur est unique pour l'ensemble du réseau, identifie un nœud (a, b, c..., h).
- L'identificateur de voisinage est local à un routeur et cherche une interface de sortie (par exemple, le cas b, les interfaces sortantes sont représentées par L1, L2, L3, L4 et L5).

Lorsqu'un paquet doit être envoyé par exemple au nœud a par le plus court chemin, il faut lui ajouter l'étiquette dans son entête. Donc une gestion du trafic beaucoup plus fine est possible :

- On ajoute l'entête [b, L1], le nœud h oblige le paquet à aller au nœud a en utilisant le lien L1 (figure ci-dessus)
- Si h veut envoyer une demande à a en passant par f, il lui faut ajouter l'entête [f, a]. Le paquet choisira le plus court chemin vers f et ce, jusqu'à a.

L'utilisation de ce protocole est un bon choix pour effectuer du ré-routage dynamique seul, comme avec SDN. Les plus courts chemins trouvés en fonction des poids des liens, il va s'agir de façon dynamique du transport des paquets selon la demande [source, destination et port de sortie].

La stratégie algorithmique est décrite en **annexe C** après celui de désactivation des liens.

- **Commentaire de la stratégie d'économie d'énergie**

Le graphe connexe est ainsi obtenu par compression des liens dans le réseau. Notre résolution d'économie d'énergie a été une solution algorithmique (**voir annexe C**). Notre arbre regroupe l'ensemble des plus courts chemins. A l'allumage des routeurs par le contrôleur, l'algorithme de Kruskal est lancé pour réduire le réseau par la désactivation des liens qui créent des cycles. Le contrôleur active un port et les autres restants (n-1) ports sont éteints. Le maximum donc des ports éteints va constituer notre gain d'énergie. Dans le réseau, il existe au moins une route ou port allumé sur chaque routeur de manière aléatoire et dynamiquement. La simulation de la stratégie d'économie donne les résultats 87.50% d'extinction dans les ports et 33.33% d'extinction dans les liens possible pour faire une économie d'énergie. Ce qui induit de manière significative des économies d'énergie dans le réseau.

Le tableau ci-dessous (**Table 3.4**) récapitule les données obtenues à la suite de notre simulation.

3.6.1. Evaluation numérique du modèle

Dans cette partie, nous présentons les résultats issus de l'algorithme de stratégie d'économie d'énergie dans la table 3.4.

Table 3.4 Evaluation de l'approche

Nombre de Routeurs ayant 8 ports	Ports éteints	Economie d'énergie dans les ports	Nombre de Liens	Liens Eteints	Economie d'énergie dans les liens
7	49	87.5	9	3	33.33

Ce tableau traduit l'évaluation numérique de notre modèle. En effet, pour tester notre modèle, nous avons utilisé sept (07) routeurs ayant huit (08) ports chacun.

3.6.2 Performance de l'approche

Notre approche d'économie énergétique quel que soit l'arbre maximal de poids le plus faible donnerait de meilleurs résultats avec le modèle mathématique. Les ports éteints sont ceux des routeurs adjacents. Une fois le port éteint, il entraîne aussi l'extinction de son lien. Le SDN est une technologie basée sur la programmabilité des équipements réseaux à travers un ou plusieurs contrôleurs centralisés dans le réseau. Les règles de routage, de gestion de trafic et de la qualité de service sont contenus dans le contrôleur. A travers la vue globale du réseau, le contrôleur éteint les ports et liens lorsque ceux-ci ne sont pas utilisés au moyen de la modification du Protocole SPRING (MSP). Ce travail préliminaire concerne l'esprit dans lequel notre algorithme de désactivation des ports fonctionne. L'objectif de cette stratégie est de faire une économie d'énergie dans les réseaux. Le choix du nombre de routeurs utilisé se justifie dans l'expérience des travaux des auteurs [1]. Sept (07) routeurs sont choisis pour faire leur expérience.

La Table 3.4 est donc obtenue à partir de l'algorithme de Kruskal utilisé dans le simulateur OMNET++ pour donner **les figures 3.4 et 3.5**. Nous avons réduit notre réseau par la minimisation de ses liens en fonction des poids de ceux-ci tout en

évitant les circuits. Ce qui aboutit à la désactivation de trois (03) liens sur 9 au total. Etant donné le réseau comporte 7 routeurs ayant 8 ports, une fois le routeur allumé, seul un est activé et tous les autres sont éteints, nous avons donc:

Nombre total des ports dans le réseau = $7 \times 8 = 56$.

Nombre de ports éteints = $7 \times 7 = 49$ comme montré dans la Table 3.4 de (la section 3.6.1).

3.7 Conclusion

Cette section a présenté la proposition de notre première contribution. Nous avons proposé un modèle mathématique et une nouvelle stratégie d'économie d'énergie dans les réseaux. Nous pouvons dire que notre modèle est souple et donne ainsi une double économie énergétique dans les ports et liens. Les résultats de nos simulations algorithmiques donnent **87.5%** d'extinction dans les ports possible et **33.33%** dans les liens pour faire d'économie d'énergie. Dans la littérature, les auteurs ont minimisé dans les liens. Nous citons par exemple les travaux des auteurs de [1, 2]. Les résultats de ces auteurs ont donné respectivement 10-20% d'extinction de liens avec comme approche le test-bed (système expérimental) [1] et 44% de liens éteints dans l'utilisation de l'approche hybride des auteurs de [2]. Nous avons utilisé l'approche hybride et notre conclusion est qu'en plus des liens, il est possible de réduire la consommation de l'énergie dans les ports.

Cependant, des améliorations restent tout de même à réaliser sur le modèle mathématique trouvé.

Dans le chapitre suivant, nous tacherons de montrer que la fonction "gain d'énergie" peut être approximée à une fonction exponentielle.

**CHAPITRE 4 : APPROCHE
ALGORITHMIQUE DE LA MINIMISATION
DE LA CONSOMMATION UTILISANT LA
FONCTION EXPONENTIELLE COMME
CHOIX D'ECONOMIE D'ENERGIE DANS
LES RESEAUX SDN**

4.1. INTRODUCTION

Depuis plusieurs décennies, l'énergie est devenue un enjeu très important dans les Technologies de l'Information et de la Communication.

L'augmentation du nombre d'équipements informatiques a des conséquences directes sur la consommation de l'énergie lorsque ceux-ci ne sont pas utilisés. L'interconnexion de ces équipements, peu importe le débit, consomme assez d'énergie. Les auteurs [92] dans "Analyse des principales techniques d'économies d'énergie en réseau filaire Ethernet, 2015" ont montré que la croissance rapide de la consommation de l'énergie est due à l'augmentation du trafic dans le réseau (voir figures 4.1).

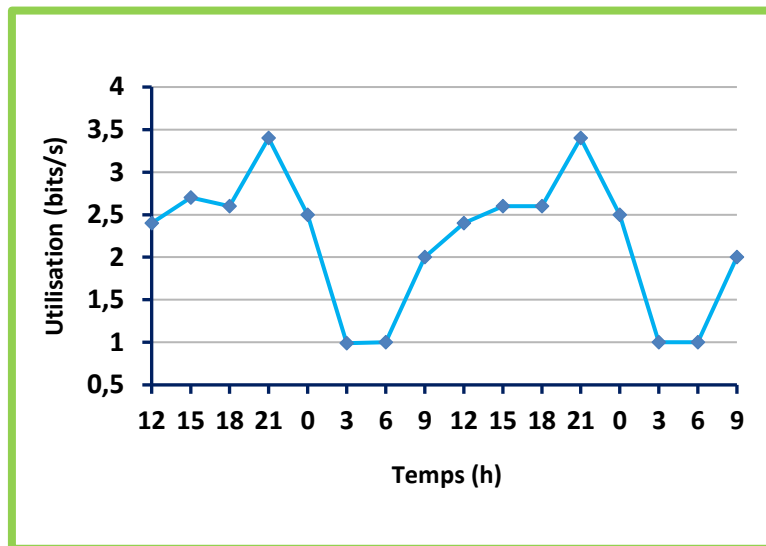


Figure 4.1 Jour typique d'utilisation du réseau [92]

Cette figure présente l'utilisation du réseau internet au cours d'une journée typique. Nous remarquons que les liens sont sous utilisés entre minuit et huit heures et plus utilisés pendant le jour de 09 heures jusqu'à 21 - 22 heures (un pic à 21 heures). Pendant ces périodes de pic, la consommation de l'énergie est préoccupante. Une solution aux périodes de pic apporterait une économie d'énergie non négligeable.

Une étude de Cisco estime qu'en 2016, on devrait dépasser le zettaoctet (10^{21}) de trafic IP échangé en un an [93].

Les auteurs [5] et [6] ont quant eux montré que les TIC consomment à elles seules, 2 à 10 % de la consommation mondiale [5]. Aux Etats Unis, elle est de 6 Twh/an [6]. Les travaux de KRA et al. [94] ont proposé une solution basée sur un modèle mathématique et une nouvelle stratégie d'économie d'énergie. Cette solution est appelée "fonction gain d'énergie" qui est linéaire et strictement croissante en fonction du nombre de ports désactivés.

Dans ce chapitre, nous proposons une approche algorithmique utilisant la fonction exponentielle pour augmenter le gain d'énergie.

Dans les sections suivantes, nous rappelons les travaux antérieurs d'abord, ensuite, le modèle mathématique de la fonction gain d'énergie approximée à la fonction exponentielle, puis la résolution et l'évaluation du modèle, enfin, nous comparons nos résultats avec les précédents avant de conclure.

4.2 Fonction exponentielle comme modèle mathématique pour réduire la consommation de l'énergie.

Les auteurs [1, 69, 88] et [94] ont traité le problème de la consommation de l'énergie avec la technologie SDN. KRA et al. [94] ont proposé un modèle mathématique et une stratégie d'économie d'énergie avec pour simulation basée sur l'extinction de ports sous contrainte de la qualité de service. Une double économie (ports et liens) a été réalisée. Une approche On/Off utilisée par [1] a permis de minimiser l'énergie en désactivant progressivement des liens ou routeurs sous contrainte de trois paramètres (délai, perte de paquets et gigue). Une désactivation de 10 à 20 % de liens a été observée, possible de faire une économie d'énergie. Mais, une congestion due à une variation de délai a entraîné des pertes de paquets, pendant le trafic dense. S'agissant des auteurs [69, 88], le nombre de ports des routeurs allumés, font

consommer le réseau. Lorsque les ports des routeurs ne sont pas utilisés, on les éteint afin de réduire l'énergie [94].

Une fois les routeurs allumés, seul un port est activé et les (n-1) autres sont désactivés. Nous utilisons la fonction exponentielle pour expliciter notre fonction "gain d'énergie" du routeur "i". Notre résolution est algorithmique s'appuyant sur l'algorithme de Kruskal tout en cherchant l'arbre maximal de plus faible poids de la fonction $f(k(i, j, t))$. Pour cela, nous posons deux hypothèses, après que nous ayons proposé le modèle mathématique de la fonction exponentielle qui est une suite de la deuxième contribution.

4.2.1. Nouvelle formulation du problème

Dans cette section, nous proposons le modèle mathématique utilisant la fonction exponentielle. Notre idée est similaire aux travaux de KRA et al. [97]. Dans le précédent chapitre, notre fonction "gain d'énergie" représenté par $a\ell(t) + b$ est linéaire strictement croissante par rapport au mode de désactivation des ports. Le choix de la fonction exponentielle parmi d'autres fonctions, vient du fait que celle-ci croît plus vite que la première (la fonction affine) et est strictement croissante. Son utilisation devrait, cependant, offrir une bonne économie d'énergie.

Nous formulons notre problème comme programme linéaire en nombres entiers, dans l'approche algorithmique de la minimisation de la consommation d'énergie dans les réseaux de type SDN.

Nous considérons "N", un réseau et $n(t)$, son sous réseau, avec ($n(t)$ inclus dans $N(t)$). L'approche étant dynamique, nous désactivons ou activons des ports à chaque temps (t). Nous observons au fur et à mesure le comportement du réseau par rapport à la consommation d'énergie.

Sachant qu'un routeur ne peut rester éteint, il est allumé et tous ses ports sont éteints sauf un sur chaque routeur dans le réseau sans dégrader la qualité de service en termes de délai, perte de paquets et gigue.

Nous définissons une fonction " $\ell(t)$ " comme étant le nombre de ports à désactiver :

$$\ell: [0, +\infty[\rightarrow \mathbb{N}$$

$$t \mapsto \ell(t)$$

Nous considérons λ_{max} comme étant le nombre total de ports sur un routeur. Ce paramètre est fixe et dépend du type de routeur.

D'après l'équation (3.2) du chapitre précédent, ci-dessous :

$$P(n(t), t) = \sum_{i \in n(t)} P(i, \ell(t), t) + \sum_{i, j \in n(t)} P(k(i, j), t) \quad (4.1)$$

$P(i, \ell(t), t)$ est la consommation de l'énergie du routeur i dont $\ell(t)$ ports ont été désactivés à l'instant t et $P(k(i, j), t)$ représente l'énergie du lien i, j au temps t . Nous rappelons que la variable k , peut être soit, à l'état 0 où le port est éteint ; soit, à l'état 1 où le port est actif. La capacité $C(i, j)$ du lien, traduit le volume de paquets à respecter sur ce lien.

Nous allons exprimer la fonction $P(i, \ell(t), t)$. La fonction gain d'énergie du routeur i : "g". "g" est une fonction exponentielle strictement croissante sur \mathbb{R}^+ par rapport aux nombres de ports désactivés à l'instant t .

Soit $g_i(t, \ell(t))$, le gain de l'énergie du routeur i tel que $\max_t \ell(t) \leq \lambda_{max}$ et

$$g_i(t, \ell(t)) = e^{a_i \ell(t) + b_i}$$

Nous voulons dans ce chapitre, comparer les résultats des deux fonctions "gain d'énergie" $(a_i \ell(t) + b_i)$ et $e^{a_i \ell(t) + b_i}$

$$\text{Si } \max_t \ell(t) < \lambda_{max}, \text{ alors } P(i, \ell(t), t) = C_i - e^{a_i \ell(t) + b_i} \quad (4.2)$$

Sinon $P_i(t, \ell(t)) = 0$, où C_i est la consommation globale du routeur i avec

$$P(i, \ell(t), t) = P_i(t, \ell(t)).$$

L'équation (4.2) est donc l'expression de notre fonction consommation de l'énergie d'un routeur à minimiser. Nous déduisons que si tous les ports sont éteints, $g_i(t, \ell(t)) \approx C_i$

Notre fonction objectif est convexe sur $[0, +\infty[$ puisque sa dérivée seconde est toujours positive. Elle admet donc un minimum et est de la forme :

$$\min(P(n(t), t)) = \min \left(\sum_{i \in n(t)} (C_i - (e^{a_i \ell(t) + b_i})) + \sum_{i, j \in n(t)} P(k(i, j, t)) \right) \quad (4.3)$$

Comme $C_i = \text{Consommation du Châssis } (C_{ch}) + \text{Consommation des ports } (C_p)$ allumés, notre fonction objectif devient :

$$\min(P(n(t), t)) = \min \left(\sum_{i \in n(t)} ((C_{ch} + C_p) - (e^{a_i \ell(t) + b_i})) + \sum_{i, j \in n(t)} P(k(i, j, t)) \right) \quad (4.4)$$

$$S/C \quad q(f(t), R(F(t), n(t), k(t), t)) \leq Q(f(t)), \forall f(t) \in F(t)$$

$$\text{Max}_t \ell(t) < \lambda_{max}, \text{ et } k \in \{0, 1\}$$

Table 4.1 Résumé des notations

$P(n(t), t)$	Consommation d'énergie dans le sous réseau
λ_{max}	Nombre total des ports sur un routeur
C_i	Energie globale du routeur
$e^{a_i \ell(t) + b_i}$	Gain d'énergie du routeur i obtenu en fonction des ports désactivés.
$P(k(i, j, t))$	Consommation des ports connaissant l'état des ports à l'instant t .
q	Qualité de service observée par les utilisateurs
$F(t)$	L'ensemble de toutes les demandes à l'instant t .
$f(t) \in F(t)$	Demande observée d'un utilisateur d'un point s à p à l'instant t
$R(F(t))$	Routage de toutes les demandes à l'instant t .

$n(t) \subseteq N(t)$ Le sous réseau est inclus dans le réseau à l'instant t .

$Q(f(t))$ Qualité de service fixée à ne pas excédée pour toutes demandes observées

$\text{Max}_t \ell(t)$ Le maximum de ports désactivés à l'instant t .

4.2.2. Données du problème

Elles concernent les éléments que nous utilisons dans ce chapitre. Ce sont :

- ✓ Les ports des routeurs ont le taux de consommation
- ✓ Utiliser le Switch **Cisco catalyst 3560-E** à 48 ports dont la consommation de l'ensemble est de 15.4w (utilisation de 8 sur 48 ports dont la consommation des 8 fait 2.56w). Un port consomme donc **0.32w**.
- ✓ Nous supposons que la fonction exponentielle est une solution d'économie d'énergie dans les réseaux.

Notre méthode de résolution consiste à :

- ✓ Résoudre par le programme linéaire en nombres entiers la fonction $P(i, \ell(t), t) = C_i - (e^{a_i \ell(t) + b_i})$.
- ✓ Résoudre la fonction $P(k(i, j, t))$ par l'algorithme de Kruskal pour trouver l'arbre maximal de plus faible poids. Cet algorithme nous permet de faire une économie dans les liens.

4.3. L'approche algorithmique

Dans cette partie, nous proposons une solution algorithmique pour trouver le plus court chemin dans le réseau. Il consiste à rechercher l'arbre maximal de plus faible poids (Figure 4.3).

Notre réseau est composé de sept (7) routeurs avec huit (8) ports chacun et neuf (9) liens [95]. Les ports éteints, sont ceux des routeurs adjacents. Un port éteint a un impact sur le lien (lien s'éteint). La figure 4.2 est le schéma de notre modèle graphique.

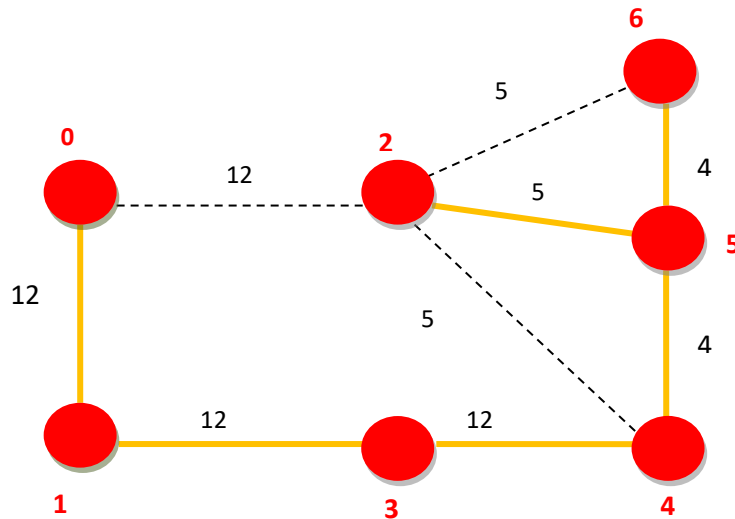


Figure 4.2 Réseau avec capacités [94]

De ce schéma (Figure 4.2), on aboutit à la figure 4.3 en appliquant l'algorithme de Kruskal sur les liens (voir l'algorithme ci-dessous).

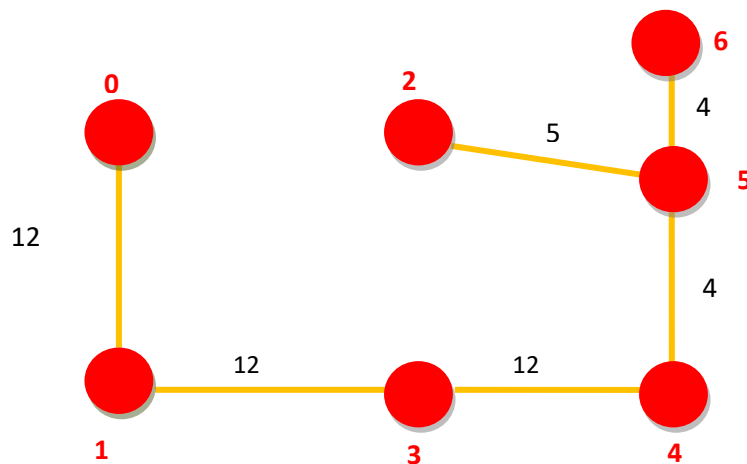


Figure 4.3 Architecture obtenue par l'algorithme de Kruskal [94]

Cette figure 4.3 est obtenue à la suite de la désactivation des liens qui créent des circuits dans le réseau.

4.3.1. Principes de l'approche

Données : Notre graphe connexe composé de X sommets est donné par la liste de ses arêtes dans l'ordre de leur poids croissant.

Soit T l'arbre recherché.

4.3.2. Méthode d'utilisation de l'approche

La liste des arêtes est lue en commençant par l'arête de plus faible poids $|U|$. Une fois, l'arête de plus faible poids est trouvée, elle est comparée à celle du sommet suivant. Si, elle ne forme pas de cycle avec U alors, elle est retenue. L'algorithme s'arrête lorsque le nombre d'arête est égal à $X-1$. L'algorithme peut être trouvé dans le chapitre 3 de notre thèse. Le paramètre U, est appelé "Ensemble des arêtes dont la somme des poids est minimal. L'arbre recherché "T" est un arbre minimal ayant le même nombre de sommets que le graphe de départ.

4.4. Stratégie d'économie d'énergie par extinction des ports

Notre stratégie d'extinction en fonction de la demande, une fois les routeurs allumés, doit sélectionner tous les ports qui ne sont pas concernés par demande et les éteindre. Dans notre expérience, il y'a 7 routeurs ayant chacun 8 ports. Une fois les routeurs allumés, seul un est activé et les (n-1) autres éteints c'est-à-dire 7. Les paquets sont redirigés vers les ports activés en fonction de l'algorithme précédent.

Le protocole SPRING adapté à notre stratégie d'économie d'énergie [94], est chargé de transférer les paquets vers des destinations (**exemple : pour une demande [0, 2] doit utiliser le routeur en passant sur le lien de poids 5**) (voir l'algorithme en annexe c).

Le contrôleur SDN voit le réseau dans sa globalité. Une fois les nouveaux ports sélectionnés, celui-ci informe chaque nœud des routes et ports qui doivent être introduits dans sa table d'acheminement.

Dans cet algorithme, lorsque les nouvelles routes sont calculées, le contrôleur qui voit le réseau dans sa globalité, informe chaque routeur des ports qui doivent être connus dans sa table d'acheminement. Après cela, les ports sont éteints ou allumés.

Pour l'extinction des ports l'algorithme décrit le processus.

Dans notre exemple, les ports des routeurs 0, 1, 3, 4, 5, 2 et 6 ont été sélectionnés ou identifiés et les autres sont immédiatement éteints.

Nous évaluons la performance de notre approche.

4.5. Evaluation et performance de l'approche

Les auteurs [1] ont travaillé dans un réseau de 44 routeurs. Mais c'est 7 routeurs sur 44 (37 routeurs allumés), soit 15 à 20 % de liens désactivés pour une économie d'énergie. Nous allons donc utiliser 7 routeurs ayant 8 ports et 9 liens. Nous utilisons aussi un type de routeur nommé Cisco Catalyst 3560E à 48 ports [95]. C'est 8 ports avec 0.32w de consommation par ports que nous utilisons. Nos résultats sont confinés dans deux tableaux. Le premier est l'évaluation du modèle de la fonction exponentielle et le second celui des auteurs [94] que nous comparons.

Table 4.2 Evaluation de la fonction exponentielle

<i>Nbre de routeurs de 8 ports</i>	<i>Ports éteints</i>	<i>Economie d'énergie</i>	<i>Gain 0.32w/port</i>	<i>Nombre de liens</i>	<i>Links éteints</i>	<i>Economie d'énergie</i>	<i>Gain 0.64w/port</i>
7	133	133 / 152 = 87.5 %	28w	9	3	3/9 = 33.33	21.32w

Ce tableau ci-dessus utilise la fonction exponentielle pour évaluer le gain d'énergie. L'objectif est de voir l'évolution de l'énergie.

Table 4.3 Evaluation de la fonction linéaire [94]

<i>Nbre de routeurs de 8 ports</i>	<i>Ports éteints</i>	<i>Economie d'énergie</i>	<i>Gain 0.32w/port</i>	<i>Nombre de liens</i>	<i>Links éteints</i>	<i>Economie d'énergie</i>	<i>Gain 0.64w/port</i>
7	49	49/56=87.5	28w	9	3	3/9=33.33	21.32w

4.6. Comparaison des résultats

En utilisant la fonction "gain d'énergie" qui est une fonction exponentielle, nous trouvons les mêmes résultats que ceux des auteurs précédents.

Nous déduisons que notre approche et stratégie est acceptable et permet ainsi de conclure que la fonction exponentielle est un bon choix dans l'économie énergétique. Cependant, nous remarquons que notre fonction ($e^{a_i \ell(t) + b_i}$) croît plus vite que la fonction affine ($a_i \ell(t) + b_i$) des travaux des auteurs [94]. Ce qui traduit que la fonction gain a une meilleure performance, aussi bien que la fonction affine. Nous présentons à travers un tableau, les résultats des deux méthodes.

Table 4.4 Tableau récapitulatif des deux approches

	Nr routeurs ayant 8 ports	Nr ports total	Nr ports éteints	Economie d'énergie dans les ports	Gain d'énergie	Nr total de liens	Liens éteints	Economie d'énergie dans les liens	Gain d'énergie
$a_i \ell(t) + b_i$	7	56	49	$49/56 = 87.5\%$	28w	9	3	$3/9 = 33.33\%$	21.33w
$e^{a_i \ell(t) + b_i}$	7	152	133	87.5%	28w	24	8	$3/9 = 33.33\%$	21.33w

Ce tableau est le récapitulatif des précédents. Il fait la comparaison d'économie de l'énergie.

4.7. Conclusion

Le modèle mathématique dans la consommation de l'énergie dans les réseaux SDN utilisant une fonction exponentielle pour exprimer notre "gain d'énergie" est bon. Contrairement à ce que nous attendions, l'utilisation de la fonction exponentielle nous donne les mêmes résultats que dans les travaux des auteurs précédents c'est-à-dire **87.5%** d'économie d'énergie dans les ports soit **28w** et **33.33%** d'économie d'énergie dans les liens soit **21.32w**. Ces résultats sont nettement meilleurs que ceux des auteurs [1] qui est de 10 à 20% dans les liens soit **12.8w**.

En somme, nos approches et simulations peuvent être proposées comme approche d'économie énergétique. Par conséquent, nous affirmons que notre fonction "gain d'énergie" peut être approximée à la fonction exponentielle.

Dans le chapitre suivant, nous présentons la simulation du modèle mathématique minimisant la consommation de l'énergie dans les réseaux de type SDN sous

contrainte de délai. Dans ce chapitre, la comparaison de nos résultats et ceux des auteurs seront présentés à travers des figures.

CHAPITRE 5 : SIMULATION DE DELAI EN FONCTION DES PORTS DESACTIVES

5.1. INTRODUCTION

La simulation dans les travaux de recherche est utilisée pour évaluer l'approche ou l'idée exprimée. La première étape d'une nouvelle idée ou solution passe par le modèle. Celui-ci est évalué par simulation et est comparé avec d'autres modèles semblables dans la littérature.

En recherche scientifique, le simulateur occupe une place très importante dans le développement d'applications (algorithmes, heuristiques etc.). Il est utilisé dans plusieurs environnements, pour évaluer des travaux afin de faciliter la modification du modèle proposé, corrigeant ainsi les résultats de simulation.

Dans nos travaux précédents acceptés et publiés, nous avons fait mention des auteurs des articles [5] et [6]. Dans ces travaux, une étude a montré que les TICs à elles seules consomment 2 à 10% de la consommation mondiale [5]. Cette consommation est de 6Twh/an aux USA dans les hubs, Switch, routeurs [6].

Ces travaux, c'est à dire, dans les premières et deuxièmes contributions, nous avons proposé un modèle de minimisation de la consommation de l'énergie dans les réseaux SDN, par désactivation de ports à chaque instant "t" sous contrainte de délai, perte de paquets et gigue et sa pertinence a été montrée par une résolution algébrique.

Dans ce chapitre, il s'agira de présenter notre fonction délai en fonction du nombre de ports désactivés et l'outil de simulation qui a été développé.

Nous présentons dans la section 5.2, la synthèse sur quelques simulations.

La section 5.3 présentera le modèle mathématique [94] et ses contraintes en termes de délai, perte de paquets et gigue dans laquelle nous expliquons notre algorithme proposé. Dans la section 5.3.2, il va s'agir de la comparaison de notre approche avec celle des auteurs dans la littérature. La conclusion suivra en section 5.4.

5.2. Synthèse sur quelques simulations

Dans cette section, nous présentons les résultats de simulation des différents auteurs concernant la consommation de l'énergie dans les réseaux SDN.

Plusieurs travaux de simulations ont été traités dans l'optimisation de la consommation d'énergie dans les réseaux de communication. C'est le cas de citer les auteurs [1] et [2]. Dans [1], les auteurs ont utilisé une approche logicielle basée sur un système automatique de commande On/Off pour désactiver les liens des routeurs. C'est un test-bed c'est-à-dire un système expérimental qui a été fait dans les réseaux classiques. Les auteurs ont choisi dans leur topologie réseau des chemins pris en charge par le protocole CPN (Cognitive Protocole Network). L'expérience était de montrer l'impact qu'il y'a entre un chemin pris en charge par un protocole de routage et celui non pris charge en terme de qualité de service. Le résultat de leur approche a donné 10 à 20% d'extinction de liens possible pour faire une économie d'énergie avec dégradation de la qualité de service (voir figures ci-après). Nous avons pu reproduire les travaux des auteurs [1] (**Figure 5.1**).

Cette figure 5.1 a montré que la désactivation de certains routeurs aux instants 100s, 300, 500, 700s a causé un délai élevé. Cela est dû au fait qu'à chaque extinction des liens ou routeurs, le protocole de routage CPN, cherche de nouveaux chemins favorables ou alternatifs. Et ces chemins étant les plus courts (instants 200s, 400s, 600s) sont vite saturés. Seuls les plus longs chemins sont empruntés. Certains paquets se perdent en ces instants et souvent les paquets se trouvent dans les routeurs lors de sa désactivation. Dans les expériences des auteurs, les paquets utilisent dans leur entête la stratégie d'extinction basée sur le protocole CPN. Les routeurs sont devenus de simples zones de repos ou de route pour les prochaines destinations. Cet esprit des auteurs ressemblait déjà à la vision actuelle du SDN. A ces endroits, les paquets ayant une expérience cherchent des destinations favorables. Pendant l'extinction, les paquets se trouvant aussi dans la file d'attente sont détruits (Packet_in) (**Figure 5.2**).

Pendant que les paquets prennent du temps à trouver un chemin, certains d'entre eux se perdent. Car ceux-ci ont déjà atteint le nombre de saut à effectuer dans le réseau (30 sauts au maximum) au-delà, les paquets sont détruits par l'algorithme de

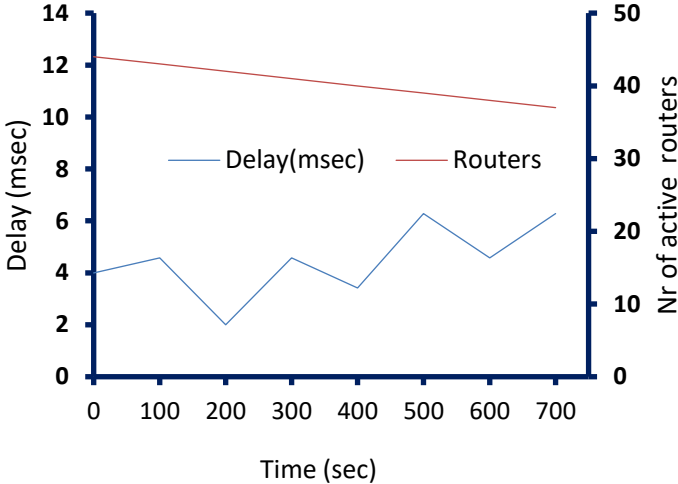


Figure 5.1 Delai- Première expérience [1]

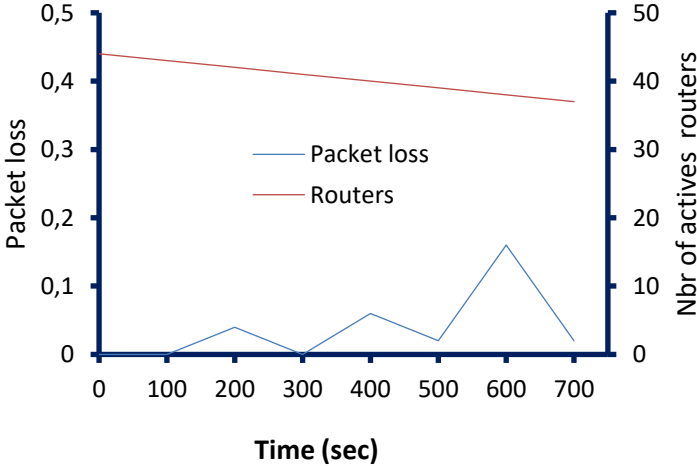


Figure 5.2 Perte de paquets – Première expérience [1]

destruction dans la configuration du routeur. Et ce, pour permettre aux autres paquets de se frayer un passage. L'extinction successive et progressive des routeurs

ou le nombre maximum de liens, augmentent le délai dû à la congestion dans le réseau entraînant ainsi des pertes de paquets atteignant 20% (**Figure 5.2**).

La figure 5.3 concerne la gigue. Elle n'a pas varié. Le rythme de variation du délai est constant. Il est de l'ordre de 0.08 soit 8%. Nous présentons les simulations des travaux des auteurs de [2]. Dans [2], les auteurs ont proposé une approche logicielle utilisant le protocole SPRING pour désactiver les liens lorsqu'ils ne sont pas utilisés. Leur approche montre dans les simulations qu'une extinction de 44% de liens est possible pour une économie d'énergie (voir les captures ci-dessous).

Dans les travaux [2], nous faisons le constat de leurs travaux sur trois points :

- ✓ **Impact sur la charge réseau** : il crée un surcoût de 18.56%. Les chemins sur lesquels sont routés les paquets ne sont pas les plus courts. Mais ce surcoût s'annule vite lorsque les liens sont allumés (**Figure 5.4(a) et Figure 5.4(c)**) ;
- ✓ **Impact sur le délai bout en bout** : l'augmentation a été de 37% avec Germany 50 alors que dans la plupart des cas, il est de 20% ;
- ✓ **Impact sur la perte des paquets** : pas de perte des paquets avec l'algorithme utilisé par les auteurs basés sur le Protocole Spring.

Nous avons utilisé l'idée des auteurs de [2] pour étendre les travaux des auteurs de [90]. Notre approche est logicielle. Elle nous a permis de désactiver le maximum des ports des routeurs. Cette désactivation des ports concerne ceux des routeurs adjacents. Elle est meilleure que celles des précédentes.

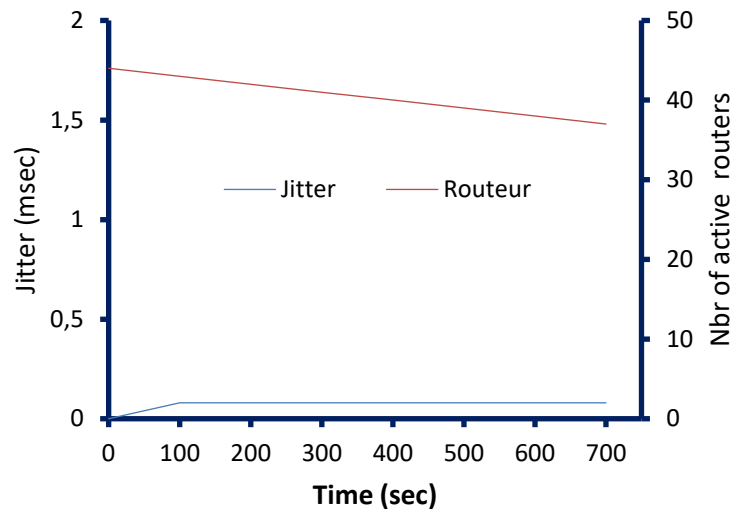


Figure 5.3 Gigue – Première expérience [1]

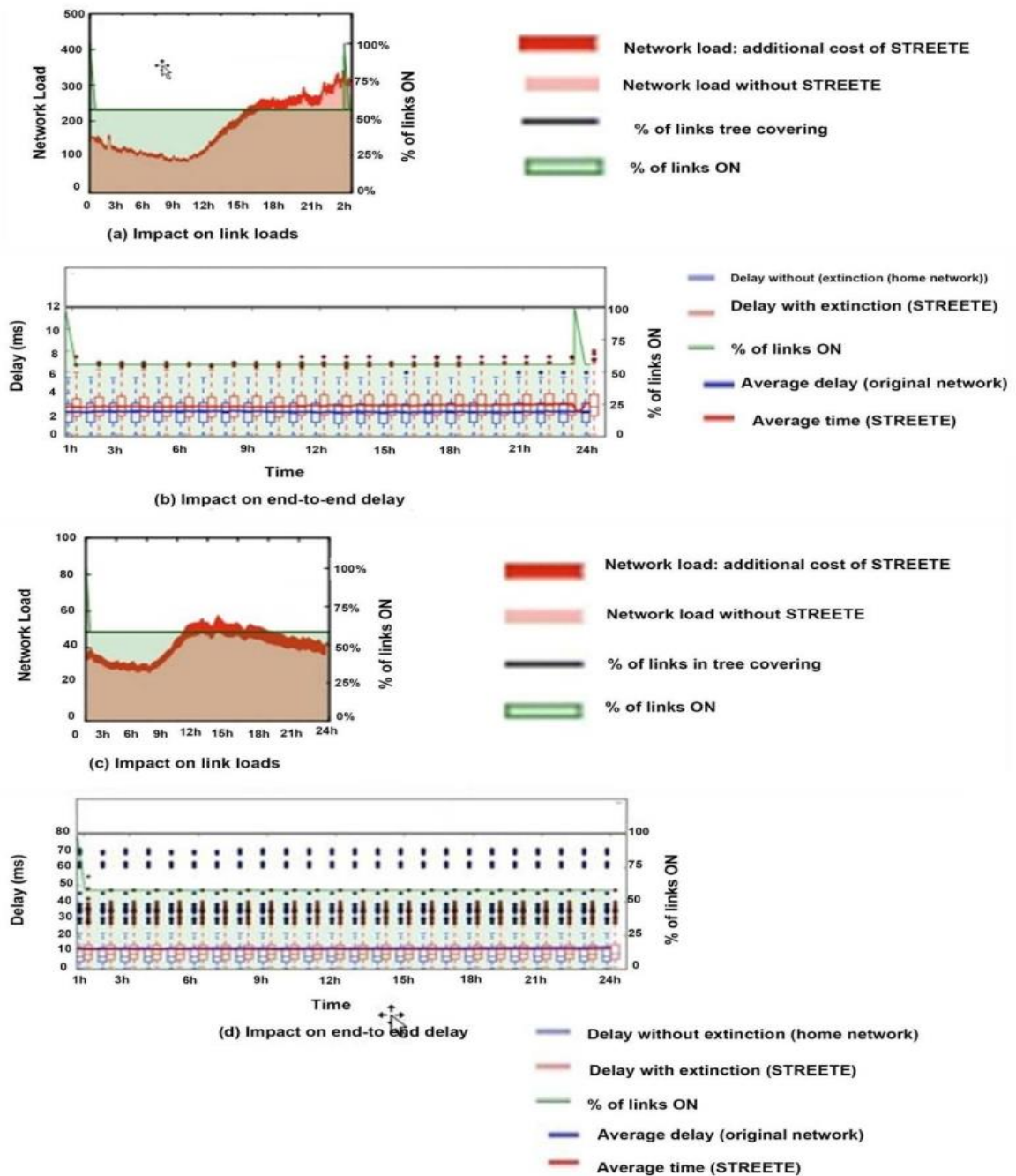


Figure 5.4 Simulations Results in Germany50 (top) and Giant (bottom) networks [2]

Notre approche offre une double économie d'énergie (dans les ports et dans les liens). Nos simulations les démontreront. Bien avant, nous rappelons notre modèle mathématique et puis suivra les résultats de nos simulations.

5.3 Présentation du modèle

Dans cette section, nous présentons notre modèle mathématique sous contrainte de délai, de perte de paquets et gigue. Nous avons montré le principe d'extinction des ports des routeurs par la modification du protocole SPRING [2] sur l'arbre recouvrant de poids minimum.

Notre modèle a été obtenu en utilisant le théorème de convexité. Ainsi, la fonction objectif obtenue est :

$$\min(P(n(t), t)) = \min \left(\sum_{i \in n(t)} ((C_{ch} + C_p) - (a_i \ell(t) + b_i)) + \sum_{i, j \in n(t)} P(k(i, j, t)) \right) \quad (5.1)$$

Si et Seulement si :

$$q(f(t), R(F(t), n(t), k(t), t)) \leq Q(f(t), \forall f(t) \in F(t) \text{ et :}$$

$$\sum f(t) \leq C_{i,j}, \forall f(t) \in F(t) \text{ Avec } \text{Max}_t \ell(t) < \lambda_{max} \text{ avec}$$

$$k \in \{0, 1\}$$

Cette fonction peut être trouvée au chapitre 3, section 3.2.2 du présent mémoire.

" $q(f(t), R(F(t), n(t), k(t), t)) \leq Q(f(t), \forall f(t) \in F(t)$ " est une fonction qui doit être respectée lorsque la fonction (5.1) s'exécute. Elle exprime que la qualité de service observée à chaque demande, ne doit pas dépasser la qualité de service fixée.

" $\sum f(t) \leq C_{i,j}, \forall f(t) \in F(t)$ " traduit que la somme des demandes observées des utilisateurs ne doit pas dépasser la capacité du lien.

" $\text{Max}_t \ell(t) < \lambda_{max}$ " exprime que le maximum de ports à désactiver ne doit pas dépasser le nombre de ports fixés et $k \in \{0, 1\}$ variable binaire.

Si $k = 0$, pas de ports actifs.

Sinon si $k = 1$ alors il existe un port actif.

Lorsque les routeurs sont allumés, le contrôleur se charge d'activer un seul un port sur chaque routeur et les (n-1) autres sont désactivés ou éteints. Les paquets ont au

moins un chemin sur chaque routeur. Cette approche va influencer le délai de transmission contrairement à celle des auteurs [1] et [2]. Nous définissons donc notre délai en fonction du nombre de ports désactivés appelé $D_{\ell(t)}$. Notre délai sera obtenu à partir des données des travaux de [1].

Ainsi, nous avons :

$$D_{\ell(t)} = \frac{\text{Délai}_n}{\max_t \ell(t)} \quad (5.2)$$

où $D_{\ell(t)}$: Délai lorsque $\ell(t)$ ports sont désactivés.

Délai_n : Délai normal (valeur obtenue à partir de celles de [1]).

$\max_t \ell(t)$: Nombre maximum de ports éteints sur chaque routeur.

Table 5.1 Résumé des notations

$P(n(t), t)$	Consommation d'énergie dans le sous réseau
λ_{max}	Nombre total des ports sur un routeur
C_i	Energie globale du routeur i
$a_i \ell(t) + b_i$	Gain d'énergie du routeur i obtenu en fonction des ports désactivés.
$P(k(i, j, t))$	Consommation des ports connaissant l'état des ports dans le sous réseau à l'instant t .
q	Qualité de service observée par les utilisateurs
$F(t)$	L'ensemble de toutes les demandes à l'instant t .
$f_{s,p}(t) \in F(t)$	Demande observée d'un utilisateur d'un point s à p à l'instant t
$R(F(t))$	Routage de toutes les demandes à l'instant t .
$n(t) \subseteq N(t)$	Le sous réseau est inclus dans le réseau à l'instant t .
$Q(f(t))$	Qualité de service fixée à ne pas excédée pour toutes demandes observées


```

cin>>Nt;
while(Nr<Nt){
    cout<<"\n\tLe nombre de test(s) doit etre inferieur ou egal( < = ) au nombre de Routeur !" <<endl;

    cout<<"\n\tEntrez le nombre de test(s): " <<endl;

    cin>>Nt;
}

int Temps[Nt], Routeur[Nt];

float Delais[Nt], Dlt[Nt];

if (Nt == 0){
    cout<<"\n\tVous avez effectuez zero (0) test." <<endl;
}
else{
for (i=0; i<Nt; i++){

    cout<<"\n\n\tSAISIES DES INFORMATIONS DU ROUTEUR " <<i+1 <<endl;
    cout<<"\t-----" <<endl;
    cout<<"\n\n\tEntrez le temps: " <<endl;
    cin>>Temps[i];
    cout<<"\n\n\tEntrez le Delais: " <<endl;
    cin>>Delais[i];
    cout<<"\n\n\tEntrez le numero du routeur : " <<endl;
    cin>>Routeur[i];
}
}

cout<<"\n\tRESULTATS : " <<endl;
cout<<"\t-----" <<endl;

if(Nt==0){
    cout<<"\n\tAucun resultat disponible!" <<endl;
}
else{
for (i = 0; i<Nt; i++){
    cout<<" " <<endl;
    cout<<"\n\tRouteur N " <<Routeur[i] <<endl;
    cout<<"\t-----" <<endl;
    cout<<"\tTemps : " <<Temps[i] <<endl;
    cout<<"\tDelais : " <<Delais[i] <<endl;
    cout<<"\tMax de Port(s) desactive(s) : " <<(Nr-Routeur[i]) <<endl;
    Dlt[i] = (Delais[i]/(Nr - Routeur[i]));
    if ((Nr - Routeur[i]) == 0){
        Dlt[i] = 0;
    }
    cout<<"\tDI(" <<Temps[i] <<"): " <<Dlt[i];
}
}
}

cout<<"\n\n\n\tVoulez-vous continuer ? (Taper O/N): " <<endl;

```

```

cin>>rep;
while(rep != 'N' && rep != 'n' && rep != 'O' && rep != 'o'){
    cout<<"\n\tRepondez par O/N :"<<endl;
    cin>>rep;
if(rep == 'O' || rep == 'o'){
    break;
}
if(rep == 'N' || rep == 'n'){
    cout<<"\n\tBye... :)"<<endl;
    cout<<"\n\n"<<endl;
    return 0;
}
}
}while(rep == 'O' || rep == 'o');

cout<<"\n\tBye... :)"<<endl;
cout<<"\n\n"<<endl;
return 0;
}

```

Nos résultats sont ainsi obtenus :

```

Entrez le nombre de Routeur(s):
44
Entrez le nombre de test(s):
0

SAISIS DES INFORMATIONS DU ROUTEUR 1
-----
Entrez le temps:
0
Entrez le Delais:
4
Entrez le numero du routeur:
44

ROUTEUR N 44
-----
Temps : 0
Delais : 4
Max de Port(s) desactive(s) : 0
DI(0): 0

ROUTEUR N 43
-----
Temps : 100
Delais : 4.57
Max de Port(s) desactive(s) : 1
DI(100): 4.57

ROUTEUR N 42
-----
Temps : 200
Delais : 2
Max de Port(s) desactive(s) : 2
DI(200): 1

Voulez-vous continuer ? (Tapez O/N):
Bye... :)
Process returned 0 (0x0)   execution time : 110.531 s
Press any key to continue.

```

Commentaire du résultat de l’algorithme de délai en fonction du nombre de ports désactivés.

Ce processus traduit l’idée de notre approche. Nous avons au moins un chemin sur chaque routeur. Les paquets ont facilement une porte de sortie dans le réseau. Ce qui induit une réduction de délai dans le réseau. Le délai est un facteur très important dans la gestion du trafic dans le réseau. L’objectif de notre thèse est d’optimiser la consommation de l’énergie dans les réseaux SDN sous contrainte de la qualité de service (délai, perte de paquet et gigue). Nous avons modélisé le délai de sorte qu’une fois, ce délai est réduit, le taux de perte de paquet est moindre. De même que la gigue.

- **Commentaires des résultats avec les travaux des auteurs précédents**

Nous avons les résultats de nos simulations et une étude comparative de ces résultats avec les précédents basés sur notre approche algorithmique dans OMNET ++ a été fait (voir la **Figure 5.5b**). Dans ce diagramme, au fur à mesure que nous désactivons les ports des routeurs, une bonne économie est observée. A chaque instant t, il existe un chemin par lesquels transitent les paquets en fonction de leurs poids. Ce qui évite aux différentes périodes, un encombrement dans le réseau. Un bon délai de transmission des paquets est constaté sur nos différentes figures. La raison s'explique, par le fait que la perte de temps pour que les paquets trouvent un port de sortie est réduit (**Figure 5.6b**).

La figure 5.6b est la conséquence de la figure 5.5b. La réduction du délai (les paquets trouvent vite un chemin) entraine un taux de perte de paquet très réduit. Dans notre cas, le taux est de 10% (**Figure 5.6b**).

Nous constatons de façon logique, une diminution double de la gigue de 0.04 (**Figure 5.7b**) contrairement à 0.08 des auteurs [1] (**Figure 5.7a**).

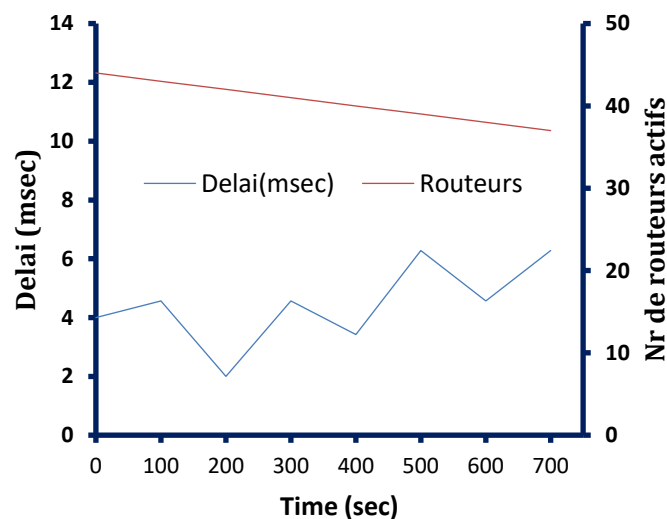


Figure 5.5a Délai -première expérience [1]

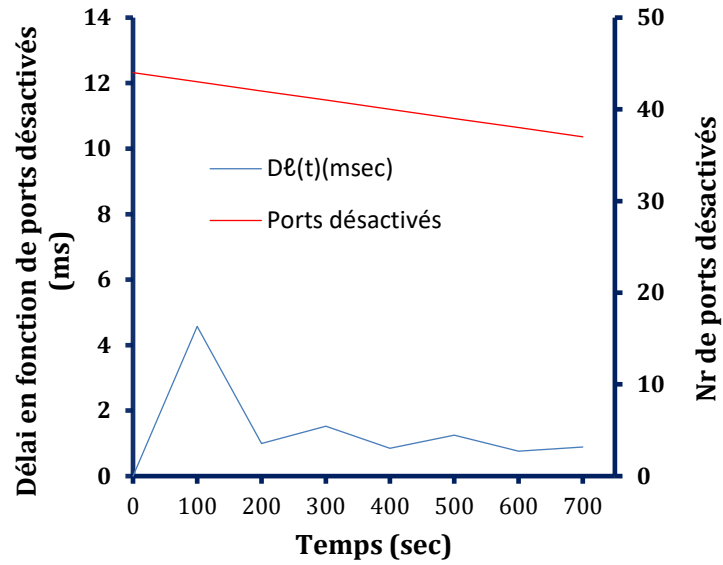


Figure 5.5b Délai avec ports désactivés Première expérience

L'importance de notre modèle de simulation de délai en fonction de ports désactivés, traduit les résultats ci-dessus. Ces résultats traduisent une bonne réduction de délai par rapport aux travaux des auteurs [1]. De 20% qu'était le délai des précédents auteurs, il est presque zéro dans le nôtre. Cela est dû à notre Stratégie d'Economie d'Energie (SEE).

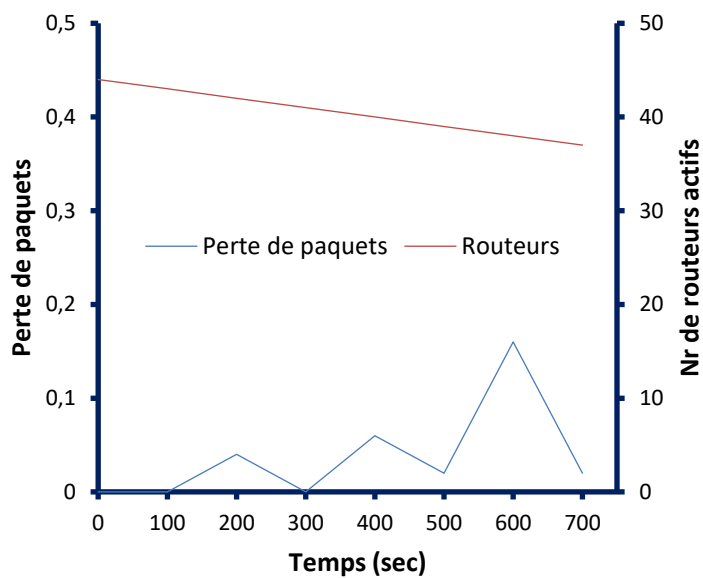


Figure 5.6a Perte de paquet- Première expérience [1]

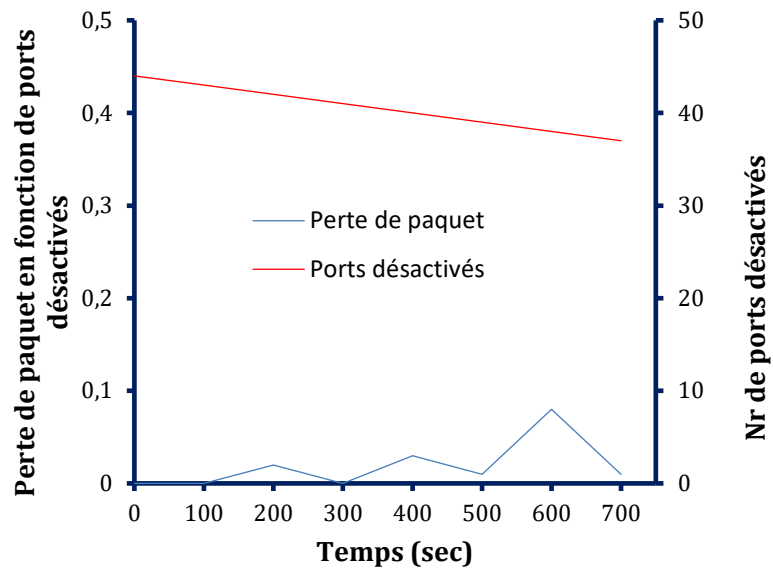


Figure 5.6b Perte de paquet avec ports désactivés - Première expérience

Ces figures ci-dessus (**Figure 5.6a**) sont la conséquence des précédentes (**Figure 5.5a**). Plus le délai augmente, plus il y a consommation de l'énergie dans les réseaux. Et donc formation d'un goulet d'étranglement qui entraîne des pertes de paquets.

Les résultats de délai dans nos travaux étant meilleur, ceci a donc influencé la perte des paquets, contrairement aux travaux des précédents auteurs. De 20% qu'était la perte des paquets, elle est de 10% dans nos travaux.

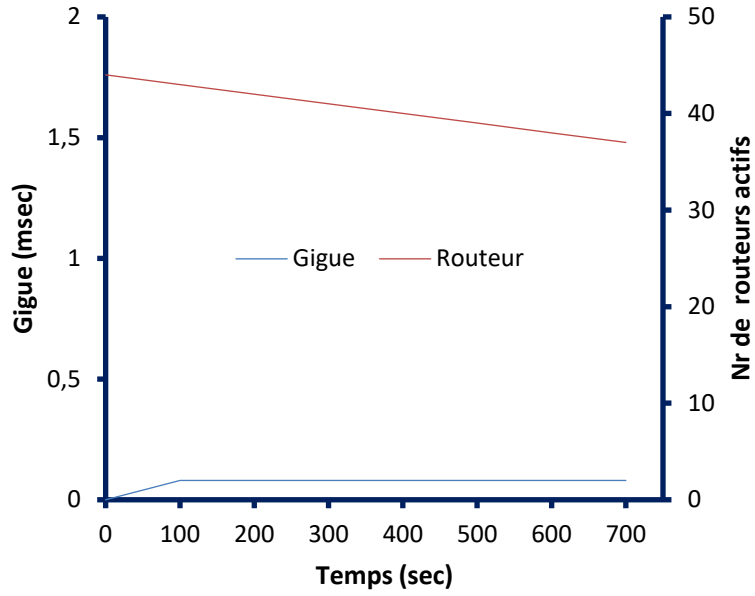


Figure 5.7a Gigue -Première expérience [1]

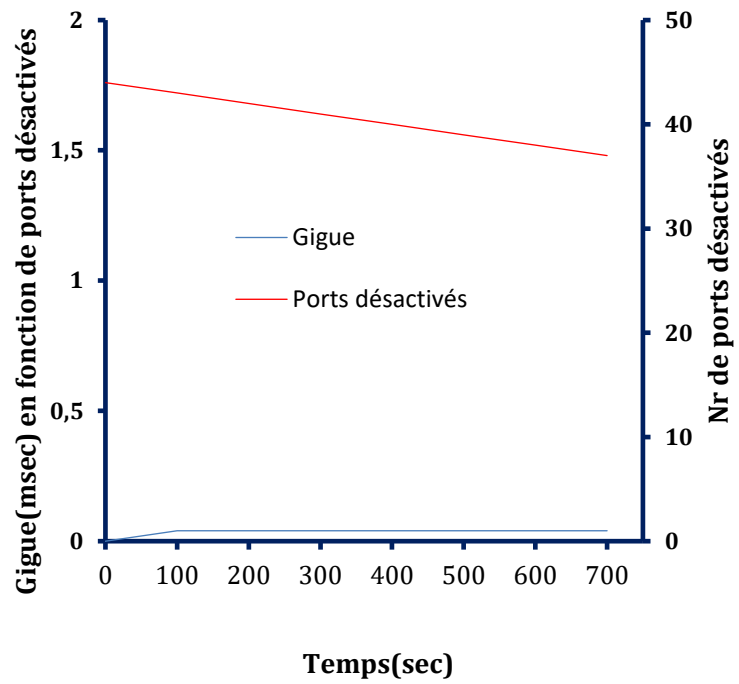


Figure 5.7b Gigue avec ports désactivés Première expérience

La gigue dépend de la variation de délai. Lorsque le délai varie, la gigue aussi varie. Elle est de 0.08% dans les travaux des auteurs [1] (**Figure 5.7a**), dans les nôtre, elle est de 0.04 %, soit la moitié (**Figure 5.7b**).

Les résultats des simulations sont donnés. Il convient de faire la comparaison de notre modèle et ceux des auteurs cités. La section suivante évaluera la performance de notre approche.

5.3.2 Evaluation de la performance

Dans cette section, nous comparons nos résultats avec ceux des auteurs cités dans le mémoire.

Nous constatons que les résultats des auteurs [2] sont nettement mieux que les travaux des auteurs [1]. Nous rappelons que les auteurs de [1] ont travaillé dans les réseaux classiques. L'efficacité dans les travaux de [2] réside en l'importance de la technologie SDN. Le fait que le SDN ait une vue globale du réseau facilite la définition de règles efficaces. L'utilisation de l'algorithme SPRING avec SDN a permis aux auteurs [2] d'obtenir de meilleurs résultats. Notre proposition de solution, vue de par les différentes figures, améliore les deux précédents. Notre approche basée sur le mode de désactivation des ports, minimise l'énergie avec une bonne qualité de service (QoS) en termes de délai, perte de paquets et gigue. Nous avons obtenu de meilleurs délais de transmission des paquets avec moins de congestion minimisant ainsi les pertes. Tout comme les auteurs [1], nous pouvons aussi dire qu'il y'a eu tout de même la perte de paquets dans nos simulations. On ne peut pas minimiser sans dégrader la qualité de service. Elle (perte de paquet) est de 20% dans [2], [1] et 10% dans le nôtre. Nous trouvons aussi, un délai nettement meilleur que les précédents auteurs.

5.4. Conclusion

Dans cette section, nous avons montré à travers les différentes figures, l'efficacité de notre approche. Nous expliquons ici l'idée et la manière dont les auteurs ont mené leur expérience.

Nous avons déjà montré dans le chapitre précédent que notre proposition de contribution était d'améliorer les travaux des auteurs référencé en [1].

La topologie réseau des auteurs a comporté 44 routeurs. Seuls 7 routeurs étaient à chaque fois utilisés pour faire les tests, soit 7/44 (15%) d'économie d'énergie possible dans les liens.

Le constat est que les auteurs éteignaient et activaient progressivement les routeurs à chaque 100s. Cela a provoqué une dégradation du réseau en termes de délai. Les paquets mettent assez de temps pour trouver un chemin favorable. Il y'a perte de 20% de paquets et une variation de délai (gigue).

Nous utilisons le même nombre de routeur que les auteurs [1] pour faire nos expériences. Nous allons procéder à l'activation et désactivation progressive des ports à chaque 100s.

Nous constatons effectivement que la manière d'éteindre les routeurs ou ports suit une série chronologique. Cela se justifie à travers notre fonction affine.

Notre amélioration se situe au niveau de la qualité de service lorsqu' un routeur est éteint ou allumé et le maximum de ses ports désactivés.

Notre expérience nous montre qu'un routeur allumé avec certains de ses ports désactivés suivant notre approche, offre une meilleure qualité de service en termes de délai diminuant ainsi le taux de perte de paquet et gigue.

CONCLUSION GENERALE ET PERSPECTIVES

L'objectif de notre travail de recherche était de trouver, d'abord, un modèle d'optimisation de la consommation d'énergie dans les réseaux communications, notamment dans les réseaux SDN. Ensuite, trouver aussi une nouvelle stratégie d'économie d'énergie et, enfin, simuler la fonction objectif trouvée.

Les propositions de solution à ces différentes problématiques sont nécessaires voire primordiales compte tenu de l'augmentation du trafic IP actuel et du nombre d'objets connectés. Nos différents constats montrent que dans les années à venir, nous assisterons à un trafic dense dans le réseau. Face à ce danger, un autre mode de vie et de pratique doit être cultivé dans l'utilisation des équipements réseaux. Les réseaux SDN devront donc permettre d'atteindre cet objectif majeur afin d'offrir une meilleure QoS dans la recherche de la réduction de leur consommation énergétique.

Nous avons fait un état ou revue bibliographique des approches d'optimisation de la consommation énergétique dans les réseaux tout en insistant sur les approches utilisées dans les réseaux classiques ou traditionnels. Notre constat est que, même si le réseau classique a connu du succès et facilité les transmissions des paquets ou informations dans l'utilisation de l'internet avec l'utilisation de certains protocoles tels que OSPF, ICMP etc., il aura du mal à répondre aux défis actuels en termes de délai de transmission, de contrôle de congestion, de perte de paquets et de gigue, compte tenu des limites que nous avons mentionné dans le chapitre 2. Car la configuration des équipements est faite en parcourant équipement par équipement. Les deux plans (contrôle et données) des routeurs étant intégrés, difficile de les programmer. D'un côté, la gestion du réseau en fonction des demandes, dégrade la qualité de service (QoS) provoquant ainsi un délai long et des pertes d'informations ou de paquets.

Après avoir présenté l'état de l'art des techniques ou approches d'Optimisation de la Consommation Energétique (OCE) dans les réseaux classiques, nous avons défini l'importance des SDN (réseaux de nouvelles générations) et son rôle actuel dans tous les domaines technologiques.

Dans le cadre de ce travail de thèse, nous nous sommes focalisés sur les équipements qui font consommer le réseau par leur extinction lorsqu'ils ne sont pas utilisés et de façon dynamique à chaque demande pour atteindre nos objectifs. Notre première publication est le fruit d'une collaboration. Ce travail a consisté à utiliser la technologie SDWN pour réduire la consommation de l'énergie en utilisant des fonctions virtuelles pour agir sur les équipements afin de mettre en veille ceux qui ne sont pas utilisés. Dans ces travaux, les auteurs ont utilisé une technologie qui sépare l'architecture physique de celle qui concerne l'intelligence dans les stations de base (Architecture logique ou virtuelle). Et ce, afin d'optimiser la consommation énergétique en réduisant la puissance du contrôleur, dans le fronthaul et celle des équipements. Cette nouvelle architecture facilitera les communications tout en évitant les interférences. L'approche des auteurs a permis de conclure que lorsque le rayon diminue (proche de la station de base) la puissance consommée par les utilisateurs diminue.

Pour notre première contribution, nous avons proposé un modèle mathématique et une nouvelle stratégie d'économie d'énergie comme apport. Nous avons estimé qu'un routeur ne doit pas être éteint et donc ce sont les interfaces de communications notamment ses ports qu'il faut éteindre lorsque ceux-ci ne sont pas utilisés de façon aléatoire et dynamique. Nous avons supposé que chaque port a le même taux de consommation d'énergie sur un routeur.

Notre fonction Gain ($g_i(t, \ell(t))$) obtenue par la maximisation du nombre de ports désactivés a donné de meilleurs résultats : 87.5% d'économie d'énergie, lorsque les ports sont éteints et 33.33% d'économie d'énergie, lorsque les liens sont éteints. Sachant que le routeur par exemple Cisco 3560-E donne 0.32w comme puissance d'un port, nous avons calculé la réduction de la consommation énergétique qui donne $0.875 * 0.32 = 28W$ et celle des liens est de $33.33\% * 0.64 = 21.33W$.

Notre approche et stratégie d'économie énergétique est efficace puisqu'elle permet une double économie (ports et liens).

La deuxième contribution est une suite de la première qui a approximé notre fonction "gain d'énergie" à une fonction exponentielle en proposant une approche algorithmique. Les résultats sont les mêmes que dans la fonction affine.

Quant à la troisième contribution, elle a consisté à montrer l'importance du délai dans la consommation de l'énergie dans les réseaux. Plus le délai augmente, plus il y'a consommation énergétique.

Nous avons donc déterminé la fonction délai en fonction du nombre de ports désactivés selon notre modèle et stratégie nouvelle. Notre constat est que les simulations nous ont permis de conclure que notre modèle d'économie de l'énergie offre de meilleurs résultats que dans la littérature avec une qualité de service intéressante en terme de délai, qui tend vers zéro, perte de paquets de 10 % et gigue qui est de l'ordre de 4%. Les paquets ont un port de sortie sur chaque routeur à chaque demande. Ce qui réduit le délai de transmission. Ceci améliore donc les travaux des auteurs [1]. Ce qui nous permet de dire que notre hypothèse d'extinction des ports est une solution d'économie d'énergie.

Nous avons pu proposer un modèle mathématique et une nouvelle stratégie d'économie d'énergie comme contribution. Cependant, des imperfections peuvent être constatées puisque toute œuvre humaine n'est pas parfaite.

Au total, dans la littérature, certains auteurs tels que [1, 2] ont réduit l'énergie dans les liens. Dans notre approche, nous avons montré qu'en plus des liens, il est aussi possible de minimiser l'énergie dans les ports.

Comme perspectives, nous avons cité plusieurs approches qui ont traité la problématique de la consommation de l'énergie.

Sur la base de notre modèle et stratégie d'économie d'énergie, nous pouvons orienter nos réflexions sur la mise en place de certaines approches ayant plusieurs critères par la technologie SDN, d'où une proposition d'idée de consommation d'énergie multicritère dans les réseaux SDN.

Dans la littérature, nous avons noté un certain nombre de contraintes parmi lesquelles la capacité des liens à prendre en compte à chaque demande et l'espace des

règles à prendre en compte également selon le dimensionnement du réseau. Pour cela, nous proposons une réflexion s'appuyant sur une approche intelligente tenant compte d'espaces de règles (limitées ou illimitées) selon le réseau (l'interopérabilité des règles).

Dans la continuité de nos travaux, nous pouvons prévoir un modèle d'activation de deux ports qui fonctionneraient de façon alternée en fonction de la charge du trafic dans le réseau et dynamiquement. A chaque allumage du routeur, deux ports sont activés et les autres ports sont éteints.

Le SDN comme décrit dans la Thèse, présente trois (03) couches : Application - Contrôleur et Plan de données. Nous avons orienté nos travaux sur l'optimisation de la consommation de l'énergie au niveau du plan de données. Cependant, dans des travaux futurs, en plus du plan de données, nous pouvons proposer un modèle d'optimisation de la consommation d'énergie à la fois dans le contrôleur et dans le plan de données, en évaluer la puissance totale consommée afin de la réduire.

Références Bibliographiques

- [1] **Gelenbe, E. and Silvestri, S.** (2009). Optimisation of power consumption in wired packet networks. In Bartolini, N., Nikolettseas, S., Sinha, P., Cardellini, V., and Mahanti, A., editors, *Quality of Service in Heterogeneous Networks*, volume 22 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 717–729. Springer Berlin Heidelberg.
- [2] **Radu Carpa, Olivier Glück, Laurent Lefèvre, Jean-Christophe Mignot,** (2015) "STREETE : Traffic engineering for energy-efficient core networks", Conference Compas'2015, Jun2015 Lille, France.
- [3] **Peng M., Li Y., Zhao Z., & Wang, C.** (2015). System architecture and key technologies for 5G heterogeneous cloud radio access networks. *IEEE network*, 29(2), 6-14.
- [4] **Feamster N., Rexford J. et Zegura E.** (2014) "*The Road to SDN: An Intellectual History of Programmable Networks*", *ACM SIGCOMM Computer Communication Review*, 44(2), pp. 87–98
- [5] **Chiaraviglio L., Leonardi E., Mellia M.** (2009) "How much can internet be greened ?" in *Green Comon*, 2009.
- [6] **Nordman Bruce and Ken Christensen,** (2005) "Reducing the energy consumption networked devices", *IEEE 802.3 Tutorial*, July 19 (San Francisco).
- [7] **Giroire, F., Moulhierac, J. and Phan, T.K.** (2014) Optimizing Rule Placement in Software Defined Networks for Energy-Aware Routing. 2014 IEEE Global Communications Conference, Austin, TX, 8-12 December 2014, 2523-2529. <https://doi.org/10.1109/GLOCOM.2014.7037187>
- [8] **Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Eredal Cayirci.** (2002) *Wireless sensor networks : a survey*. *Computer Networks*, 38(4) :393 - 422.
- [9] **Project,R.T.S.**(2014) **Internetlivestats.**

- [10] **Global Mobile Data Traffic Forecast Update, 2015-2020'** ,Cisco Visual Networking Index, White Paper, Cisco, [En ligne].
- [11] **GeSI** (2008). Smart 2020 : Enabling the low carbon economy in the information
- [12] **Bolla, R., Davoli, F., Bruschi, R., Christensen, K., Cucchietti, F., and Singh, S.** (2011c). The potential impact of green technologies in next-generation wireline networks: Is there room for energy saving optimization ? *Communications Magazine, IEEE*, 49(8) :80–86.
- [13] **Zamble R., Kra L., Koffi M., Gbegbe R., Goore B.T., and Asseu O.,** (2018). «Power Consumption Optimization in 5 G Radio Access Network, based on a call Admission Control Algorithm». *Far East Journal Electronics and Communications* Vol. 18, n° .4, pages 507-520.
<http://dx.doi.org/10.17654/EC01804050>
- [14] **Havet, F. Nicolas Huin, Joanna Moulhierac, Khoa Phan.** Routage vert et compressi de règles SDN. *ALGOTEL 2015-17èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications*, juin 2015,Beaume, France.
<hal01148471>
- [15] **Mohamed Said Seddiki.** Allocation dynamique des ressources et gestion de la qualité de service dans la virtualisation des réseaux. *Réseaux et télécommunications [cs.NI]. Université de Lorraine*, 2015. Français.
<tel-01242730>
- [16] **Ibrahima DIANE,** Optimisation de la consommation d'énergie par la prise en compte de la redondance de mesure dans les réseaux de capteurs, thèse soutenue 17/07/2014, *l'Université Toulouse 3 Paul Sabatier(UT3 Paul Sabatier)* Cotutelle internationale avec *l'Université Cheikh Anta Diop de Dakar.*
- [17] **Diery Ngom,** Optimisation de la durée de vie dans les réseaux de capteurs sans fil sous contraintes de couverture et de connectivité réseau. *Réseaux et télécommunications [cs.NI].Université de Haute Alsace-Mulhouse*, 2016.

Français.

- [18] **OUNIS Anis (2013)**,Optimisation de la capacité et de la consommation énergétique dans les réseaux maillés sans fil, thèse soutenue 12/12/2013- ISAL-0150.
- [19] **Stefan MILEV et al.**,28 mars 2016:« SDN avec GRID'5000
- [20] **Software-Defned Networking (SDN) Defnition**, [En ligne].
- [21] **SDN " Une nouvelle révolution dans la gestion des réseaux"**. 2015
- [22] **Wael Zouaoui**. « Synthèse de commande pour des réseaux de communication énergétiquement performants» : Thèse soutenue, 2016. UPS Toulouse - Université Paul Sabatier
- [23] **Larousse 2018** [en ligne]. Consulté le 02/02/2019.
- [24] **www.cnrtl.fr/définition/ Consommation, Consommation**: Définition de consommation, consulté le 02/02/2019 [En ligne].
- [25] **Open Flow Switch Consortium and Others (2014)**. Open Flow Switch Specification Version 1.5.0, [En ligne]
- [26] **Forwarding and Control Element Separation (ForCES) Forwarding Element Model**, RFC 3746, April, 2004. [En ligne]
- [27] **Lakshman T., Nandagopal T., Ramjee R., Sabnani K., and Woo T.** (2004). Te sofrouter architecture", *Workshop Hot Top Network*,ACM SIGCOMM.
- [28] **Rodriguez A., Natal S., Barkai V., Ermagan D., Lewis F., Maino, and Farinacci D.**(2014).“Software Defned Networking extensions for the Locator/IDSeparationProtocol”[En ligne]
- [29] **Contrail Open SDN NFV & Cloud Solutions (2016)**. Juniper Networks, [En ligne]
- [30] **BENAMRANE, Fouad (2017)**, Etude des performances des Architectures du plan de Contrôle des Réseaux 'Software- Defined Network', Université Mohammed V, Faculté des Sciences, Rabat.
- [31] **Giridhar A. and Kumar P.** (2005) “Maximizing the functional lifetime of sensor

- networks", in *proceedings of the 4th International Symposium on Information Proceeding in Sensor Network (IPSN)*, pp. 13-19
- [32] **Douglas M. Blough and Paolo Santi.**(2002.) Investigating upper bounds on network lifetime extension for cell-based energy conservation techniques in stationary ad hoc networks. In *Proceedings of the 8th annual international conference on Mobile Computing and networking (MobiCom'02)*, pages 183-192, New York, NY, USA, ACM.
- [33] **Santosh Kumar, Anish Arora, and Ten. H. Lai.** (2005) On the lifetime analysis of always-on wireless sensor network applications. In *Proceedings of the 2nd IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS'05)*, pages 186-188, Washington, DC, USA, November.
- [34] **Sameer Tilak, Nael B. Abu-Ghazaleh, and Wendi Heinzelman.** (2002) A taxonomy of wireless micro-sensor network models. *SIGMOBILE Mobile Computing and Communications Review*, 6(2) :28-36.
- [35] **Younis M. and Nadeem T.** (2004) "Energy efficient MAC protocols for wireless sensor Networks", in *Technical report, university of Mryland baltimre County, USA*
- [36] **Plancoulaine S., Bachir A., and Barthel D.** (2006). Wsn node energy dissipation Technical report, France Telecom R&D, internal report, july.
- [37] **Auer G., Blume O., and Giannini V.** (2010) : Energy efficient analysis of the reference systems, areas of improvements and target breakdown. INFSO-ICT-247733 EARTH (Energy Aware Radio and Network Technologies), Tech. Resp, 36, November.
- [38] **Chee D., Min K., Howon L., and Bang J.** (2011). A study on the green cellular network with femtocells. In *Third International Conferences on Ubiquitous and Future Network (ICUFN)*, pages 235-240, Dalian, China, June.
- [39] **LAGRANGE Xavier, GODELEWSKI P., et TABBANE Sami.** Réseaux GSM des principes à la norme. 2000.
- [40] **Copyrigt EFFORT 2014.** << Evolution des réseaux mobiles 4G EFFORT>>.

- [41] **Kullin C. and Ran D.** (2011). C-ran the road towards green ran, China Mobile Research Institute, White Paper.
- [42] **3GPP. Technical Specification 25.401, UTRAN overall description,** Dec.2008.
- [43] **Bouguen Y., Hardouin E., F-X. Wolff,** « LTE et les réseaux 4G », livre, 2012
- [44] **Jennifer Mertens** (2018), *"De la 2G à la 5G: Comment les réseaux mobiles ont influencé l'évolution technologique"*. Rapport Télécoms [en ligne]
- [45] **Mugen Peng, Yuling Yu, Hongyu Xiang and H. Vincent Poor.** (2016). Energy-efficient resource allocation optimization for multimedia heterogeneous cloud radio access networks, *IEEE Transactions on Multimedia* 18(5), 879-892
- [46] **Mugen Peng, Yuan Li, Jiamo Jiang, Jian Li and Chonggang Wang,** (2014). Heterogeneous cloud radio access networks: a new perspective for enhancing spectral and energy efficiencies, *IEEE Wireless Communications* 21(6), 126-135.
- [47] **Greenberg A., Hjalmtysson G., Maltz D.A., Myers A., Rexford J., Xie G., Yan H., Fabrizio J., Granelli, Anteneh A Gebremariam, Muhammad Usman, Filippo Cugini, Veroniki Stamati, Marios Alitska, and Periklis Chatzimisios.** (2015). Software defined and virtualized wireless access in future wireless networks: scenarios and standards. *IEEE Communications Magazine*, 53(6):26–34.
- [48] **Brandon, Heller, Srinivasan Seetharaman, Priya Mahadevan, Yiannis Yiakoumis, Casado M., Freedman M. J., Petit J., Luo J., McKeown N. et Shenker S.** (2007). "Ethane: Taking Control of the Enterprise", *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pp. 1–12.
- [49] **Tennenhouse D. et Wetherall D.** (1996). "Towards an active network architecture", *ACM Computer Communication Review*, 26(2), pp. 5–17.
- [50] **Schwartz B., Jackson A. W., Strayer W. T., Zhou W., Rockwell R. D., and Partridge C.** (1999). "Smart Packets for active networks", *IEEE Second Conference on Open Architectures and Network Programming Proceedings OPENARCH '99*, IEEE

- xplore, pp. 90–97.
- [51] **Wetherall D. J., Gutag J. V., and Tennenhouse D. L.** (1998). “ANTS: a toolkit for building and dynamically deploying network protocols”, *IEEE Open Architectures and Network Programming*, IEEE xplore, pp. 117–129
- [52] **Alexander D. S., W. A. Arbaugh, Hicks M. W., Kakkar P., Keromytis A. D., Moore J. T., Gunter C.A., Netles S. M., and Smith J. M.** (1998). “Te SwitchWare active network architecture”, *IEEE Network*, IEEE xplore, 12(3), pp. 29–36.
- [53] **Campbell A.T., De Meer H.G., Kounavis M.E., Miki K., Vicente B.J. et Villela D.** (1999). “A survey of Programmable Networks”, *SIGCOMM Computer Communication Review*, 29(2), pp.7–23
- [54] **Devolved Control of ATM Networks**, [En ligne]
- [55] **Campbell A.T., Katzela I., Miki K. et Vicente J.** (1999). “Open Signaling for ATM, Internet and Mobile Networks (OPENSIG’98)”, *SIGCOMM Computer Communication Review*, 29(1), pp.97– 108.
- [56] **Gupta M., Grover S., and Singh S.** (2004) A feasibility study for power management in lan switches. In *Network Protocols, 2004. ICNP 2004. Proceedings of the 12th IEEE International Conference on*, pages 361–371, Oct.
- [57] **Gude N., Koponen T., Petit J., Pfaff B., Casado M., McKeown N. et S. Shenker** (2008) . “NOX: Towards an Operating System for Networks”, *ACM SIGCOMM Computer Communication Review*, 38(3), pp. 105–110.
- [58] **Ericsson mobility report (2016)**. [En ligne]
- [59] **Seiffeddine Ben Chahed**, (2015). Mise en œuvre des aspects de Gestion des réseaux définis par logiciels (SDN).p 6;46-48
- [60] **Jérôme Durand**. Cisco system, 11 rue camille Desmoulins 92130 Issy- les Moulinaux. **SDN pour les nuls**. JRES 2015 – Montpellier.
- [61] **AMD and Hewlett Packard** (1995). Next generation connectivity solutions : Amd’s managed performance portfolio. <http://fr.wikipedia.org/wiki/Wake-on-LAN>

- [62] **Gupta, M. and Singh, S.** (2007b). Using low-power modes for energy conservation in Ethernet lans. In INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE, pages 2451–2455.
- [63] **Roberts, L.** (2009b). A radical new router. *Spectrum, IEEE*, 46(7) :34–39.
- [64] **Ananthanarayanan, G. and Katz, R.H.** (2008). Greening the switch. In Proceedings of the 2008 Conference on Power Aware Computing and Systems, HotPower'08, pages 7–7, Berkeley, CA, USA. USENIX Association.
- [65] **Gupta, M. and Singh, S.** (2003). Greening of the internet. In Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, SIGCOMM 03, pages 19–26, New York, NY, USA. ACM.
- [66] **Amaldi, E., Capone, A., and Gianoli, L.** (2013). Energy-aware {IP} traffic engineering with shortest path routing. *Computer Networks*,57(6):1503– 1517
- [67] **Chabarek J., Sommers J., Barford P., Estan C., Tsang D., and Wright S.** (2008) Power awareness in network design and routing. In *IEEE INFOCOM*, pages 457–465.
- [68] **Puype B., Vereecken W., Colle D., Pickavet M., and Demeester P.** (2009). Power Reduction Techniques in Multilayer Traffic Engineering. In ICTON.
- [69] **Frederic Giroire, Dorian Mazauric, Joanna Moulhierac.** (2011). Routage efficace et énergie. Du courthial, Bertrand et Felber, Pascal. 13es Rencontres Francophones sur les Aspects Algorithmiques de Télécommunications (AlgoTel), 2011,Cap Esterel, France.
- [70] **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*
- [71] **Richard, Brown.** (2008). « Report to congress on server and data center energy efficiency : Public law 109-431 ». *Lawrence Berkeley National Laboratory*.

- [72] **Hiroki, Shirayanagi, Hiroshi Yamada et Kono Kenji.** (2013). « Honey guide : 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.
- [73] **Gupta M., Grover S., and Singh S.** (2004). A feasibility study for power management in lan switches. In *Network Protocols, 2004. ICNP 2004. Proceedings of the 12th IEEE International Conference on*, pages 361–371, Oct.
- [74] **Jin, S., Fan, R., and Yue, W.** (2012). A hybrid energy saving strategy with lpi and t for energy- efficient ethernet. In *Computer Science and Network Technology (ICCSNT), 2012 2nd International Conference on*, pages 311–315.
- [75] **Yongyu Dai, Shi Jin, Leyuan Pan, Xiqi Gao, Lei Jiang and Ming Lei** (2015), Interference control based on beamforming coordination for heterogeneous network with RRH deployment, *IEEE Systems J.* 9(1), 58-64
- [76] **Ying Loong Lee, Li-Chun Wang, Teong Chee Chuah and Jonathan Loo** (2016) Joint resource allocation and user association for heterogeneous cloud radio access networks, 2016 28th International Tele traffic Congress (ITC 28), IEEE, pp. 87-93.
- [77] **Karen Boulos, Melhem El Helou and Samer Lahoud** (2015) RRH clustering in cloud radio access networks, 2015 International Conference on Applied Research in Computer Science and Engineering (ICAR), IEEE, pp. 1-6
- [78] **Peng M., Li Y., Jiang J., Li J., & Wang C.** (2014). Heterogeneous cloud radio access networks: A new perspective for enhancing spectral and energy efficiencies. *IEEE Wireless Communications*, 21(6), pp. 126-135.
- [79] **Shinobu Namba, Takayuki Warabino and Shoji Kaneko** (2012) BBU-RRH switching schemes for centralized RAN, 2012 7th International ICST Conference on Communications and Networking in China (CHINACOM), IEEE, pp. 762-766.
- [80] **Botero, J., Hesselbach, X., Duelli, M., Schlosser, D., Fischer, A., and de Meer,**

- H.(2012). Energy efficient virtual network embedding. *Communications Letters, IEEE*, 16(5) : 756–759.
- [81] **Yossi, CKanizo, David Hay, et Isaac Keslassy.** (2013). « Palette : Distributing tables in softwaredefined networks ». In *INFOCOM, 2013 Proceedings IEEE*, p. 545-549
- [82] **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.
- [83] **Zemmouri, Samy** (2017), Modèle de routage éco-énergétique dans les réseaux définis par logiciel. Mémoire de Maitrise électronique, Montréal, Ecole de Technologie Supérieure
- [84] **Sameer Tilak, Nael B. Abu-Ghazaleh, and Wendi Heinzelman.** (2002) A taxonomy of wireless micro-sensor network models. *SIGMOBILE Mobile Computing and Communications Review*, 6(2) :28-36.
- [85] **Myriana Rifai, Nicolas Huin, Christelle Caillouet, Frédéric Giroire, Joanna Moulierac, et al..** MINNIE : enfin un monde SDN sans (trop de) règles. *ALGOTEL 2016 - 18èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications*, May 2016, Bayonne, France
- [86] **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
- [87] **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.
- [88] **Reviriego P., Sivaraman V., Zhao Z., Maestro A., Vishwanath J., Sanchez A., Macian A. and Russel, C.** (2012) An Energy Consumption Model for Energy Efficient Ethernet Switches. 2012 International Conference on High Performance

Computing & Simulation (HPCS), Madrid, 2-6 July 2012, 98-104
<https://doi.org/10.1109/HPCSim.2012.6266897>

- [89] **Pickavet M., Vereecken W., Demeyer S., Audenaert P., Vermeulen B., Develder, C., Colle D., Dhoedt B., and Demeester, P.** (2008). Worldwide energy needs for ict :The rise of power-aware networking. In *Advanced Networks and Telecom-munication Systems, 2008. ANTS '08. 2nd International Symposium on*, pages 1–3.
- [90] **Doherty, J.** (2016) *SDN and NFV Simplified: A Visual Guide to Understanding Software Defined Networks and Network Function Virtualization* Addison-Wesley Professional, Boston
- [91] **Définition de Myrinet [en Ligne], visité le 06/12/2018**
- [92] **Timothée Haudebourg**, Analysis of the main techniques of energy saving in Ethernet Wired network, University of Rennes, June 2015.
- [93] **Cisco, The Zettabyte era.** (2015). : Trends and analysis, White paper, May.
- [94] **Kra, L., Gondo, Y., Gooré, B.T. and Asseu, O.** (2018) Contribution to the Optimization of the Energy Consumption in SDN Networks. *Journal of Sensor Technology*, 8, 1-9.
<https://doi.org/10.4236/jst.2018.83005>
- [95] **Cisco Catalyst 3560-E Series Switches :** All Contents are Copyright © 1992-2007, 2009 Cisco Systems, Inc. All rights reserved. This document is Cisco Public Information.
- [96] **A Quick Overview of the OMNET++ 4.0 IDE04**
- [97] **Azadeh Abdolrazaghi**, (2009). "Unifying Wirless Sensor Network Simulators", Master's Degree Project Stockholm, Sweden.

ANNEXES

ANNEXE A : Le Simulateur OMNET++

Le choix de simulateur OMNET++

Nous citons quelques simulateurs réseaux : NS2, GloMoSim et OMNET++. Chacun des simulateurs a ses spécificités. Nous optons pour OMNET++ parce qu'il est utilisé dans les réseaux SDN pour traiter les problèmes de modélisation. L'importance de la simulation est de tester à moindre coût la performance d'une solution. Nous l'avons utilisé dans notre modélisation (les captures sont montrées à la suite de nos différents algorithmes).

OMNET++ est un environnement de simulation à événement discret basé sur le langage c++. Il est en Open source sous licence GNU [96]. Totalement programmable, paramétrable et modulaire, OMNET++ est une architecture flexible et générique. Il a été utilisé dans de nombreux domaines pour :

- La modélisation de réseaux sans fils d'attente ;
- La modélisation de protocole de communication ;
- La validation des architectures hardware ;
- L'évaluation de la performance pour les systèmes software complexes.

Nous avons utilisé ce simulateur grâce à son architecture en implémentant un modèle spécifique de routeurs en testant les fonctions mathématiques.

✓ **Présentation du Simulateur OMNET++**

OMNET++ est un environnement de simulation à événement discret utilisé pour la simulation de réseaux de communication et d'autres systèmes distribués [97]. Grâce à son architecture modulaire, OMNET++ est très largement répandu dans divers domaines déjà cités haut. Il est basé sur la plateforme Eclipse et fournit pour la création, la configuration des modèles de réseaux (fichiers Ned et INI) et des outils d'exécution ainsi que celui de l'analyse des résultats.

Environnement Matériel

Processeur	INTEL® Core™ i3
Mémoire	4Go
Disque dur	500Go

Environnement logiciel

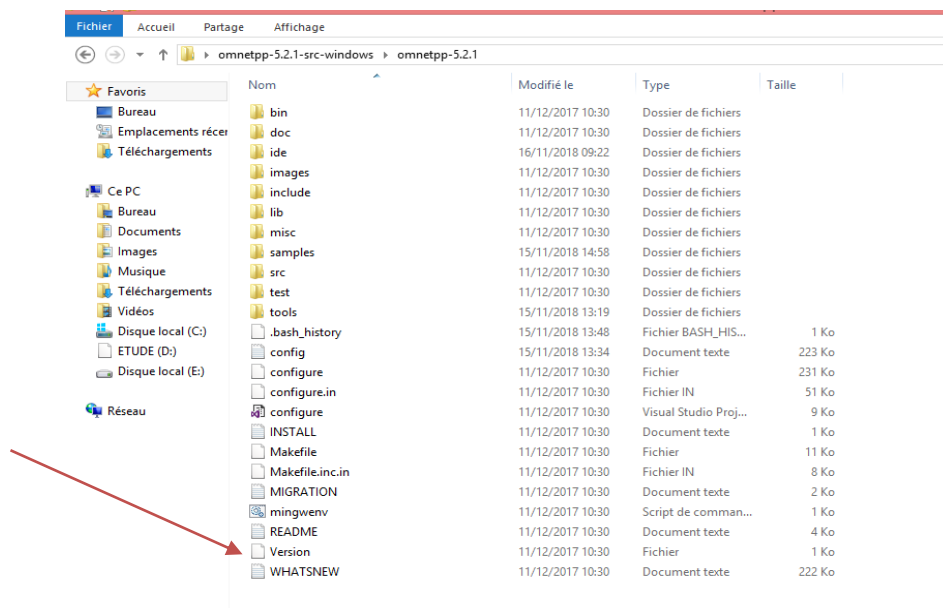
Nous avons travaillé dans un environnement logiciel suivant:

Système d'exploitation : Windows 8.1 Entreprise

Simulateur : OMNET++ 5.2.1

ANNEXE B : Installation OMNET++ 5.2.1

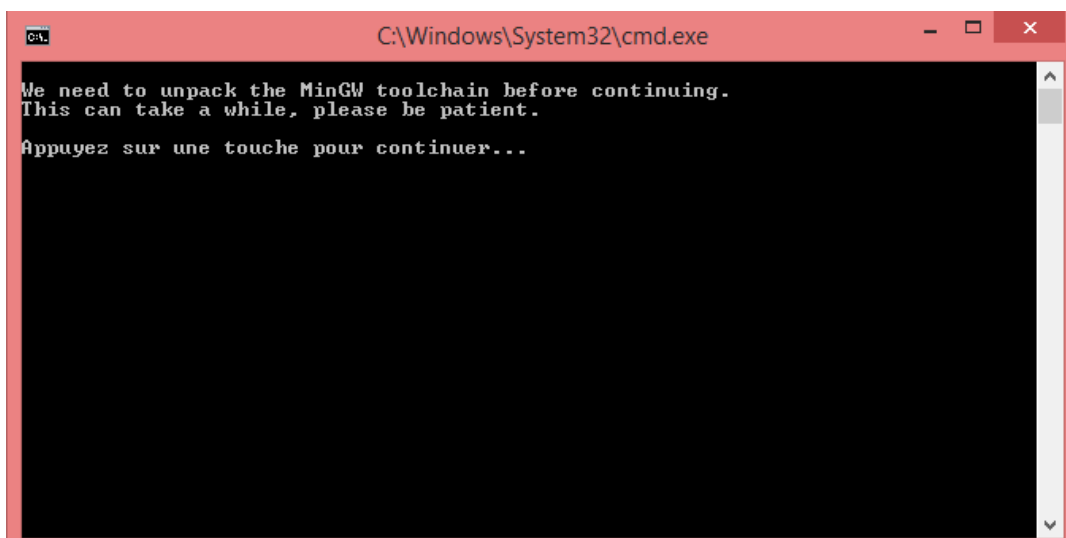
1)



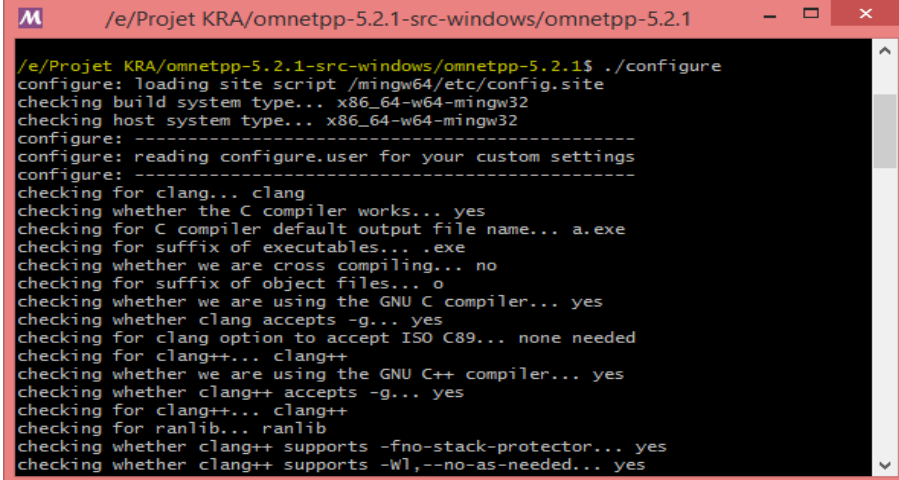
Le <<mingwenv>> permet de faire les différentes configurations des différents packages des Framework.

Double cliquer pour visualiser pour voir les étapes suivantes.

2)

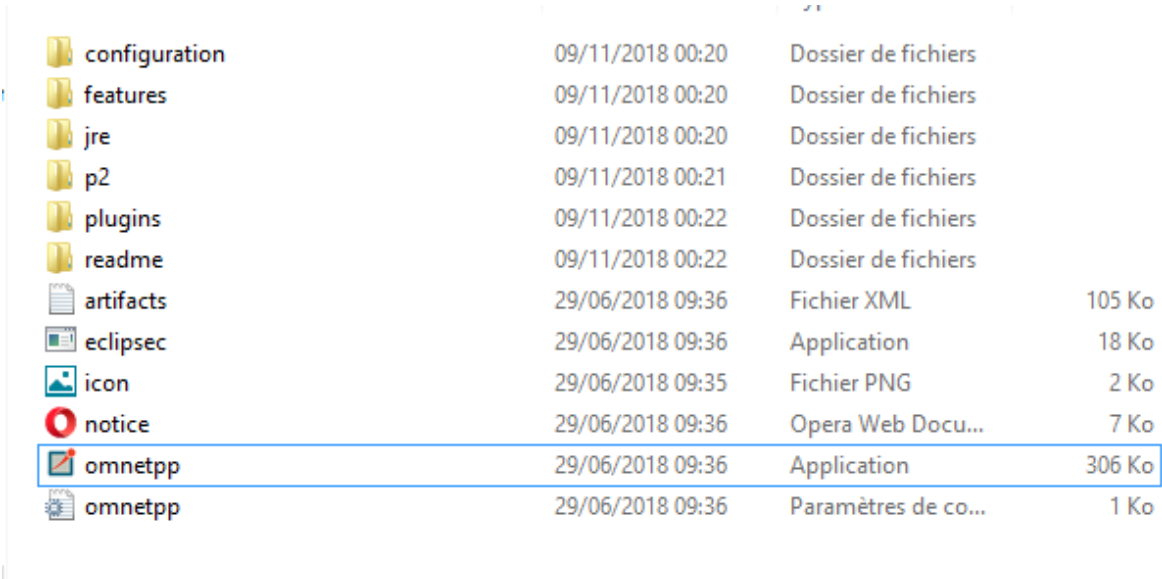


3)



```
/e/Projet KRA/omnetpp-5.2.1-src-windows/omnetpp-5.2.1$ ./configure
configure: loading site script /mingw64/etc/config.site
checking build system type... x86_64-w64-mingw32
checking host system type... x86_64-w64-mingw32
configure: -----
configure: reading configure.user for your custom settings
configure: -----
checking for clang... clang
checking whether the C compiler works... yes
checking for C compiler default output file name... a.exe
checking for suffix of executables... .exe
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether clang accepts -g... yes
checking for clang option to accept ISO C89... none needed
checking for clang++... clang++
checking whether we are using the GNU C++ compiler... yes
checking whether clang++ accepts -g... yes
checking for clang++... clang++
checking for ranlib... ranlib
checking whether clang++ supports -fno-stack-protector... yes
checking whether clang++ supports -Wl,--no-as-needed... yes
```

Cliquer sur **ide** dans le dossier (voir figure ci-dessous).



configuration	09/11/2018 00:20	Dossier de fichiers	
features	09/11/2018 00:20	Dossier de fichiers	
jre	09/11/2018 00:20	Dossier de fichiers	
p2	09/11/2018 00:21	Dossier de fichiers	
plugins	09/11/2018 00:22	Dossier de fichiers	
readme	09/11/2018 00:22	Dossier de fichiers	
artifacts	29/06/2018 09:36	Fichier XML	105 Ko
eclipsec	29/06/2018 09:36	Application	18 Ko
icon	29/06/2018 09:35	Fichier PNG	2 Ko
notice	29/06/2018 09:36	Opera Web Docu...	7 Ko
omnetpp	29/06/2018 09:36	Application	306 Ko
omnetpp	29/06/2018 09:36	Paramètres de co...	1 Ko



Interface graphique



ANNEXE C : Code et les différents algorithmes

Le choix de OMNET++ a été déjà montré et donc nous présentons dans cette partie le code source ayant permis de construire notre réseau ainsi que les autres algorithmes de désactivation de liens et de stratégie d'économie d'énergie. Des captures vérifiant notre fonction objectif sont aussi présentées dans cette partie.

Notre stratégie consiste à trouver dans un réseau l'arbre maximal qui minimise ayant le plus petit poids. Cet arbre qui est un chemin doit éviter les cycles empêchant les paquets de se promener dans le réseau. Une voie est trouvée sur chaque routeur de façon alternative vu que le taux de consommation d'un port est le même (voir hypothèse).

Nous rappelons que notre principe a utilisé 7 routeurs comme dans le cas des travaux des auteurs [1].

Les fichiers ayant permis d'atteindre cet objectif sont entre autres, le fichier. NED. Notre réseau a été construit grâce à ce fichier pour modéliser les "routeurs". Le fichier. NED est un fichier graphique «. Ned" (voir figure ci-dessous)

Partie A : Code permettant de créer les routeurs sous OMNET++

```
package org.omnetpp.queueing.routage7;
simple Router
{
    parameters:
        @group (Queueing);
        @display ("i=block/routing");
        string routingAlgorithm
@enum ("random", "roundRobin", "shortestQueue", "minDelay") = default ("random");
        volatile int randomGateIndex = default (intuniform (0, sizeof (out) -
1));
        // the destination gate in case of random routing

    gates:
        inout gate[];
}
//
//
network Network
{
    @display ("bgb=512,314;bgl=2");
    types:
```

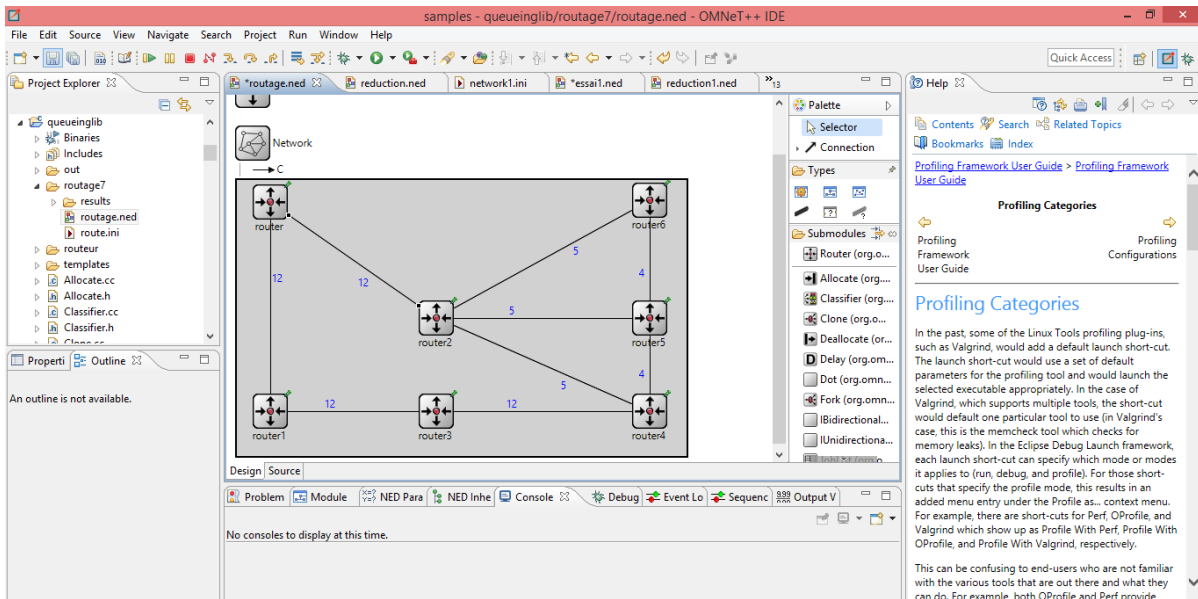
```

channel C extends ned.DatarateChannel
{
    delay = 0;
    datarate = 100Mbps;
}

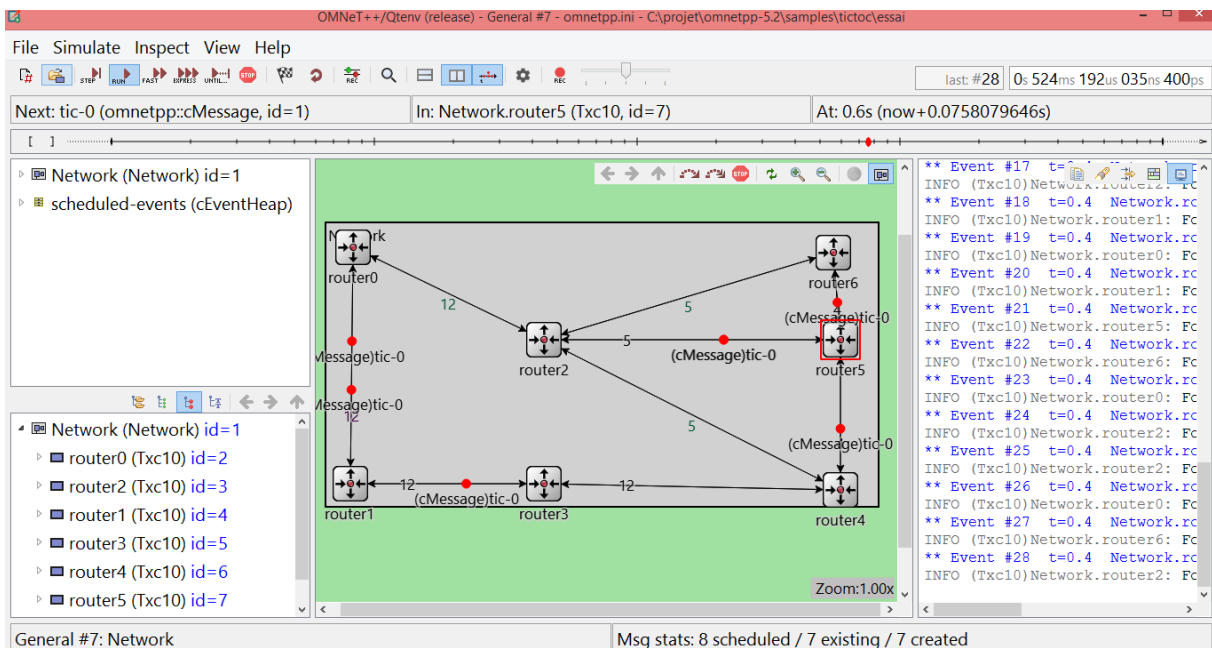
submodules:
router: Router {
    @display("p=38,25");
}
router1: Router {
    @display("p=38,263");
}
router2: Router {
    @display("p=227,157");
}
router3: Router {
    @display("p=227,263");
}
router4: Router {
    @display("p=470,263");
}
router5: Router {
    @display("p=470,157");
}
router6: Router {
    @display("p=470,23");
}

connections:
router.gate++ <--> { delay = 100ms; } <--> router1.gate++;
router1.gate++ <--> { delay = 100ms; } <--> router4.gate++;
router1.gate++ <--> { delay = 100ms; } <--> router3.gate++;
router3.gate++ <--> { delay = 100ms; } <--> router1.gate++;
router3.gate++ <--> { delay = 100ms; } <--> router4.gate++;
router4.gate++ <--> { delay = 100ms; } <--> router3.gate++;
router4.gate++ <--> { delay = 100ms; } <--> router5.gate++;
router5.gate++ <--> { delay = 100ms; } <--> router4.gate++;
router5.gate++ <--> { delay = 100ms; } <--> router2.gate++;
router2.gate++ <--> { delay = 100ms; } <--> router5.gate++;
router5.gate++ <--> { delay = 100ms; } <--> router6.gate++;
router6.gate++ <--> { delay = 100ms; } <--> router5.gate++;
router4.gate++ <--> { delay = 100ms; } <-->router2.gate++;
router2.gate++ <--> { delay = 100ms; } <-->router4.gate++;
router2.gate++ <--> { delay = 100ms; } <-->router.gate++;
router.gate++ <-->{ delay = 100ms; } <--> router2.gate++;
router6.gate++ <--> { delay = 100ms; } <-->router2.gate++;
router2.gate++ <--> { delay = 100ms; } <-->router6.gate++;
}

```

Circulation des paquets ou informations dans le réseau tout en évitant les cycles.



L'image est ainsi obtenue à partir de l'algorithme de Kruskal:

Partie B : Algorithme de Kruskal pour obtenir l'arbre maximal de plus faible poids.

Principe de l'algorithme

L'algorithme de Kruskal consiste à classer par ordre croissant poids des arrêtes du graphe. Le choix des arrêtes se fait suivant l'ordre du classement (croissant) à ajouter sur l'arbre recherché. Cherché tant que cet ajout ne crée pas de circuit dans le graphe.

*** Déclaration des variables***

```
#include <iostream>
#define MAX 30
int N;
float Delais;
float TD[100];
float delaiN=0;
float Nombre_port_ettiend=0;
float Dlt=0;
float Nombre_router_etteind=0;
float cont=0;
float gainrecup=0;
float gain_links_reup=0;
int nombre_router;
int nombre_links;
int nombre_port;
float links=0;
float links_off;
float Nbliinksr;
float nberT;
float link_off=0;
float gain_port=0;
float gain_links=0;
float Ci=0;

typedef struct edge
{
    int u,v,w;
}edge;
typedef struct edgelist
{
    edge data[MAX];
    int n;
}edgelist;

edgelist elist;

int G[MAX][MAX],n;
edgelist spanlist;

void kruskal();
int find(int belongs[MAX],int vertexno);
void union1(int belongs[MAX],int c1,int c2);
void sort();
void print();
void sort1();
main()
{
    /** remplacement de code source/**
    int i,j,total_cost;
    cout<<("\n\n Number of Router ")<<endl ;
    cin>>nombre_router;
    cout<<("\n\n Number Ports of Router ")<<endl;
    cin>>nombre_links;
    cout<< (" \n\n Number Links");
    cin>> (links);
    nberT=nombre_router*nombre_links;
    cout<< (" \n\n Number of Links: ");
```

```

cin>> (n);
//*****//
int i,j,total_cost;

```

Saisie des variables d'entrées

```

cout<<("\ Number Router:")<<endl;
cin>>(Number_router);
cout<<("\ Number Ports of Router:")<<endl;
cin>>(Number_port_on);
cout<<("\n Entrer le nombre total de links dans le resaux:")<<endl;
cin>>(NlinkT);
nber_t= Number_port_on*Number_router;

//*****//
// ***( "\nle nombre total port off %d\n",port_off);
cout<<("\ Number of Links:")<<endl;
cin>>(n);

cout<<("\n Enter the adjacency matrix:\n")<<endl;
for(i=0;i<=n-1;i++)

```

// Entrer la matrice d'adjacence à partir du réseau construit//
// Il s'agira de parcourir tout les sommets ou routeur du réseau en entrant les arêtes et leurs poids//

```

    for(j=0;j<=n-1;j++)
    {
        cin>>(G[i][j]);
    }
}
int kruskalMST();
}
void kruskal()
{
    int belongs[MAX],i,j,cno1,cno2;
    elist.n=0;

    for(i=1;i<n;i++)
    {
        for(j=0;j<i;j++)
        {
            if(G[i][j]!=0)
            {
                elist.data[elist.n].u=i;
                elist.data[elist.n].v=j;
                elist.data[elist.n].w=G[i][j];
                elist.n++;
            }
        }
    }

    sort();
    for(i=0;i<n;i++)
        belongs[i]=i;

    spanlist.n=0;

```

```

for(i=0;i<elist.n;i++)
{
    cno1=find(belongs,elist.data[i].u);
    cno2=find(belongs,elist.data[i].v);

    if(cno1!=cno2)
    {
        spanlist.data[spanlist.n]=elist.data[i];
        spanlist.n=spanlist.n+1;
        union1(belongs,cno1,cno2);
    }
}

int find(int belongs[],int vertexno)
{
    return(belongs[vertexno]);
}
void union1(int belongs[],int c1,int c2)
{
    int i;

    for(i=0;i<n;i++)
        if(belongs[i]==c2)
            belongs[i]=c1;
}

void sort()
{
    int i,j;
    edge temp;

    for(i=1;i<elist.n;i++)
        for(j=0;j<elist.n-1;j++)
            if(elist.data[j].w>elist.data[j+1].w)
            {
                temp=elist.data[j];
                elist.data[j]=elist.data[j+1];
                elist.data[j+1]=temp;
            }
}

void print()
{
    int Nblinksetteind;
    int Ac=0;
    int Nblilinksr=0;
    int Nbportactif;
    int Nbrouter;
    int NbTrouter;

    int i,cost=0;
    //int col = 0;
    cout<<"voici la liste des links retenus \n"<<endl;
    for(i=0;i<spanlist.n;i++)
    {
        cout<<(spanlist.data[i].u<<spanlist.data[i].v<<spanlist.data[i].w)<<endl;
        cost=cost+spanlist.data[i]. w ;
    }
    // Affichage de l'arbre couvrant de poids minimum//
}

```

```

Number of Router 7
Number Ports of Router 8
Number of Links9
enter the adjacency: 7
Enter the adjacency matrix:
12 0 12 12 0 12 12
12 0 4 12 0 12 12
0 12 12 0 12 4 12
12 0 5 12 0 12 4
12 0 12 0 5 4 12
12 4 12 0 12 5 12
12 0 4 12 0 12 4

5          1          4
6          2          4
3          2          5
1          0          12
2          1          12
4          0          12

Maximum tree with the lowest weight: 49

```

(Temps d'exécution : 146.976s)
// valeur la plus importante pour la suite//
// récupérer le nombre de lien éteint//

Notre stratégie d'économie tourne autour de cette fonction en rouge. Elle récupère le nombre de ports désactivés et va au suivant

NbliInksr= NbliInksr++;

```

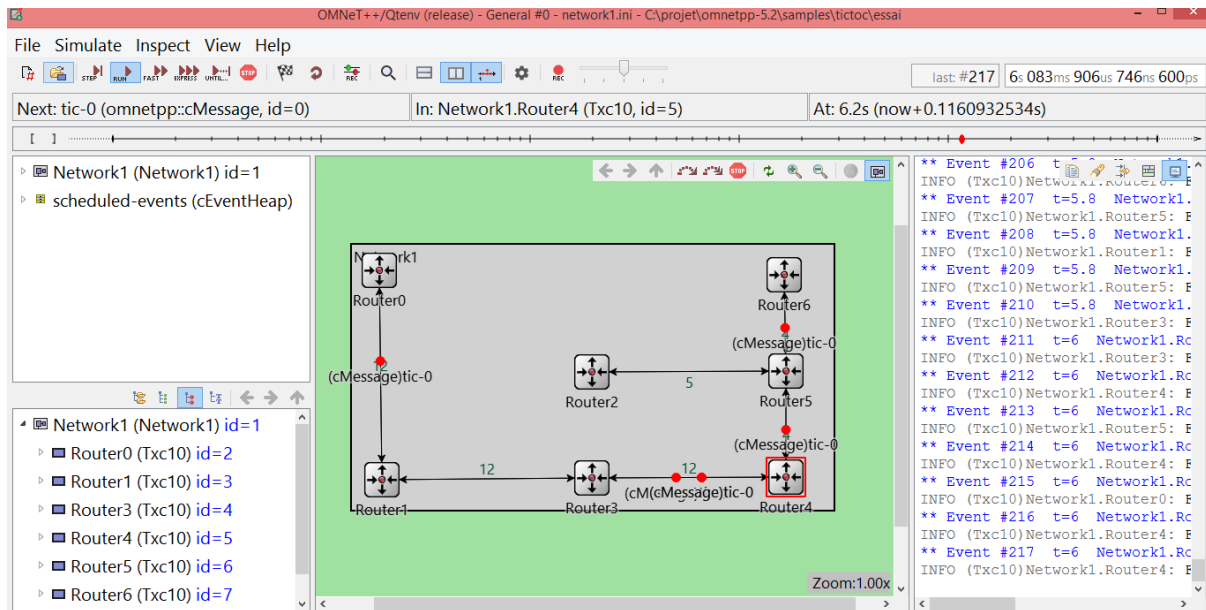
}
Nbportactif=NbliInksr*2;
Nbrouter=(NbliInksr+1);
Nblinksetteind=NlinkT - NbliInksr;
port_off1=(cost/nuber_t);
float gain;
int Nvport=0;

Ac=Nblinksetteind*2;
Nvport=Nbrouter*Number_port_on;
gain= cost/NbliInksr;

cout<<(" \n Nombre actif " )<<Nbportactif<<endl;
cout<<(" \n port Off:")<<port_off1<<endl;
cout<<("\n\n Number liens déjà etteind")<<Nblinksetteind<<endl;
cout<<("\n\n port OFF")<<cost<<endl;
cout<<("\n\n Nombre Links rester")<<NbliInksr<<endl;
cout<<("\n\n Nombre de routeur dans le reseau")<<Nbrouter<<endl;
cout<<("\n\n le nombre total de port dans le réseau est ")<<Nvport<<endl;
cout<<("\n\n le nombre de port desactiver en fonction du temps (ALt est ") <<Ac<<endl;
cout<<("\n\n gain:")<<gain<<endl;
cout<<("\n le nombre total port est:")<<number_t<<endl;
}

```

Voir le schéma suivant :



Algorithme de stratégie d'économie d'énergie

Une fois l'arbre maximum de plus faible poids est trouvé, le nombre de ports éteint et le nombre de liens déjà connu, on peut donc déterminer les pourcentages d'extinction possibles dans les ports ainsi que dans les liens.

Cost = Minimum Spanning Trees

NberT = Number of ports in the network

Gain = Ports off

Link_off = Number of Links

```
float Port_desactif=0;
```

```
float gain=0;
```

```
// calcul de pourcentage//
```

```
gain=(cost/nberT)*100;
```

```
link_off=links-Nblinksr;
```

```
links_off=(link_off/links)*100;
```

```
Port_desactif=link_off*2;
```

```
gainrecup=0;
```

```
gain_links_reup=(Port_desactif/links)*100;
```

```
//affichage des résultats//
cout<<("\n\n port OFF")<<cost<<endl;

cout<<("\n\n %PORTS OFF :)<<gain<<endl;

cout<<("\n\n NUMBER OF links: )<<links<<endl;

cout<<("\n\n LINKS OFF: <<link_off<<endl;

cout<<("\n\n % LINKS OFF : <<links_off<<endl;
```

```
//procédure de calcul de gain//
```

Cette procédure consiste à évaluer la consommation d'énergie des ports désactivés et liens désactivés connaissant la puissance du port.

```
cout("\n\n Entrer la consommation globale :")<<endl;

cin>>Ci;

gain_port= gain*Ci;

gain_links=gain_links_reup*Ci;

// affichage des gains//
gain_links=gain_links_reup*Ci;

cput<< ("\n\n le Gain dans les Ports est:)<<,gain_port<<endl;

printf("\n\n le Gain dans les Liens est:)<<gain_links<<endl;

}
```

En sortie, nous avons ceci:

```
Number of router : 7
Number ports of router : 8
Number of links : 9
Port off : 49
Number of total port in network : 56
% port off = 87.5%
Link off : 3
% link off = 33.33%
```

ANNEXE D : Liste des Publications tirées de la thèse

➤ **Article 2 (indexé par CASSI, IF= 1.67)**

Kra, L., Gondo, Y., Gooré, B. T. and Asseu, O. (2018) « Contribution to the Optimization of the Energy Consumption in SDN Networks ». Journal of Sensor Technology, Vol.8, No.3, pages 1- 9.

<http://doi.org/10.4236/jst.2018.83005>

➤ **Article 3 (indexé par Scopus)**

Lagasane Kra¹, Pamela Yoboué², Adama Konaté^{1,2} and Olivier Asseu^{1,2} (2019)
«Algorithmic Approach to Minimizing Energy Consumption in SDN type Networks» Far East Journal of Mathematical Sciences (FJMS), Volume 113, Number 1, 2019, Pages 115-126.

<http://dx.doi.org/10.17654/MS113010115>

➤ **Article 4 (indexé par CAS et Electronics, IF= 0.75)**

Lagasane Kra¹, Bi Tra Gooré¹, Manlandon Koffi¹ and Olivier Asseu^{1,2}(2018)
« Simulation of a mathematical model minimizing energy consumption in SDN type networks». Journal Engineering (ENG).

<https://doi.org/10.4236/eng.2018.1011056>

Autres contributions d'articles publiés par l'auteur

➤ **Article 1 (indexé par Scopus)**

R. Zamblé¹, L. Kra², M. Koffi², R. Gbégbé², B. T. Gooré² and O. Asseu^{1,2}(2018)
«Power consumption optimization in 5g radio access networks, based on a call admission control algorithm». Far East Journal Electronics and Communications, Vol.18, No.4, Pages 507-520.

<http://dx.doi.org/10.17654/EC018040507>



POWER CONSUMPTION OPTIMIZATION IN 5G RADIO ACCESS NETWORKS, BASED ON A CALL ADMISSION CONTROL ALGORITHM

R. Zamblé¹, L. Kra², M. Koffi², R. Gbégbé², B. T. Gooré² and O. Assou^{1,2}

¹Ecole Supérieure Africaine des TIC (ESATIC)

Abidjan, Côte d'Ivoire

²Institut National Polytechnique Felix Houphouët Boigny (INP-HB)

Yamoussoukro, Côte d'Ivoire

Abstract

The combination of Cloud Radio Access Network (C-RAN) and Heterogeneous Network (HetNet) networks has recently attracted a great deal of interest in academic and industrial fields. This architecture, called H-CRAN (Heterogeneous C-RAN) is therefore rightly considered as a promising solution for increasing the spectral efficiency and energy efficiency of 5G networks. As C-RAN network offers the advantage of energy reduction because BBU can manage many RRHs and the HetNets increases user's throughput on account of the reduction between base stations and users. This article proposes a call admission control (CAC) algorithm focused on the radio remote heads (RRHs) in order to further reduce the power consumed by the H-CRAN by switching into sleep mode, the under-used RRHs. Simulations were performed into two contexts depending on the cell radius: in the context 1, the radius is 1 km and 0.5 km in the second context. These simulations focused on the power consumed as function of UEs number.

The simulations have shown that, more the cell radius is lower and

Received: October 11, 2017; Accepted: February 1, 2018

Keywords and phrases: 5G, H-CRAN, RRH, CAC, HetNet, sleep mode, idle mode.

more our algorithm reduces the power consumption with rates between 66.66% and 88.28% and the energy efficiency is multiplied by a ratio of 3.4.

I. Introduction

Today's mobile networks are facing a scarcity of allocated spectrum resources and exponential growth of services that are increasingly greedy in terms of throughput and very demanding of low latency. To provide a satisfying level of QoS to their customers, they are expanding their radio access network. But this expansion increases the operating and maintenance costs (OPEX, CAPEX) and energy consumption. The fifth generation of mobile network (5G), which is planned to be launch in 2020, is expected to deliver data rates of 10 to 100 times compared to the 4G network and an energy consumption reduction around 90% [1, 2]. Many recent studies in academic and industrial fields have shown that the combination of C-RAN and HetNets networks could be one of the key technologies to meet the challenges mentioned above and to enhance the performance of 5G [3-5]. This type of network is called **H-CRAN** (*Heterogeneous Cloud Access Network*). HetNets consist of integrating many low power nodes (*LPN*) or small cells (micro, pico or femto cells) inside macro-cells or HPNs (High Power Nodes). This increases network coverage and user throughput. As for C-RAN, it is a centralized architecture in which the part responsible for the baseband processing is separated from the part responsible for basic radiofrequency processing of the base stations (BS) [5]. The baseband processing is performed by BBUs (*Base Band Unit*) which are located together on a centralized site called BBU Pool, to better manage resource allocation, and interferences mitigation. The radiofrequency processing is done by remote radio heads (*RRH: Remote Radio Head*). Beyond efficient resource management, the centralization of BBUs can reduce the energy costs of air conditioning [4].

In H-CRANs the macro BS (MBS or M_RRH) is responsible for signalling control, while the RRHs are responsible for the transmission of high-speed data to the users. Several researchers are working to optimize the

use of spectrum and energy in H-CRANs. Among these works we can cite the so-called “clustering” method as an objective to determine the best BBU-RRH association in order to minimize the number of RRHs allocated to a data stream or to maximize the number of RRHs by BBU [6-8]. By reducing the number of active BBUs or RRHs, clustering mechanisms reduce the energy consumption of the system. On the other hand, some research’s works focuses on mechanisms of switching into sleep mode BBUs or RRHs to increase the energy efficiency [9]. For example, Zuo et al. [10] studied a joint mechanism for RRHs activation, a best UE-RRH association and resources allocation to maximize energy efficiency H-CRANs. Other researchers [11, 12] focused their studies on algorithms whose aim is to minimize both the total power consumed and the number of switching of **ACTIVE/IDLE** modes in a H-CRAN network with QoS guarantee.

This paper describes an algorithm whose objective is to minimize the total power consumption in the H-CRAN. The principle is to switch in sleep mode, the RRHs which are underused, that is to say the RRHs whose total power emitted is below a certain threshold of power that we are going to fix. The UEs associated with these RRHs will then be reassigned to the other RRHs that are not saturated and underused. The simulations were carried out in two contexts: in the first one we consider a macro cell of radius of 1 km and in the second the radius of the cell is 0.5 km. We then analyse the evolution of the total power consumed by the network as a function of the number of UEs in the cell. The paper is organized as follows: Section II describes the model of system and the formulation of the optimization problem. In Section III, we describe our optimization algorithm, and show the results of the simulations that have been carried out. The last part, the section IV, deals with the conclusion and perspectives of ours studies.

II. System Model

Our system model is an H-CRAN network consisting of a set of BBUs located in a BBU Pool, and macro cells within we have macro RRH or HPN (*High Power Node*) and RRHs or LPNs (*Low Power Node*). Each macro cell

is composed of a macro BS or macro RRH responsible for signalling and a set of micro, pico or femto RRHs that are responsible for the high bit rate transmission of user data. The RRHs are randomly distributed inside the macro cell.

In the following sections, we focus only on the power consumed by the set of $(M-1)$ RRHs of a single macro cell as illustrated in Figure 1.

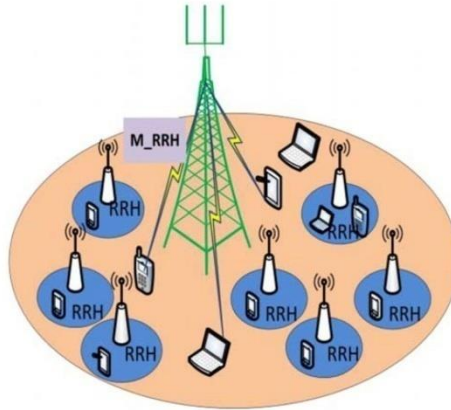


Figure 1. System model.

In the rest of the article, it was assumed that the latency induced by the links between a BBU Pool and the set of RRHs is null.

1. Energy model of base stations

It is assumed that the macro cell is composed of a number M of RRHs: a macro BS or macro RRH and $(M-1)$ RRHs of the “small cell” type. The variables P_{MBS} or P_{M_RRH} are defined as the power consumed by the macro BS or M_RRH and R_{RRHm} defines the power consumed by the RRH number m of the microcell. We define β_m as the activation binary variable of the RRH_m and λ as that of the macro BS; so, we have:

$$\beta_m = \begin{cases} 1 & \text{if the } RRH_m \text{ is ON,} \\ 0 & \text{if the } RRH_m \text{ is OFF,} \end{cases} \quad (1)$$

$$\lambda = \begin{cases} 1 & \text{if the } RRH_m \text{ is ON,} \\ 0 & \text{if the } RRH_m \text{ is OFF.} \end{cases} \quad (2)$$

For UE $_k$ association with a BS we define the binary variable a_{mk} which is equal to 1 when the UE $_k$ is served by the RRH_m and 0 in otherwise. We also introduce b_k the binary variable indicating the attachment of the UE $_k$ to the macro BS.

The power consumption of the RRH_m in the macro cell is modeled as below:

$$P_{RRH,m} = \begin{cases} P_m + P_{CC,m} & \text{if the } RRH_m \text{ is active,} \\ P_{0,stat} & \text{if the } RRH_m \text{ is in sleep mode,} \end{cases} \quad (3)$$

where $P_{CC,m}$ is the static power of RRH_m when it is active and P_m is the sum of the powers supplied to the UEs served by the RRH_m :

$$P_m = \sum_{k=1}^K a_{mk} P_{mk};$$

P_{mk} is the power transmitted by the RRH_m to the UE $_k$. It is thus deduced that:

$$P_{RRH,m} = \beta_m \left(P_{CC,m} + \sum_{k=1}^K a_{mk} P_{mk} \right) + (1 - \beta_m) P_{0,stat}. \quad (4)$$

Similarly, the energy consumption model of the macro MBS or M_{RRHm} , can be defined as follows:

$$P_{MBS} = \begin{cases} P_{MBS} + P_{CC,BS} & \text{if the } MBS \text{ is active,} \\ P_{BS,stat} & \text{if the } MBS \text{ is in sleep mode} \end{cases} \quad (5)$$

with $P_{CC,MBS}$ the static power of the MBS when it is active and P_{MBS} is the

sum of the powers supplied to the UEs served by the MBS:

$$P_{MBS} = \sum_{k=1}^K b_k P_{BS,k}.$$

$P_{MBS,k}$'s define the power transmitted by the MBS to the UE $_k$. Therefore, the power consumed by the MBS can be defined as follows:

$$P_{MBS} = \lambda \left(P_{CC,MBS} + \sum_{k=1}^K b_k P_{MBS,k} \right) + (1 - \lambda) P_{MBS,stat}. \quad (6)$$

For each macro cell, the total power consumed is therefore:

$$P_T = P_{MBS} + \sum_{m=1}^{M-1} P_{RRH,m} = P_{M_RRH} + \sum_{m=1}^{M-1} P_{RRH,m}. \quad (7)$$

2. Problem formulation

Minimize:

$$P_T = \lambda \left(P_{CC,MBS} + \sum_{k=1}^K b_k P_{MRRH,k} \right) + \sum_{m=1}^{M-1} \beta_m \left(P_{CC,m} + \sum_{k=1}^K a_{mk} P_{mk} \right). \quad (8)$$

$$(1) \sum_{k=1}^K a_{mk} P_{mk} \leq P_{RRH_max}; \forall m < M,$$

$$(2) \sum_{k=1}^K a_{mk} P_{mk} \geq P_{RRH_seuil}; \forall m < M,$$

$$(3) \sum_{k=1}^K b_k P_{MBS,k} \leq P_{MBS_max},$$

$$(4) \sum_{k=1}^K b_k P_{MBS,k} \geq P_{MBS_seuil}.$$

3. Calculation of the required power of an RRH to guarantee the user bit rate

The scenario considered here assumes that each UE receives the minimum bit rate $R_{\min,k}$ satisfying its QoS. According to Shannon's formula, we can say that:

$$R_{mk} = W_k \cdot \log_2 \left(1 + \frac{P_{mk} \cdot g_{mk}}{N_0 W_k + \sum_{i \neq m}^I P_{ik} \cdot g_{ik}} \right)$$

with

- W_k : the bandwidth of UE k ,
- P_{mk} : the power transmitted by RRH m to UE k ,
- g_{mk} : the channel gain,
- N_0 : the noise spectral density,
- $\sum_{i \neq m}^I P_{ik} \cdot g_{ik}$: the co-channel interferences,
- I : the number of interfering BS.

It is assumed that $W_k = W_{PRB} = 180$ kHz, the bandwidth of an OFDM resource block in LTE. Co-channel interference is neglected in our scenario. To guarantee the QoS, we must have:

$$R_{\min, k} = W_{PRB} \cdot \log_2 \left(1 + \frac{P_{mk} \cdot g_{mk}}{N_0 W_{PRB}} \right),$$

$$\frac{R_{\min, k}}{W_{PRB}} = \log_2 \left(1 + \frac{P_{mk} \cdot g_{mk}}{N_0 W_{PRB}} \right).$$

From this we deduce that

$$P_{mk} = \left(2^{\frac{R_{\min, k}}{W_{PRB}}} - 1 \right) \cdot \frac{N_0 W_{PRB}}{g_{mk}} \quad (9)$$

or $P_{mk} = \left(2^{\frac{R_{\min, k}}{W_{PRB}}} - 1 \right) \cdot N_0 W_{PRB} \cdot L_{mk}$ with $L_{mk} = 1/g_{mk}$ pathloss between

RRH m and UE k

4. Formulation of the optimization problem

Minimize:

$$P_T = \lambda_q \left(P_{CC, MBS} + \sum_{k=1}^K b_{qk} P_{M_{RRH, k}} \right) + \sum_{m=1}^{M-1} \beta_m \left(P_{CC, m} + \sum_{k=1}^K a_{mk} P_{mk} \right). \quad (10)$$

Constraints:

- i. $\sum_{k=1}^K a_{mk} P_{mk} \leq P_{RRH_max}; \forall m < M,$
- ii. $\sum_{k=1}^K a_{mk} P_{mk} \geq P_{RRH_seuil}; \forall m < M,$
- iii. $\sum_{k=1}^K b_k P_{MBS,k} \leq P_{MBS_max},$
- iv. $\sum_{k=1}^K b_k P_{BS,k} \geq P_{MBS_seuil}.$

The power $P_{M_RRH,k}$ is calculated in a similar way to (9):

$$P_{M_RRH,k} = \left(2^{\frac{R_{\min k}}{W_{PRB}}} - 1 \right) \cdot N_0 W_{PRB} \cdot L_{M_RRH,k}. \quad (11)$$

- $L_{M_RRH,k}$ = pathloss between the M_{RRH} and the UE k

We assume in our algorithm that the consumption of a RRH in sleep mode is null:

$$P_{0,stat} = P_{BS,stat} = 0.$$

5. Simulation parameters

We consider a scenario in which the network architecture is a H-CRAN architecture consisting of one macro cell. The macro cell includes a macro BS or macro RRH and a set of (M-1) small RRHs.

Table 1. Simulation parameters and their meaning

Parameters	Meaning	Values
M	Number of RRHs per macro cell	Variable
K	Number of UEs per macro cell	Variable
P_{RRH_max}	Maximum power of a small RRH	200 mw
P_{MBS_max}	Maximum power of a macro RRH	20 W
P_{RRH_seuil}	Minimum power threshold for a small RRH	$0 < threshold < 1$

P_{MBS_seuil}	Minimum power threshold for a macro RRH	$0 < threshold < 1$
P_{CC_m}	Constant power corresponding to the power consumed by the RRH circuits	20% P_{RRH_max}
P_{CC_MBS}	Constant power corresponding to the power consumed by the macro RRH circuits	20% P_{MBS_max}
N_0	Noise spectral density	-174 dbm/Hz
L_{mk}	Pathloss between the RRHm and the UEk	$130.62 + 37.6\log_{10}(d_{mk})$ d _{mk} = distance (RRHm, UEk)
$R_{min,k}$	Minimum bit rate required by an EU for a service, assumed to be constant for all UEs	2 Mbps

6. Functions to be simulated

In this article, we are going to study the evolution of the total power consumed P_T by the system as a function of the number of UEs: $P_{T_UE} = f(\text{number of UEs})$.

It is assumed that the UEs are randomly distributed throughout the cell as shown in Figure 2.

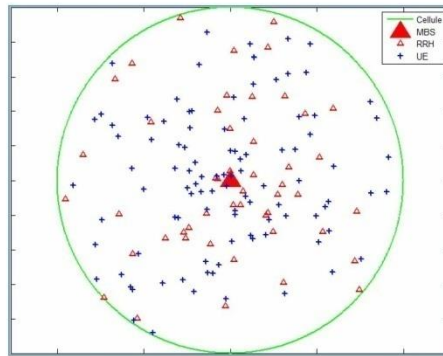


Figure 2. Distribution of the UEs in the cell.

III. Description of the Algorithm and Simulation Results

1. Description of the algorithm

Our algorithm consists of 2 main functions:

- F1: allocation by standard CAC,
- F2: allocation by CAC taking into account a power threshold.

An RRH may be considered to designate a small RRH or the macro RRH.

The diagrams below describe the main components of our CAC algorithm, also called UE-BS association algorithm.

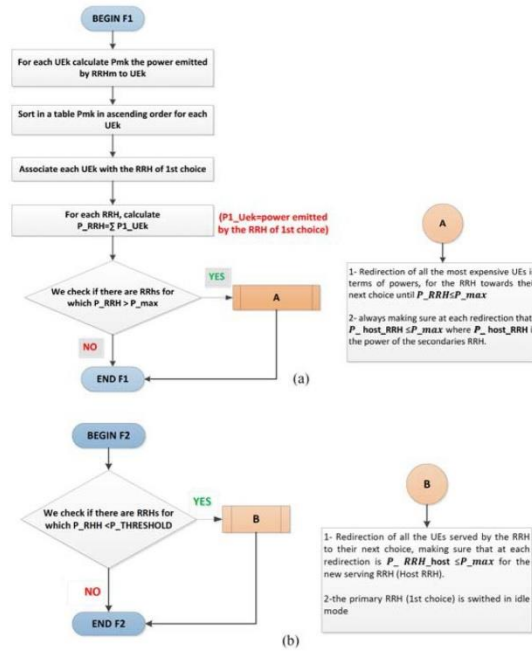


Figure 3. Description of the main functions of UE-BS algorithm: (a) conventional CAC; (b) CAC with threshold.

After running the algorithm, we compare the power obtained by the function F1 to the power obtained by the function F2.

2. Simulations results

In this section, we are presenting the results of the simulations we carried out according to the two contexts: in context 1, the radius of the cell is 1 km and in the context 2, the radius is fixed at 0.5 km. In each context, we represent the evolution of the total power as a function of the number of RRHs: $P_{T_UE} = f(\text{number of UEs})$. In our program, we average 500 snapshots to represent the function P_{T_UE} . It is assumed that the throughput of each UE is set to 2 Mbps.

Table 2. Simulation parameters values

Function	Threshold	K = number of UEs	M = number of RRHs
$P_{TR} = f(\text{nombre_RRH})$	0.5	Variable	50

Figure 4 below shows the results of our simulation.

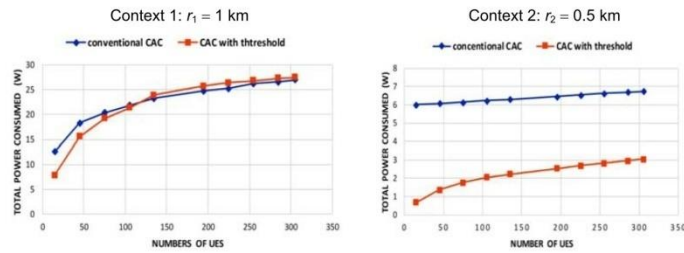


Figure 4. Impact of the cell on the total power consumed, as a function of UEs number.

Figure 4 shows that for a given number of RRHs and a given power threshold the power consumed $P_{T_UE} = f(\text{number of UEs})$ is slightly lower (an average of 15%) in our CAC algorithm than conventional CAC in the context 1, up to 100 UEs.

Beyond 100 UEs, still in context 1 our CAC algorithm consumed about 3.14% of power than the conventional CAC. On the other hand, the ratio of RRHs switched in idle mode varies between 40.96% and 87.42%. It appears that our algorithm in this context can put a significant number of RRHs in sleep mode, but does not offer satisfactory results especially when the number of UEs increases in the cell.

The classical CAC allows the lower power to the UE (first choice), without taking to account the power threshold. But the threshold CAC put in idle mode many RRHs and reassigned them to their second, third choice or other choice, in order to switch in sleep mode, the underused RRHs. But, as the cell radius is high the secondary choice of each reallocated UEs is high in terms of power. So even our CAC algorithm puts the underused in idle mode, but it does not offer enough power reduction, because of the high radius of the cell.

In the context 2, the cell radius is 0.5 km, the power consumed by the two algorithm is lower than the power consumed by both algorithms in context 1. It varies between 6 and 6.75 W with conventional CAC, and goes from 0.7 W (15 UEs) to 3W (305 UEs). This represents a power reduction ratio between 54.77% and 88.39% (305 UEs). The ratio of RRHs switched into sleep mode varies between 66.66% and 88.28%. This reduction of power is due to the cell radius. We have a density of RRHs/km² which is multiplied by 4 compared ($r = 0.5$ km in context 2) to the context 1 ($r = 1$ km). So, each UE is closer to its RRHs of first choice and secondary choice. As a result, the power consumed in context 2 is lower compared to the context 1.

IV. Conclusion

In this paper, we proposed to design call admission control (CAC) algorithm or RRH-UE algorithm in order to reduce power consumption in a 5G H-CRAN network. We have defined an algorithm that switches in sleep mode all RRHs whose radiated power is below a power threshold that we have set. We then analyzed the evolution of the total power in two contexts of cell radius with a random distribution of UES throughout the cell. In the

second context, the EE is multiplied averagely by 3.4 and reduces the power between 54.77% and 88.39%. So, when the cell radius is lower and more the EE increases in general, but even more with our CAC algorithm.

In the future research it will be interesting to test this algorithm in a context where the UEs are in mobility situation and also taking into account the effect of interferences because practically, the UEs are subject to interferences and most of them are moving.

References

- [1] Mugen Peng, Yuling Yu, Hongyu Xiang and H. Vincent Poor, Energy-efficient resource allocation optimization for multimedia heterogeneous cloud radio access networks, *IEEE Transactions on Multimedia* 18(5) (2016), 879-892.
- [2] Mugen Peng, Yuan Li, Jiamo Jiang, Jian Li and Chonggang Wang, Heterogeneous cloud radio access networks: a new perspective for enhancing spectral and energy efficiencies, *IEEE Wireless Communications* 21(6) (2014), 126-135.
- [3] Mugen Peng, Yong Li, Zhongyuan Zhao and Chonggang Wang, System architecture and key technologies for 5G heterogeneous cloud radio access networks, *IEEE Network* 29(2) (2015), 6-14.
- [4] C. Kullin and D. Ran, C-ran the road towards green ran, China Mobile Research Institute, White Paper, 2011.
- [5] Shinobu Namba, Takayuki Warabino and Shoji Kaneko, BBU-RRH switching schemes for centralized RAN, 2012 7th International ICST Conference on Communications and Networking in China (CHINACOM), IEEE, 2012, pp. 762-766.
- [6] Karen Boulos, Melhem El Helou and Samer Lahoud, RRH clustering in cloud radio access networks, 2015 International Conference on Applied Research in Computer Science and Engineering (ICAR), IEEE, 2015, pp. 1-6.
- [7] S. Xu and S. Wang, Efficient algorithm for baseband unit pool planning in cloud radio access networks, 2016 IEEE 83rd Vehicular Technology Conference (VTC Spring) (2016, May), IEEE, pp. 1-5.
- [8] Ying Loong Lee, Li-Chun Wang, Teong Chee Chuah and Jonathan Loo, Joint resource allocation and user association for heterogeneous cloud radio access networks, 2016 28th International Teletraffic Congress (ITC 28), IEEE, 2016, pp. 87-93.

- [9] Marco Dolfi, Simone Morosi, Cicek Cavdar and Enrico Del Re, Energy efficient optimization of a sleep mode strategy in heterogeneous cellular networks, 2017 European Conference on Networks and Communications (EuCNC), IEEE, 2017, pp. 1-6.
- [10] Jun Zuo, Jun Zhang, Chau Yuen, Wei Jiang and Wu Luo, Energy efficient user association for cloud radio access networks, IEEE Access 4 (2016), 2429-2438.
- [11] Yongyu Dai, Shi Jin, Leyuan Pan, Xiqi Gao, Lei Jiang and Ming Lei, Interference control based on beamforming coordination for heterogeneous network with RRH deployment, IEEE Systems J. 9(1) (2015), 58-64.
- [12] M. Khan, Raad S. Alhumaima and Hamed S. Al-Raweshidy, Reducing energy consumption by dynamic resource allocation in C-RAN, 2015 European Conference on Networks and Communications (EuCNC), IEEE, 2015, pp. 169-174.

Contribution to the Optimization of the Energy Consumption in SDN Networks

Lagasane Kra¹, Yaké Gondo², Bi Tra Gooré¹, Olivier Asseu^{1,3}

¹Institut National Polytechnique F. Houphouët Boigny (INP-HB), Yamoussoukro, Côte d'Ivoire

²Université Félix Houphouët Boigny (UFHB), Abidjan, Côte d'Ivoire

³Ecole Supérieure Africaine des TIC (ESATIC), Abidjan, Côte d'Ivoire

Email: oasseu@yahoo.fr

How to cite this paper: Kra, L., Gondo, Y., Gooré, B.T. and Asseu, O. (2018) Contribution to the Optimization of the Energy Consumption in SDN Networks. *Journal of Sensor Technology*, 8, 59-67.
<https://doi.org/10.4236/jst.2018.83005>

Received: September 4, 2018

Accepted: September 26, 2018

Published: September 29, 2018

Copyright © 2018 by authors and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

With the advent of new technologies such as IoT (Internet of Things) and Big Data, the increase in users and their different communications have led to a significant increase in energy consumption in network equipment. A new networking technology called SDN (Software Defined Network) is born. It aims to make network management easier. The SDN consists of decoupling the control plane that is the brain, the data plane or the muscles of the network. It allows the programmability of network devices and also the redirection of flows. One or more centralized controllers use algorithms to act remotely on network devices. Because of its operation, this new technology offers opportunities to improve network performance and optimize energy consumption. In this paper, we will use this technology (SDN) to suspend links or routers when they are not used while taking into account the congestion that degrades the quality of service in the network. We have formulated this problem as a linear integer program and proposed algorithms to process it in normal period and peak period. We have used the OMNET ++ simulator to evaluate our algorithms. Thus, our approach showed that 87.5% of ports and 33.33% of links could be shut down to save energy.

Keywords

IoT, SDN, Big Data, Optimization, Peak Period, Normal Period, Heuristic Algorithms

1. Introduction

Communication networks are progressively evolving in terms of size and performance. There are two types of network equipment: active devices such as

routers, switches, etc. (information transmission devices) and passive devices such as cables, fiber optics, etc. (interconnection equipment). Of the two types of equipment, only assets are energy intensive. The evolution of the networks leads to an increase of these equipments in quantity and in performance. The increase of these and their performance increases the consumption of the electrical energy they need for their operation.

A study conducted in 2009 [1] shows that information and telecommunications technologies (ICTs) alone consume 2% to 10% of global energy consumption. Hubs, switches and routers consume 6 Twh/year in the USA [2]. Therefore the search for a mathematical model for the consumption of energy in the communication networks becomes a real concern for the companies of the moment. In traditional networks, when a packet arrives on a port of a switch or router, it applies the routing or switching rules that are registered in an operating system. Generally, all packets that have the same destination follow the same path. In high-end models, hardware is able to recognize the type of application and apply the specific rules to it. But this programming is rigid. It can only be changed manually by the administrator, which obviously takes time [3].

The advent of Software Defined Network (SDN) technologies appears to be a good alternative for acting remotely and dynamically on equipment in order to model energy consumption.

In SDN technology, it is a centralized controller that will be responsible for routing packets in the network via SDN protocol (openflow) by programmability by injecting routing rules provided by the application layer (Figure 1).

In this paper, we use this paradigm to act on network devices by enabling or disabling router ports when they are not working. A new strategy has been developed which takes into account the peak periods (dense traffic) and normal (low traffic) minimizing energy while avoiding congestion. The authors of [4] have in their minimized approach with high delays leading to packet losses.

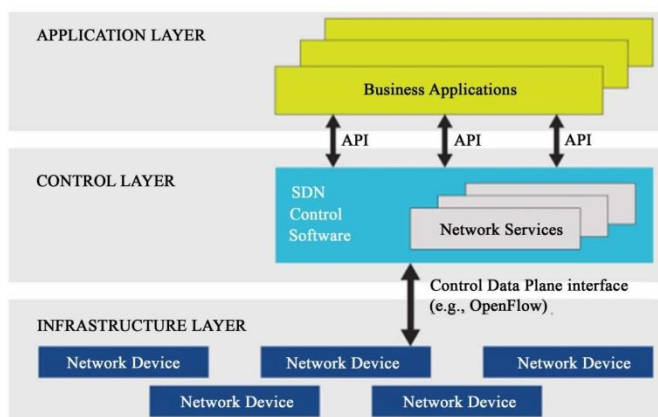


Figure 1. SDN architecture [3].

Our model is based on that of these authors [4]. We have implemented an energy minimization model based on a new strategy. Our work will revolve around the following points:

Section 2 will cover previous work. Section 3 describes the mathematical model of our approach. We present in Section 4, the resolution approach. In Section 5, we will evaluate the performance of the model. Section 6 concludes our article.

2. Previous Work

In this section, we will present some previous work that has addressed the issue of energy consumption in networks. We also formulate some basic assumptions that our future mathematical model must respect.

2.1. Mathematical Model Research Reduce Energy Consumption Optimally

The consumption of energy has been treated by some researchers [4] [5] [6] [7] [8]. In [4], the authors have minimized under three constraints using on/off technology in conventional wired networks, the consumption of energy in the networks (delay, packet loss and jitter). A saving of 40% was obtained. But at each extinction, a delay variation led to packet losses in case of heavy traffic. The authors of [5] [8] minimized the energy in SDN networks using the “compression” approach of the routing table. According to the authors, the devices that can implement the SDN rules uses TCAM (Content-addressable Ternary Memory). This memory in which are recorded the rules, is expensive and greedy in energy. So you have to compress the routing table. Compression maximizes rule space, increasing the number of paths. The consumption of a high-capacity link and that of an unsolicited link is low, some flows can be redirected to other links. As a delay solution, the authors have defined the default rule for forwarding packets to the default port without contacting the SDN controller. An energy saving has been observed but with degradation of the quality of service in terms of congestion in the network. As for the works of [6] and [7], the authors have shown that it is the number of ports of the routers on the network that consume. Consumption modeling would reduce router interfaces when they are not in use.

We orient our work in this way while finding a mathematical model of energy minimization in the SDN network which satisfies the “QoS” in terms of delay of transmission and loss of packets.

2.2. Basic Assumptions

- ✓ Each port has the same rate of energy consumption.
- ✓ Each port can handle multiple services so it can redirect traffic.
- ✓ Choose in the graph, the path or paths that seem to offer more energy saving.

Our future mathematical model will be obtained through a set of processes that will involve theories such as graphs; trees etc.

3. Modeling the Problem

This section will cover the description of our mathematical model.

We formulate our problem as a linear integer program in contributing to the optimization of energy consumption in SDN networks. Let N be a network and $n(t)$ a sub-network of N ($n(t) \subseteq N$). We defined a dynamic approach based on two situations of graph theory:

- 1) The activation and deactivation of the ports of the routers for a given time t . we note the change in energy consumption.
- 2) We will assume that a router cannot be turned off and therefore must remain awake.

Once switched on, at each given instant t , a link is established between a router i and a router j . It will therefore be a question of maximizing a number of ports to be deactivated in order to have energy savings under "QoS" constraints. In our approach, we will enable or disable router ports using a smart approach.

The number of ports to disable $\ell(t)$ is defined as:

$$\begin{aligned} \ell: [0, +\infty[&\rightarrow N \\ t &\mapsto \ell(t) \end{aligned}$$

Consider λ_{\max} , the total number of ports on a router. This parameter λ_{\max} is fixed and depends on the types of routers. If the total number of ports on a router is 8, how much should I disable and how, to hope to save the maximum energy?

Let us consider the function f the consumption of energy in the sub-networks. We express the model of our approach in mathematical form called objective function of the solution of the problem that is to say (1) and (2).

Which amounts to:

$$f(n(t), t) = \sum_{i \in n(t)} f(i, t, \ell(t)) + \sum_{i, j \in n(t)} f(k(i, j, t)) \quad (1)$$

A link is the junction between two routers interfaces. $i, j \in N$ having a maximum load capacity of $C(i, j)$ packet traffic per second. The links can be in the following state 0 or 1. Let k be the link state variable: 0 state where the router is off and 1 state where the router is on.

The objective function to be minimized becomes:

$$f(n(t), t) = \sum_{i \in n(t)} f(i, \ell(t), t) + \sum_{i, j \in n(t)} f(k(i, j, t)) \quad (2)$$

where $f(i, \ell(t), t)$ is the energy consumption of the router i whose $\ell(t)$ ports were deactivated at time t and $f(k(i, j, t))$ represents the energy of the link i, j at time t .

$f(i, t, \ell(t))$ is the function to be explained. Power gain function of router i "g". The function g is a linear and increasing function with respect to the number of ports deactivated at time t .

Let $g_i(t, \ell(t))$: the energy gain of the router i such that:

$$\max_t \ell(t) \leq \lambda_{\max}, \text{ with } g_i(t, \ell(t)) = a\ell(t) + b$$

where $a > 0$.

$$\text{if } \max_t \ell(t) \leq \lambda_{\max}, \text{ then } f_i(t, \ell(t)) = C_i - (a\ell(t) + b) \quad (3)$$

This is the final expression of our function “energy consumption” of router. Otherwise $f_i(t, \ell(t)) = 0$ where C_i is the total consumption of the router i and $f(i, t, \ell(t)) = f_i(t, \ell(t))$.

We deduce that if all ports are disabled, $g_i(t, \ell(t)) \approx C_i$.

We showed above that our function is linear in $\ell(t)$. It is thus refined in the form of:

$$f(n(t), t) = \sum_{i \in n(t)} (C_i - (a\ell(t) + b)) + \sum_{i, j \in n(t)} f(k(i, j, t))$$

Now every affine function is convex and therefore admits a minimum. Our model then becomes:

$$\min(f(n(t), t)) = \min \left(\sum_{i \in n(t)} (C_i - (a\ell(t) + b)) + \sum_{i, j \in n(t)} f(k(i, j, t)) \right) \quad (4)$$

With $C_i =$ chassis consumption (C_{ch}) + consumption of ports (C_p) on. From where:

$$C_i = C_{ch} + C_p$$

The explicit objective function becomes:

$$\min(f(n(t), t)) = \min \left(\sum_{i \in n(t)} ((C_{ch} + C_p) - (a\ell(t) + b)) + \sum_{i, j \in n(t)} f(k(i, j, t)) \right) \quad (5)$$

Under constraint of:

$$q(f(t), R(F(t), n(t), k(t), t)) \leq Q(f(t)), \forall f(t) \in F(t)$$

$$\text{and } \sum f(t) \leq C_{i,j}, \forall f(t) \in F(t)$$

$$k \in \{0, 1\}$$

Summary of the ratings:

$f(n(t), t)$	Power consumption in the subnet
λ_{\max}	Total number of ports on a router
C_i	Overall energy of the router
$a\ell(t) + b$	Gain of energy obtained during the deactivation of the ports
$f(k(i, j, t))$	The state of ports in s/networks at time t
q	Quality of service perceived by the user
$F(t)$	Set of all requests at the moment t
$f_{s,p}(t) \in F(t)$	Observed request of a user at the instant t from a point s to p
$R(F(t))$	Routing of all requests at time t (see the routing algorithm)
$n(t) \subseteq N(t)$	The subnet included in the network at time t
$Q(f(t))$	Quality of service set to not exceeded for all requests observed

We have found our mathematical model which minimizes, so a method of resolution proves necessary.

4. Resolution Approach

In this part, we proposed a resolution method of our mathematical model.

Solving the problem consists of:

- ✓ Solve by integer linear programming $f_i(t, \ell(t)) = C_i - (a\ell(t) + b)$
- ✓ Solve by the **Kruskal** algorithm by looking for the maximal tree the function $f(k(i, j, t))$

4.1. Principle of Resolution

We are going to build a network (lowest total weight covering tree) from which we will obtain our possible energy savings. The ports to be shut down will consist of those of the adjacent routers. For our simulation, we will use a network of 7 routers each with 8 ports and 9 links. All extinctions of adjacent ports cause those of these links (Figure 2).

Our tree covering the lowest total weight becomes (see Figure 3).

Total weight: $12 + 12 + 12 + 4 + 4 + 5 = 49$.

4.2. Port Extinction/Ignition Algorithm

Our demand-side extinguishing strategy must select all ports that are not affected by the request and dynamically shut them down.

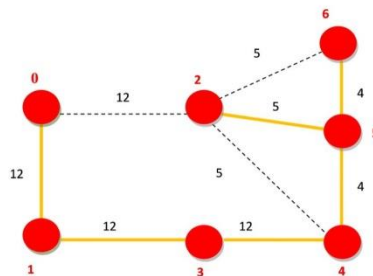


Figure 2. Network with link capabilities.

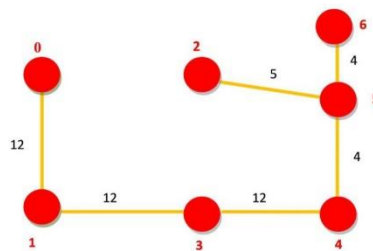


Figure 3. Architecture obtained by the Kruskal algorithm.

In our approach, when a router is turned on, it establishes a link with its neighbors and so on in the network.

Our 7-router network has 8 ports each, only one port is turned on and the (n – 1) ports are off with its links. The circuits are to be avoided in the choice of favorable paths (having the lowest weight).

The Modified SPRING Protocol (**MSP**), is responsible for the extinction of ports and links unsolicited by the request (see Algorithms below).

Algorithm 1: Centralized SDN:<RerouteAndRouterOff>

Data: List-of-Ports:{(0-4),(2-6)..} and the new roads / *Supplied by<SelectPortstoOff>and <ComputeNewRoutes>respectively */

```

1 Begin
2 The controller gives all adjacent nodes the new routes to use for the virtual paths;
3 The controller sees the network as a whole and informs neighboring routers to turn
off all enabled ports except one;
4 Routers 0.1.3 ... send an (ACK) to the controller after the task is executed;
5 The SDN controller identifies the lit ports and turns off the others;
6 All routers apply the change to the forwarding tables for redirection by other ports;
7 Nodes 0,1, 2 ... stop sending IGP packets;
8 Ports (0-1, 2-5 ...) stop and enter sleep mode to be turned off;
```

Algorithm 2: Main loop

```

1 While energy efficient traffic engineering is
active do
2 List-Of-Ports = <SelectPortsToOn>();
3 if <ComputeNewRoutes>(All-Ports-that-are-ON U
List-Of-Ports) then
4 <turn off the links to avoid circuits>;
5 List-Of-Ports = <SelectPortsToOff>();
6 if <ComputeNewRoutes>(All-Ports-that-are-ON \
List-Of-Ports) then
7 <RerouteAndRouterOff>(List-Of-Ports);
```

This **Figure 3** (maximum tree of minimum weight) is obtained thanks to the Kruskal algorithm. The ports of the adjacent routers are stopped using a modification of the SPRING or Segment Routing algorithm as well as the links (see algorithms) [9]. Note that SPRING is designed for SDN.

The resolution approach having been found, we will evaluate the performance of our model.

5. Evaluation and Performance

In **Table 1** below, we present our results.

Table 1. Evaluation of our approach.

Number of routers with 8 ports	Ports off	% Ports off	Number of links	Links off	% Links off
7	49	49/56 = 87.5	9	3	3/9 = 33.33

In our model, we explained high (see 4.1) that the extinct ports are those of adjacent routers. Once the port is off, it drives the one of its links. The SDN is a technology based on the programmability of remote equipment via a centralized controller in the network. The SDN controller, thanks to its global view of the network, is responsible for the extinction of the ports and links, if they are not used by means of a modification of the algorithm SPRING.

Our **Table 1** is obtained from **Figure 2** and **Figure 3**. We applied the Kruskal algorithm to obtain the tree with the lowest weight. This allowed us to have 3 disabled links out of 9. However, we used a network of 7 routers each having 8 ports from which $7 \times 8 = 56$ total ports in the network. When the routers are turned on, only one port on each router is enabled and the $n - 1$ ports are off. In our case, there are 7 ports off on each router i , so 49 (7×7) in total in the network (see **Table 1**).

6. Conclusions

Our mathematical model in SDN networks minimizes energy consumption. Any router chosen, disabling some ports would give a strong energy saving by this model. Our model is flexible and gives us a double energy saving in ports and links. In the ports, an extinction of 87.5% is possible to make savings and 33.33% in the links to also make savings.

In this article, we found our mathematical model; however, improvements are possible and will be the subject of our future work.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Chiaraviglio, L., Leonardi, E. and Mellia, M. (2009) How Much Can Internet Be Greened? 2009 *IEEE Globecom Workshops*, Honolulu, HI, 30 November-4 December 2009, 1-6.
- [2] Bruce, N. and Christensen, K. (2005) Reducing the Energy Consumption of Networked Devices. *IEEE 802.3 Tutorial*, San Francisco, 19 July 2005.
- [3] Doherty, J. (2016) SDN and NFV Simplified: A Visual Guide to Understanding Software Defined Networks and Network Function Virtualization. Addison-Wesley Professional, Boston.
- [4] Gelenbe, E. and Silvestri, S. (2009) Optimisation of Power Consumption in Wired Packet Networks. In: Bartolini, N., Nikolettseas, S., Sinha, P., Cardellini, V. and Mahanti, A., Eds., *Telecommunications Engineering*, Springer Berlin Heidelberg, 717-729.
- [5] Havet, F., Huin, N., Moulrierac, J. and Phan, K. (2015) Green Routing and Compression of SDN Rules. *ALGOTEL 2015-17th French Meeting on the Algorithmic Aspects of Telecommunications*, Beaune, June 2015.
- [6] Giroire, F., Mazauric, D. and Moulrierac, J. (2011) Energy Efficient Routing. *Francophone Meetings on Algorithmic Aspects of Telecommunications (AlgoTel)*, Cap

-
- Esterel, 2011.
- [7] Reviriego, P., Sivaraman, V., Zhao, Z., A-Maestro, J., Vishwanath, A., Sanchez-Macian, A. and Russel, C. (2012) An Energy Consumption Model for Energy Efficient Ethernet Switches. 2012 *International Conference on High Performance Computing & Simulation (HPCS)*, Madrid, 2-6 July 2012, 98-104. <https://doi.org/10.1109/HPCSim.2012.6266897>
 - [8] Giroire, F., Moulhierac, J. and Phan, T.K. (2014) Optimizing Rule Placement in Software Defined Networks for Energy-Aware Routing. 2014 *IEEE Global Communications Conference*, Austin, TX, 8-12 December 2014, 2523-2529. <https://doi.org/10.1109/GLOCOM.2014.7037187>
 - [9] Carpa, R., Glück, O., Lefèvre, L. and Mignot, J.-C. (2017) STREETE: Traffic Engineering for Energy-Efficient Core Networks. *Conference Compas* 2015, Lille, June 2015.



ALGORITHMIC APPROACH TO MINIMIZING ENERGY CONSUMPTION IN SDN TYPE NETWORKS

Lagasane Kra¹, Pamela Yoboué², Adama Konaté^{1,2} and Olivier Asseu^{1,2}

¹Institut National Polytechnique F. Houphouët Boigny (INP-HB)
Yamoussoukro, Côte d'Ivoire

²Ecole Supérieure Africaine des TIC (ESATIC)
Abidjan, Côte d'Ivoire

Abstract

SDN (Software Defined Network) technology is the decoupling of control plane and data plane. This new paradigm has been used by several researchers to propose contributions as a solution for minimizing the consumption of energy. It is one or more centralized controllers that use algorithms to act on remote network devices [1]. Researchers have shown that it is the network equipment that makes it consume when lit (chassis, routers, etc.) [2, 3]. In this paper, we have proposed an algorithmic routing approach that makes the choice of routes by turning off the ports when they are not used. Our algorithm is based on Modified SPRING Protocol (MST). Our model is an improvement of that of the authors [1] and the function "energy gain" is an exponential form. Our results gave 87.5% of dead ports representing 28w of energy saving and 66.62% of links, i.e. 21.32w. The same results of 87.5% or 28w in the ports and $(33.33 \times 2 = 66.62\%)$ in the links either 21.32w are found in the works of the previous authors.

Received: September 3, 2018; Revised: November 13, 2018; Accepted: March 3, 2019

Keywords and phrases: SDN, controller, gain, energy, exponential function, algorithm, model, ports.

1. Introduction

For several decades, energy has become a very important issue in Information Communication Technologies (TICs). The increase in the number of IT equipment has a direct impact on energy consumption when it is used. The interconnection of these devices, regardless of the flow rate, consumes enough energy. The others [4] have shown that the rapid growth in energy consumption is due to the increase in traffic (Figure 1).

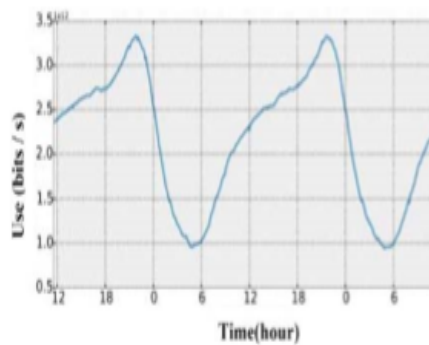


Figure 1. Typical network usage day [4].

A Cisco study estimates that in 2016, it is expected to exceed the zettabyte (10^{21}) of IP traffic traded in one year [5]. As for the authors [6] and [7] also show that “ICTs” alone consume 2 to 10% of world consumption [6]. It is 6Thw/year in the United States [7]. An improvement of the mathematical model of the authors [1] is necessary. We use Software Defined Network (SDN) technology to act on the ports of network devices by enabling and disabling them when not in use. The model of the previous authors minimizes the constrained energy of the quality of service in term of delay, loss of packets and jitter. We will use it by proposing an algorithmic approach for the choice of routes by extinction of the ports. An off port causes the one of its link because the junction between two router interfaces is called “link”. Our energy gain will use an exponential function, what could or might not increase the energy economy.

Section 2 recalls previous work. Section 3 displays the improved author

mathematical model. Section 4 presents the model resolution. In Section 5, our assessments and comparisons with previous work are given. The conclusion follows in Section 6.

2. Previous Literature

This section presents previous literature on the issue of energy consumption in networks. We formulate assumptions and basic data.

2.1. Improved mathematical model to reduce energy consumption optimally

The authors [1, 8, 2] and [3] have dealt with the problem of energy consumption with SDN. In [1], the authors proposed a mathematical model and a strategy of energy saving with simulation based on the extinction of ports under constraint of quality of service. A double economy (ports and links) has been realized. An on/off approach is used by [8] minimized energy by phasing out links or routers under constraint of three parameters (delay, packet loss and jitter). A saving of 40% of economy was observed, but a congestion in dense period brought about an increase of delay, loss of packet and jitter. As for the authors [2] and [3] have shown that it is the number of ports of the routers lit that consume the network. The works of the authors [1], reduced the ports of the routers when these are not used. In our experience, when switching on routers, only one port is enabled and $(n - 1)$ ports are disabled. We will use the exponential function to explain our function "energy gain" of the router i . Our resolution is algorithmic using Kruskal looking for the maximal tree function $f(k(i, j, t))$.

2.2. Basic assumptions

- ✓ Our improved model would minimize energy consumption in SDN networks.
- ✓ We assume that the exponential function would improve the energy gain of the router i in the networks.

2.3. Basic data

- ✓ Each port has the same rate of power consumption on the router types.
- ✓ Use the 48-port Cisco Catalyst 3560-E router example, which has a 15.4w consumption (8 out of 48 ports with a power consumption of 8 is 2.56w). A port consumes 0.32w.

3. New Modeling of the Problem

In this section, we propose our new mathematical model. The idea is similar to the work of the authors [1] except that our economy uses an exponential function as a “gain energy” function.

We formulate our problem as a linear integer program in the algorithmic approach of minimizing energy consumption in SDN-type networks.

Let N be a network and $n(t)$ be a subnet of N with $(n(t) \subseteq N(t))$. Our approach is dynamic:

(i) Enabling or disabling router ports for a given time $n(t)$. We observe the change in the network in terms of energy consumption.

(ii) A router should not shut down. Because its extinction entails that of its neighbors and the ignition, a considerable increase of energy is observed. We consider that a link (i, j) is established between two routers as soon as the routers are switched on. We disable a maximum number $\ell(t)$ ports under QoS constraint.

We define a function that translates the number of ports to disable:

$$\ell : [0, +\infty[\rightarrow N$$

$$t \mapsto \ell(t).$$

We consider λ_{\max} , the total number of ports on a router i . The parameter λ_{\max} is fixed and depends on the type of router.

Let f be the energy consumption function in sub-networks.

(i) and (ii) become:

$$f(n(t), t) = \sum_{i \in n(t)} f(i, t, \ell(t)) + \sum_{i, j \in n(t)} f(k(i, j, t)). \quad (1)$$

The junction of two routers interfaces is called "link". The capacity is denoted by $C(i, j)$, expressing the volume of the packet to be respected on this link. Two states translate the behavior of the links or ports: **0** state where the port is off; **1** state where the port is enabled.

We define the parameters of our objective function:

$f(i, \ell(t), t)$ represents the consumption of the energy of the router i whose ports have been deactivated at time t and $f(k(i, j, t))$ represents the energy of the link i, j at time t . That is $f(i, \ell(t), t)$ should be an explicit function. Power gain function of router i : "g". The function g is a strictly increasing exponential function on \mathbb{R} with respect to the number of ports deactivated at time t .

Let $g_i(t, \ell(t))$, the energy gain of the router i such that:

$$\begin{aligned} \max_t \ell(t) \leq \lambda_{\max} \text{ and } g_i(t, \ell(t)) &= (e^{a\ell(t)+b}), \\ \text{if } \max_t \ell(t) < \lambda_{\max}, \text{ then } f(i, \ell(t), t) &= C_i - (e^{a\ell(t)+b}). \end{aligned} \quad (2)$$

Equation (2) is the final expression of our energy consumption function of a router to minimize.

Otherwise $f_i(t, \ell(t)) = 0$, where C_i is the global consumption of the router i and $f(i, t, \ell(t)) = f_i(t, \ell(t))$. We deduce that if all ports are disabled, $g_i(t, \ell(t)) \approx C_i$.

Our new function is convex on $[0, +\infty[$ because its second derivative is always positive. Therefore, it admits a minimum which can be written as

follows:

$$\min(f(n(t), t)) = \min\left(\sum_{i \in n(t)} (C_i - (e^{a(t)+b}))\right) + \sum_{i, j \in n(t)} f(k(i, j, t)). \quad (3)$$

Knowing that C_i = consumption of the chassis (C_{ch}) + consumption of the port (C_p) which amounts to:

$$C_i = C_{ch} + C_p.$$

Our objective function thus obtained is:

$$\min(f(n(t), t)) = \min\left(\sum_{i \in n(t)} ((C_{ch} + C_p) - (e^{a(t)+b})) + \sum_{i, j \in n(t)} f(k(i, j, t))\right). \quad (4)$$

Under constraint:

$$q(f(t), R(F(t), n(t), k(t), t)) \leq Q(f(t)), \quad \forall f(t) \in F(t))$$

and

$$\sum f(t) \leq C_{i,j}, \quad \forall f(t) \in F(t),$$

$$\text{Max}_t \ell(t) < \lambda_{\max}, \quad k \in \{0, 1\}.$$

Ratings:

Table 1. Summary of ratings

$f(n(t), t)$:	Power consumption in the subnet.
λ_{\max}	:	Total number of ports on a router.
C_i	:	Total energy of the router.
$e^{a(t)+b}$:	Energy savings achieved by deactivating the ports.
$f(k(i, j, t))$:	State of links in the subnet.
q	:	Quality observed with each request.

$F(t)$:	All the requests at instant t .
$f(t) \in F(t)$:	Observed request of a user at instant t .
$R(F(t))$:	Routing all requests at the instant t .
$n(t) \subseteq N(t)$:	The subnet included in the network at time t .
$Q(f(t))$:	Quality not to be exceeded for all requests observed.

4. Model Resolution

A method of solving the model is necessary. It consists of:

- ✓ Solve the ILP function $f(i, \ell(t), t) = C_i - (e^{a\ell(t)+b})$.
- ✓ Solve the Kruskal algorithm by searching the maximum tree for the function that represents the states of the links $f(k(i, j, t))$.

4.1. Algorithm approach

We propose an algorithmic approach to detect the solution of the tree with the lowest weight (Figure 2).

We build a network (the lowest tree with the lowest weight, the extinct ports are those of the adjacent routers, and our network will use 7 routers that respect the number of routers or links made by the authors [8]. Each has 8 ports and 9 links [1]. All port extinctions affect that of its links (Figures 2 and 3).

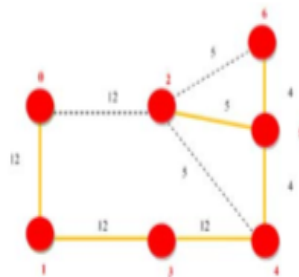


Figure 2. Network with capabilities [1].

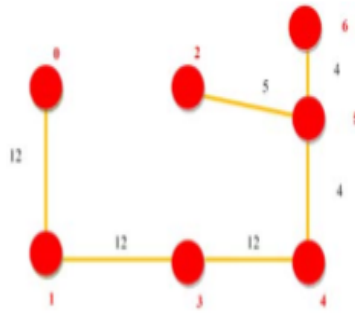


Figure 3. Architecture obtained by the Kruskal algorithm [1].

Our schema is obtained due to the Kruskal algorithm (see algorithm below).

Algorithm 1: Kruskal Algorithm

Data: a simple weighted graph $G = (X, E, W)$;

Result: a tree $A = (X, T)$ covering minimum weight;

// initialization

1 $Cost := 0$;

2 $T := \emptyset$;

// Main loop

3 **While** $|T| < N - 1$ **do**

4 Look for the smallest stop $[u, v]$ not belonging to T

5 **if** $T \cup [u, v]$ does not contain a cycle **then**

6 $T \cup [u, v]$

7 $Cost := Cost + W_{u,v}$

8 **else**

9 Eliminate the edge $[u, v]$

10 **return** A

4.2. Strategy of saving by extinction of the ports

Our demand-side extinguishing strategy, once the routers are turned on, must select all ports that are not affected by the request and turn them off. In our experience, there are 7 routers each with 8 ports. Once the routers are switched on, only one port is activated and the $(n - 1)$ others are off, that is 7. The packets are redirected to the activated ports according to the previous algorithm. The Modified SPRING Protocol [1] is responsible for transferring the packets to specified destinations (example: for a request [0, 2], it must use the routers (1, 3, 4 and 5) while passing on the link of weight 5).

Algorithm 2 (below) shows how ports are turned off and packet redirections.

The SDN controller sees the network as a whole. Once the new ports are selected, it informs each node of the routes and ports that must be entered into its routing table. After that the ports and links are turned **Off** or **On** (Algorithm 2).

Algorithm 2: Centralized SDN: <RerouteAndPortsOff>

Data: List-of-Ports: [(0-4),(2-6)..] and the new roads / * Supplied by
<SelectPortstoOff> and <ComputeNewRoutes> respectively */

- 1 Begin
 - 2 The controller gives all adjacent nodes the new routes to use for the virtual paths;
 - 3 The controller sees the network as a whole and informs neighboring routers to turn off all enabled ports except one;
 - 4 Routers 0.1.3 ... send an (ACK) to the controller after the task is executed;
 - 5 The SDN controller identifies the lit ports and turns off the others;
 - 6 All routers apply the change to the forwarding tables for redirection by other ports;
 - 7 Nodes 0,1, 2 ... stop sending IGP packets;
 - 8 Ports (0-1, 2-5 ...) stop and enter sleep mode to be turned off;
-

Algorithm 2: Main loop

```

1  While energy efficient traffic engineering is
    active do
2  List-Of-Ports = <SelectPortsToOn> ();
3  if <ComputeNewRoutes> (All-Ports-that-are-ON  $\cup$ 
    List-Of-Ports) then
4    <turn off the links to avoid circuits>;
5  List-Of-Ports = <SelectPortsToOff>();
6  if <ComputeNewRoutes>(All-Ports-that-are-ON\
    List-Of-Ports) then
7    <RerouteAndRouterOff>(List-Of-Ports);

```

5. Evaluation and Performance

The authors [8] worked in a network of 44 routers. But it is 7 routers that are each time used to do their experiment. We will therefore use 7 routers each having 8 ports and 9 links.

We also use a 48-port Cisco Catalyst 3560-E type of router [9]. It is 8 ports with 0.32w of consumption per port that we will use as mentioned in point 2.3). Our results are confined in two tables. The first is the assessment of the improved model and the second that of the authors [1] that we compare.

The results are:

Table 5.1. Evaluation of our approach

Number of 8-port routers	Ports extinguished	% off ports	Gain 0.32w/port	Number of links	Links extinguished	% off links	Gain 0.32w/port
7	49	49/56=87.5	28w	9	3	3/9=33.33*2	21.32w

Table 5.2. Evaluation of the approach [1]

Number of 8-port routers	Ports extinguished	% off ports	Gain 0.32w/port	Number of links	Links extinguished	% off links	Gain 0.32w/port
7	49	49/56=87.5	28w	9	3	3/9=33.33*	2

5.3. Comparison of results

By using the function “energy gain” which is an exponential function, we find the same results as those of the previous authors.

We deduce that our approach and strategy are acceptable and allow us to conclude that our gain function is a good choice in the energy economy. Moreover, we say that our function gain ($e^{a\ell(t)+b}$) grows faster than that of authors [1], which is an affine function ($a\ell(t) + b$). This means that the gain function has a better performance than the affine function. Our new model is better than previous ones.

6. Conclusion

The improved mathematical model in energy consumption in SDN networks using an exponential function to express our “energy gain” is good. Its use allowed us to have the same results as the previous authors that is 87.5% in the ports is 28w and 66.66% in the links is 21.32w. These results are significantly better than those of the authors [8] which is 40% in the links 25.6w.

In sum, our approaches and simulations can be proposed as energy saving approach.

Acknowledgements

The authors wish to thank the anonymous referees for his/her comments which allowed us to improve this article. The authors also wish to thank the Higher School of ICT (ESATIC) and the Polytechnic Institute of the Doctoral School F.H. Boigny of Yakro (INP-HB).

References

- [1] L. Kra, Y. Gondo, B. T. Gooré and O. Asseu, Contribution to the optimization of the energy consumption in SDN networks, *J. Sensor Tech.* 8(3) (2018), 1-9.
- [2] P. Reviriego, V. Sivaraman, Z. Zhao, J. A. Maestro, A. Vishwanath, A. Sánchez-Macian and C. Russell, An energy consumption model for energy efficient

- Ethernet switches, 2012 International Conference on High Performance Computing and Simulation (HPCS), 2012, pp. 98-104.
DOI: 10.1109/HPCSim.2012.6266897.
- [3] Frederic Giroire, Dorian Mazauric and Joanna Moulhierac, Energy efficient routing, Francophone Meetings on Algorithmic Aspects of Telecommunications (AlgoTel), 2011, Cap Esterel, France, 2011.
- [4] Timothée Haudebourg, Analysis of the main techniques of energy saving in Ethernet wired network, University of Rennes, June 2015.
- [5] Cisco, The zettabyte era: trends and analysis, White Paper, May 2015.
- [6] L. Chiaraviglio, D. Ciullo, E. Leonardi and M. Mellia, How much can the Internet be greened?, 2009 IEEE Globecom Workshops, 2009, pp. 1-6.
- [7] Bruce Nordman and Ken Christensen, Reducing the energy consumption of networked devices, IEEE 802.3 Tutorial – July 19, 2005 (San Francisco).
- [8] E. Gelenbe and S. Silvestri, Optimisation of power consumption in wired packet networks, Telecommunications Engineering, N. Bartolini, S. Nikolettseas, P. Sinha, V. Cardellini and A. Mahanti, eds., Springer, Berlin, Heidelberg, 2009, pp 717-729.
- [9] Cisco Catalyst 3560-E Series Switches: All Contents are Copyright© 1992-2007, 2009 Cisco Systems, Inc. All rights reserved. This document is Cisco Public Information.

Simulation of a Mathematical Model Minimizing Energy Consumption in SDN Type Networks

Lagasane Kra¹, Bi Tra Gooré¹, Manlandon Koffi¹, Olivier Asseu^{1,2}

¹Institut National Polytechnique F. Houphouët Boigny (INP-HB), Yamoussoukro, Côte d'Ivoire

²Ecole Supérieure Africaine des TIC (ESATIC), Abidjan, Côte d'Ivoire

Email: oasseu@yahoo.fr

How to cite this paper: Kra, L., Gooré, B.T., Koffi, M. and Asseu, O. (2018) Simulation of a Mathematical Model Minimizing Energy Consumption in SDN Type Networks. *Engineering*, 10, 805-813. <https://doi.org/10.4236/eng.2018.1011056>

Received: September 28, 2018

Accepted: November 17, 2018

Published: November 20, 2018

Copyright © 2018 by authors and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Several works by the authors have shown that energy consumption in communication networks does not only depend on the traffic load but on all connected equipment in the network. We have contributed a new mathematical model and a new energy saving strategy with Software Defined Network (SDN) technology [1]. Our Model solution is based on the Modified SPRING Protocol (MSP). In this paper, we simulated our work and compared it to that of the authors [2] and [3]. The OMNET++ simulator was used for our work. Thus, the results of the simulations gave a delay that tends to zero, a packet loss of the order of 10% and a constant jitter of 4% better than the previous authors.

Keywords

SPRING, Protocol, Simulator, OMNET++, Graph, Mathematical Model

1. Introduction

In Wael Zouaoui's thesis, which was defended on January 15, 2016, the authors showed that, following recent advances in the technological fields, the energy bill has continued to grow. This is due to the interconnections of computer equipment. These devices are increasingly bulky with high computing power (see **Figure 1**).

In our previous work accepted for publication in [1], we mentioned the authors of [5] and [6]. In this work, a study has shown that ICTs alone consume 2% to 10% of global consumption [5] and a consummation of 6 Twh/an to USA in hubs, Switch, Routers [6]. It was necessary to find a mathematical model that is able to minimize the consumption of energy.

We have proposed a mathematical model that minimizes energy and its relevance has been shown by an algebraic resolution. In this paper, we will

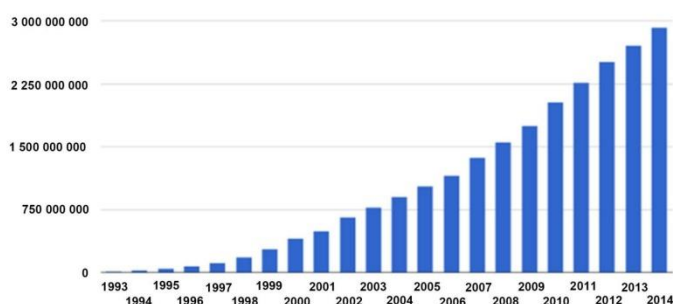


Figure 1. Number of global users [4].

simulate our approach proposal. The work will consist in presenting in Section 2 the previous works. Section 3 will present our mathematical model and its constraints (delay, packet loss and jitter) in which we will explain our algorithmic approach. We present in Section 4 the results of our simulations. In Section 5, we will compare our work with that of previous authors. The conclusion will come in Section 6.

2. Previous Work

In this section, we present the results of the simulations of different authors on energy consumption in SDN networks.

Several simulation works have been processed in the optimization of energy consumption in communication networks. This is the case for citing the authors [2] and [3]. In [2], the authors used a software approach based on an automatic On/Off control system to enable and disable router links. This is a test-bed that has been done in conventional networks. The authors chose in their network topology paths supported by the Cognitive Packet Network (CPN) protocol. It has been found that non-protocol defined routes are not degraded. The results of their approach gave 10% to 20% extinction of energy-saving links with QoS degradation (see figures below). These works of the authors, we reproduced them [2] (Figure 2).

Figure 2 showed that disabling some routers at instants 100 s, 300 s, 500 s, 700 s caused a delay. This is due to the fact that each time the links or routers are shut down, the CPN routing protocol looks for new favorable paths. And these paths being the shortest paths (instants 200 s, 400 s, 600 s) are quickly saturated. Only long paths are borrowed. Some packages are lost. In the authors' experience, the packets use in their header the extinction strategy based on the Cognitive Packet Network (CPN) protocol. Routers have become mere rest areas or roads for future destinations. This is where packages with experience look for favorable destinations. During shutdown, packets in the queue are destroyed (Figure 3).

While packages take time to find a way, some of them are lost. Because these have already reached the number of jumps to be made in the network (30 jumps

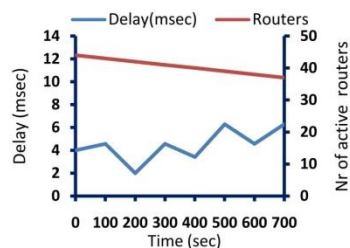


Figure 2. Delay-first experience.

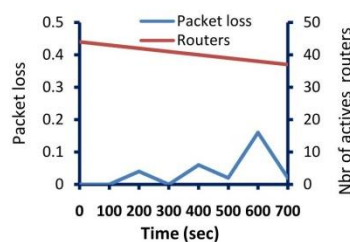


Figure 3. Packet loss-first experience.

maximum) beyond, the packets are destroyed to allow others to make their way. The successive extinction of the routers or the maximum number of links increases the delay due to congestion in the network thus causing packet losses that reach 20% (Figure 4).

Here, the jitter does not vary. The delay follows the same rhythm. Its value is constant of 0.08 (or 8%). We present now the simulations of the work of the authors of [3]. In [3], the authors proposed a software approach using the SPRING protocol to turn off links when they are not used. Their approach shows in the simulations that it was possible to turn off 44% of the links to save energy that we present below through captures:

Figure 5 simulations Results in Germany 50 (top) and Giant (bottom) networks [3]

In this work [3], we report their results on three points:

- ✓ **Impact on the network load:** it creates an additional cost of 18.56%. The paths on which packets are routed are not the shortest. But this extra cost is quickly canceled when the links are lit (Figure 5(a) and Figure 5(c)).
- ✓ **Impact on end-to-end delay:** the increase was 37% with Germany 50 while in most cases it is 20%.
- ✓ **Impact on packet loss:** no loss of packets with the algorithm used by the authors based on the Spring Protocol.

We used the idea of the authors of [3] to improve the work of the authors [2]. Our approach is software. It allowed us to disable the maximum of routers ports. Ports are disabled for adjacent routers. It is better than those of the previous

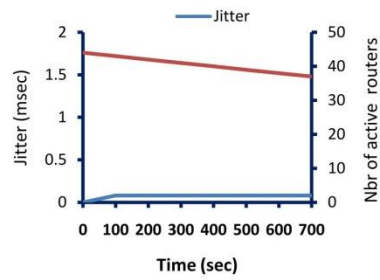


Figure 4. Jitter-first experience.

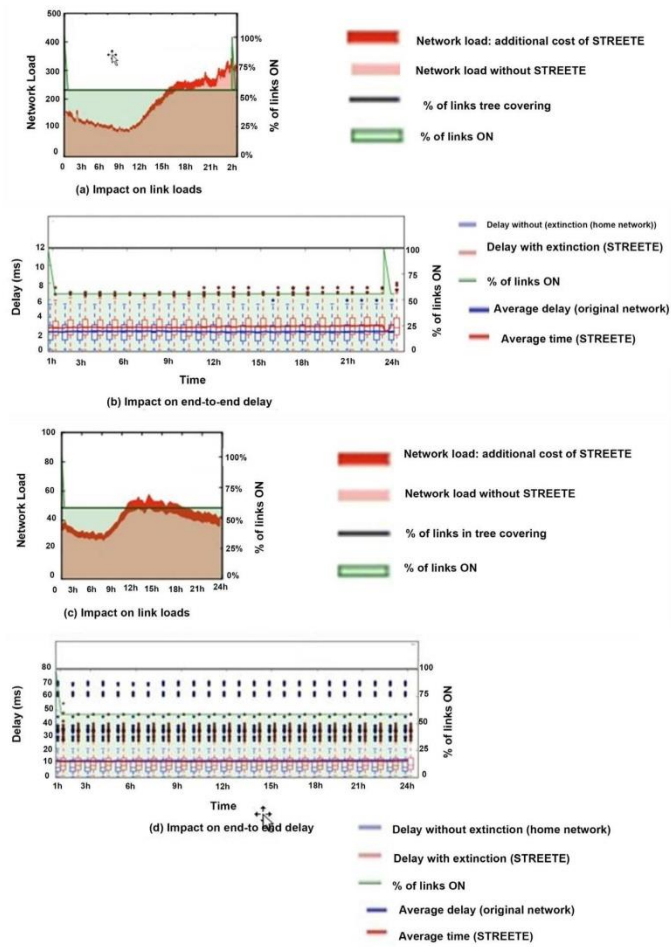


Figure 5. Simulations results in Germany 50 (top) and giant (bottom) networks [3].

ones. Because it offers us a double saving of energy saving (in ports and links). Our simulations will demonstrate them. Well before we remember our mathematical model and then follow our simulations.

3. Model Presentation

In this section, we present our mathematical model under constraint of delay, packet loss and jitter. We showed the principle of the extinction of the ports of the routers by our Modified SPRING Protocol [3] on a tree covering minimum weight.

Our model was obtained using the convexity theorem. Thus, the objective function obtained is:

$$\min(f(n(t),t)) = \min\left(\sum_{i \in n(t)} ((C_{ch} + Cp) - (a\ell(t) + b)) + \sum_{i,j \in n(t)} f(k(i,j,t))\right) \quad (1)$$

Under Constraint of:

$$q(f(t), R(F(t), n(t), k(t), t)) \leq Q(f(t)), \forall f(t) \in F(t) \text{ and:}$$

$$\sum f(t) \leq C_{i,j} \forall f(t) \in F(t) \text{ and } \text{Max}\ell(t) < \lambda_{\max}$$

$$k \in \{0,1\}$$

When the routers are turned on, only one port on each router is activated and the (n-1) others are turned off. The packets have at least one path on each router. This approach will influence the delay of transmission unlike that of the authors [2] and [3]. So we define our delay based on the number of disabled ports called $D\ell(t)$. Our delay will be obtained from the data of the work of [2].

So, we have:

$$D\ell(t) = \frac{\text{Delay}_n}{\text{Max}\ell(t)} \quad (2)$$

where $D\ell(t)$: delay when $\ell(t)$ ports are disabled.

Delay_n : Normal delay (value obtained from those of [2]).

$\text{Max}\ell(t)$: Maximum number of ports off on each router.

- **The ratings:** (Table 1)

We have found our mathematical model which minimizes, so we present our simulations on the basis of our Modified SPRING Protocol (MSP).

4. Our Simulation Results

Our extinction principle approach by Modified SPRING protocol are described in our algorithms and simulation results are presented in **Algorithm 1** and **Algorithm 2**.

Algorithm 1. Centralized SDN: <Reroute And Router Off>.

Data: List-of-Ports: [(0-4), (2-6)...] and the news! * Supplied by <Select Portsto Off> and Compute New Routes> respectively

1) Begin

2) The controller gives all adjacent nodes the new routes to use for the virtual paths;

3) The controller sees the network as a whole and informs neighboring routers to turn off all enabled ports except one;

Continued

-
- 4) Routers 0.1.3... send an (ACK) to the controller after the task is executed;
 - 5) The SDN controller identifies the lit ports and turns off the others;
 - 6) All routers apply the change to the forward tables for redirection by other ports;
 - 7) Nodes 0, 1, 2... stop sending IGP packets;
 - 8) Ports (0 - 1, 2 - 5...) stop and enter sleep mode to be turned off;
-

Algorithm 2. Main loop.

-
- 1) While energy efficient traffic engineering is active do
 - 2) Initialization of the selected ports to the number of lit ports;
 - 3) If new roads and all ports are on then
 - 4) <turn off the links to avoid circuits>;
 - 5) List of ports \leftarrow list of links;
 - 6) If the new routes (all ports turned on) then
 - 7) <Reroute And Router Off> (List-Of-Ports);
-

Table 1. Summary of the ratings.

$f(n(t), t)$	Power consumption in the subnet
λ_{\max}	Total number of ports on a router
C_i	Overall energy of the router
$a(t)+b$	Gain of energy obtained during the deactivation of the ports
$f(k(i, j, t))$	The state of ports in sub-networks at time t
q	Quality of service perceived by the user
$F(t)$	Set of all requests at the moment t
$f_{s,p}(t) \in F(t)$	Observed request of a user at the instant t from a point s to a p.
$R(F(t))$	Routing of all requests at time t (see the routing algorithm).
$n(t) \subseteq N(t)$	The subnet included in the network at time t
$Q(f(t))$	Quality of service set to not exceeded for all requests observed.
$D^l(t)$	delay when $l(t)$ ports are disabled
Delay _n	Normal delay
Max $l(t)$	Maximum number of ports off on each router

We have the results of our simulations and a comparative study of these results with the previous ones based on our algorithmic approach in OMNET ++ has been done (see [Figure 6](#)).

In this diagram, as we disable the ports of the routers, the delay decreases or even tends to zero. This is explained by our approach. There is at least one port on, thus establishing a link between different routers avoiding the waste of time finding a path ([Figure 7](#)).

[Figure 7](#) is the consequence of that ([Figure 6](#)). The delay decreases, resulting in a decrease in packet loss (relatively low). In our case, it is 10% ([Figure 7](#)).

Jitter, in our case it decreased doubly 0.04 ([Figure 8](#)) contrary to 0.08 of the authors of [2].

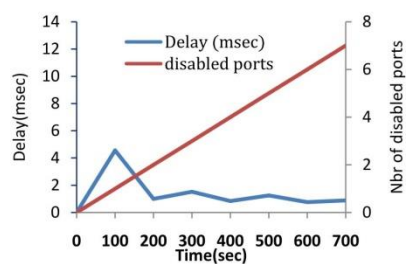


Figure 6. Delay based on the number of disabled ports.

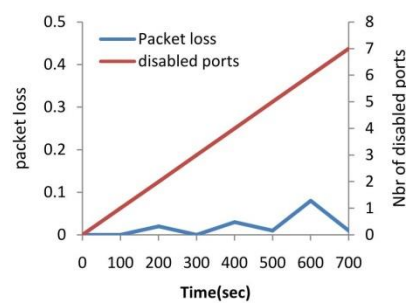


Figure 7. Loss of package based on the number of disabled ports.

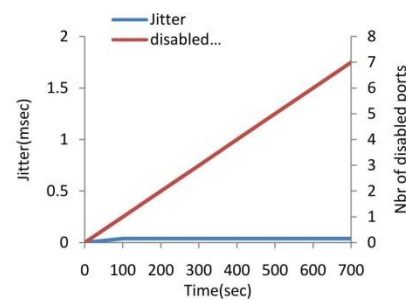


Figure 8. Jitter based on the number of ports disabled.

Given these results, a comparison of our model with that of the authors cited in our work is necessary.

5. Performance Evaluation

In this section, we compare our results with those of the authors cited in this paper.

We find that the results of the authors [3] are much better than that of the authors [2]. We recall that the authors of [2] worked in classical networks. The

efficiency in the work of [3] lies in the importance of SDN technology. The fact that the SDN has a global view of the network facilitates the definition of effective rules. The use of the SPRING algorithm with SDN allowed the authors of [3] to obtain better results. Ours view of the different figures, is still interesting than the previous two. Our Modified SPRING Protocol (MSP) -based approach minimizes energy with good quality of service (QoS) in terms of delay, packet loss and jitter. We achieved better packet forwarding times with less congestion, thus minimizing losses. Just like the authors of [3], we can also say that the loss of packets in our simulations are the same. We cannot minimize without degradation. It (packet loss) is 20% in [3], [2] and 10% in ours. Our delay is much better than previous authors.

6. Conclusions

We simulated our mathematical model with the OMNET++ simulator. The results show that the ports of the routers, when deactivated according to our energy saving strategy, have a good quality of service (QoS) in terms of delay that tends to zero, packet loss of the order of 10% and constant jitter of 4%.

The quality of service degradation can be related to several factors: network connectivity difficult at times, over-provisioning of the network causing configuration rules and correct routing problems. Because the size of the memory (TCAM) in which the rules used by SDN devices are stored is small. This orientation is interesting and could be developed in our future work. The advent of SDN by the programmability thus brings flexibility and good results in the management of the network.

Acknowledgements

The authors would like to thank the anonymous referee for his/her comments that helped us improve this article. The authors also wish to thank the Higher School of ICT (ESATIC) and the Doctoral School Polytechnic Institute F. H. Boigny of Yakro (INP-HB).

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Kra, L., Gondo, Y., Gooré, B.T. and Asseu, O. (2018) Contribution to the Optimization of the Energy Consumption in SDN Networks. *Journal of Sensor Technology*, 8, 1-9. <https://doi.org/10.4236/jst.2018.83005>
- [2] Gelenbe, E. and Silvestri, S. (2009) Optimisation of Power Consumption in Wired Packet Networks. In: Bartolini, N., Nikolettseas, S., Sinha, P., Cardellini, V. and Mahanti, A., Eds., *Telecommunications Engineering*, Springer Berlin Heidelberg, 717-729.
- [3] Carpa, R., Glück, O., Lefèvre, L. and Mignot, J.-C. (2015) STREETE: Traffic Engineering for Energy-Efficient Core Networks. *Conference Compas'2015*, Lille.

- [4] Project, R.T.S. (2014) Internetlivestats.
<http://www.internetlivestats.com/internet-users-by-country/2014/>
- [5] Chiaraviglio, E.L. and Mellia, M. (2009) How Much Can Internet Be Greened? Green Comon.
- [6] Bruce, N. and Christensen, K. (2005) Reducing the Energy Consumption of Networked Devices.