SUP'COM Université de Carthage

UNIVERSITÉ DE LORRAINE

École doctorale EDTIC **École doctorale IAEM Lorraine**

Allocation dynamique des ressources et gestion de la qualité de service dans la virtualisation des réseaux

THÈSE

présentée et soutenue publiquement le 14 Avril 2015

pour l'obtention du

Doctorat de l'Université de Lorraine
(mention Informatique)

et

Doctorat de l'École Supérieure des Communications de Tunis
(mention Technologies de l'Information et de la Communication)

par

Mohamed Said Seddiki

Composition du jury

Président :	Salah Benabdallah	Professeur à Tunis Business School
Rapporteurs :	Francine Krief	Professeur à ENSEIRB-MATMECA
	Noureddine Hamdi	Professeur à INSAT Tunis
Examinateur :	Tijani Chahed	Professeur à Telecom SudParis
Directeurs de Thèse :	Ye-Qiong Song	Professeur à l'Université de Lorraine
	Mounir Frikha	Professeur à Sup'Com Tunis

Laboratoire Lorrain de Recherche en Informatique et ses Applications (LORIA)
Laboratoire de Recherche en Réseaux Radio Mobile Multimedia (MEDIATRON)

Mis en page avec la classe thesul.

Remerciements

Je tiens à exprimer mes sincères remerciements et ma reconnaissance à mes directeurs de thèse Monsieur le Professeur Ye-Qiong Song et Monsieur le Professeur Mounir Frikha qui m'ont encadré et soutenu tout au long de ce travail de thèse. Leurs grandes qualités humaines, leurs conseils scientifiques, et leurs critiques constructives ont rendu ce travail particulièrement enrichissant. Je tiens à les remercier pour m'avoir communiqué leurs passions pour la recherche scientifique.

Je remercie vivement Madame Francine Krief, Professeur à l'École Nationale Supérieure d'électronique, Informatique, Télécommunications, Mathématique et Mécanique de Bordeaux et Monsieur Noureddine Hamdi, Professeur à l'Institut National des Sciences Appliquées et de Technologie de Tunis d'avoir accepté d'être mes rapporteurs et d'avoir bien voulu évaluer ce travail de thèse.

Je remercie également Monsieur Tijani Chahed, Professeur à Télécom SudParis et Monsieur Salah Benabdallah, Professeur à Tunis Business School pour leur participation au jury de cette thèse et pour le temps qu'ils ont bien voulu consacrer à l'évaluation de ce travail.

Je suis tout particulièrement reconnaissant envers Professeur Nick Feamster de Georgia Tech pour m'avoir accordé l'opportunité d'effectuer un stage de recherche d'un an au sein de son laboratoire de recherche à Atlanta, USA.

Mes remerciements vont également aux membres de l'équipe MADYNES du Loria, l'équipe MEDIATRON de Sup'Com et l'équipe GTNOISE de Georgia Tech avec qui c'était un bonheur de travailler. Je les remercie infiniment pour leurs amitiés et leurs apports scientifiques à mes travaux.

Je tiens à remercier mes parents, Ezeddine et Hela, et ma soeur Ons pour tout l'amour, le soutien et la confiance qu'ils m'ont apporté afin d'affronter ces années de labeur avec une grande confiance. Je leur serai toujours reconnaissant pour tout leurs efforts et leurs sacrifices. Du fond du coeur, merci.

Je tiens à remercier ma financé Linda qui a fait preuve de patience à mon égard. Elle a supporté mon indisponibilité et mes sautes d'humeur en m'écoutant et en me conseillant. Cette aide morale fut d'un grand réconfort. Merci d'être à mes côtés chaque jour et de m'apporter ton amour et ta gaieté.

Je tiens en dernier lieu à exprimer ma reconnaissance et ma gratitude à toutes les personnes qui ont contribué de près ou de loin à l'aboutissement de ce travail de thèse.

A mon père, à ma mère, à ma soeur et à ma chérie.
votre affection et votre soutien m'ont été d'un grand secours
tout au long de ma vie professionnelle et personnelle.
Je vous remercie du fond du coeur d'être toujours
à mes cotés et de me donner cette force pour continuer.
Je vous dédie ce travail en signe de reconnaissance.

Table des matières

Chapitre 1
Introduction et problématique **1**

1 Contexte et problématique générale 1
2 Problématique et contributions de la thèse 3
3 Structure de la thèse 6

Chapitre 2
Notions préliminaires et état de l'art

1 Introduction 10
2 Qualité de service dans les réseaux 10
2.1 Mécanismes de gestion de la QoS 10
2.1.1 La classification et la régulation du trafic 10
2.1.2 La politique d'ordonnancement des paquets 10
2.1.3 Le contrôle de congestion du trafic 12
2.1.4 Le contrôle d'admission des flux 14
2.2 Les métriques de QoS 14
2.3 Les modèles de QoS dans les réseaux IP 15
2.3.1 Le modèle "Integrated Services" 15
2.3.2 Le modèle "Differentiated Services" 15
2.3.3 Utilisation des modèles de QoS dans l'Internet actuel 16
3 La virtualisation des réseaux 16
3.1 Les réseaux Overlays 16
3.2 Les techniques de la virtualisation des réseaux 17
3.2.1 Techniques basées sur la virtualisation des protocoles 17
3.2.2 Techniques basées sur la virtualisation des machines 20
3.3 Les projets de recherche sur la virtualisation des réseaux 22
4 Les réseaux programmables 23
4.1 Les réseaux définis par logiciels (SDN) 23
4.1.1 OpenFlow 25
4.1.2 FlowVisor 26
4.2 Les routeurs programmables 27

5 L'approvisionnement des réseaux virtuels . . . 28
5.1 Formulation du problème . . . 31
5.2 La découverte de l'infrastructure physique . . . 32
5.2.1 La description des ressources . . . 32
5.2.2 Le regroupement des ressources . . . 33
5.3 Le mappage du réseau virtuel . . . 34
5.4 Allocation des ressources physiques pour le réseau virtuel . . . 36
6 Conclusion . . . 39

Chapitre 3
Gestion de la QoS dans un environnement domestique virtualisé

1 Introduction . . . 41
2 Le réseau Internet haut débit . . . 42
2.1 Internet haut débit par xDSL . . . 43
2.2 Internet haut débit par câble . . . 45
3 Virtualisation des passerelles domestiques . . . 45
4 SDN et réseaux domestiques connectés à Internet haut débit . . . 46
4.1 Gestion de la QoS dans les réseaux SDN . . . 47
4.2 Gestion de la QoS dans les réseaux domestiques . . . 49
5 FlowQoS : Gestion de la QoS dans un environnement domestique virtualisé . . . 50
5.1 Architecture de FlowQoS . . . 51
5.2 Le classificateur de flux . . . 52
5.3 Le lissage du trafic . . . 53
5.4 Evaluation de FlowQoS . . . 54
6 VidShaper : Allocation dynamique de la bande passante pour les flux de streaming vidéo adaptatif . . . 60
6.1 Architecture de VidShaper . . . 61
6.2 Evaluation de VidShaper . . . 62
7 Conclusion . . . 63

Chapitre 4
Modélisation de l'allocation des ressources par la théorie des jeux

1 Introduction . . . 67
2 Rappel sur la théorie des jeux . . . 68
2.1 Représentation d'un jeu . . . 68
2.1.1 Jeu sous forme extensive . . . 69
2.1.2 Jeu sous forme stratégique . . . 69
2.2 Stratégie pure, stratégie mixte et stratégie dominée . . . 70
2.3 Les différentes catégories de jeu . . . 70
2.3.1 Jeu coopératif et jeu non-coopératif . . . 70

2.3.2 Jeu simultané et jeu séquentiel . . . 71
2.3.3 Jeu à information complète et jeu à information incomplète . . . 71
2.3.4 Jeu à somme nulle et jeu à somme non-nulle . . . 71
2.4 L'équilibre de Nash . . . 71
3 Jeux non-coopératifs pour l'allocation dynamique des ressources . . . 72
3.1 Jeu de négociation des ressources . . . 73
3.2 Jeu d'allocation du nœud physique . . . 77
3.2.1 Modélisation du jeu . . . 77
3.2.2 Évaluation du jeu d'allocation du nœud physique . . . 79
3.3 Jeu d'allocation du lien physique . . . 82
3.3.1 Modélisation du jeu . . . 82
3.3.2 Évaluation du jeu d'allocation du lien physique . . . 84
4 Conclusion . . . 87

Chapitre 5
Allocation dynamique des ressources avec la théorie du contrôle

1 Introduction . . . 89
2 Le contrôle des systèmes dynamiques linéaires . . . 90
2.1 Le contrôle en boucle ouverte . . . 90
2.2 Le contrôle en boucle fermée . . . 91
2.3 Le concept de stabilité . . . 91
3 Le contrôle adaptatif de l'allocation dynamique de la bande passante . . . 92
3.1 Modélisation de l'approche . . . 92
3.2 Contrôleur du fournisseur de services . . . 94
3.2.1 L'estimateur . . . 96
3.2.2 Le demandeur . . . 97
3.3 Contrôleur du fournisseur de l'infrastructure physique . . . 97
3.3.1 Algorithme d'allocation de la bande passante . . . 98
3.3.2 Exemple d'allocation de la bande passante par le FIP . . . 100
3.4 Evaluation de l'approche d'allocation dynamique de la bande passante . . . 101
4 Conclusion . . . 106

Chapitre 6
Conclusion et perspectives

1 Synthèse des contributions . . . 109
2 Perspectives . . . 111

Bibliographie **113**

Table des figures

1.1 Schéma d'un environnement réseau virtualisé 3

2.1 Schéma du mécanisme d'ordonnancement WFQ 11
2.2 Evolution de la fenêtre de congestion avec TCP Tahoe et TCP Reno [Kurose and Ross, 2009] 12
2.3 Scénario de partage de la capacité de plusieurs liens physiques entre trois sources utilisant l'algorithme des courbes loss-load 13
2.4 Regroupement des PCs en VLANs 18
2.5 VPN entre plusieurs sites distants d'une entreprise donnée 19
2.6 La technique du cloisonnement 20
2.7 Schématisation de la virtualisation complète 21
2.8 La technique de la para-virtualisation 22
2.9 Les différentes couches des réseaux définis par logiciels (SDN) 24
2.10 Architecture de FlowVisor [Sherwood et al., 2009] 26
2.11 Architecture d'un routeur programmable [Chao and Liu, 2007] 28
2.12 Schéma d'une architecture réseau virtualisée 29

3.1 Le fonctionnement du streaming vidéo adaptatif via HTTP 44
3.2 Architecture de L'ADSL 45
3.3 Les performances de Pantou en filaire et en sans fil. 47
3.4 Le débit maximal de OVS en filaire et en sans fil 48
3.5 Architecture de FlowQoS 51
3.6 La classification des paquets avec FlowQoS 52
3.7 Topologie des commutateurs virtuels effectuant le lissage du trafic à l'intérieur du routeur domestique 53
3.8 Montage expérimental d'évaluation de FlowQoS 55
3.9 Comparaison de la sélection du débit d'encodage du flux de streaming vidéo adaptatif au cours du temps avec et sans FlowQoS 56
3.10 Performance du débit de téléchargement streaming vidéo adaptatif avec et sans FlowQoS 56
3.11 L'évolution au cours du temps du débit de téléchargement des flux en concurrence pour la bande passante disponible avec et sans FlowQoS 57

3.12 Les performances en termes de délais de la VoIP avec et sans FlowQoS 58
3.13 Les performances en termes de gigue de la VoIP avec et sans FlowQoS 59
3.14 Comparaison des délais de classification des paquets avec un contrôleur local (Raspberry PI) et un contrôleur distant (un serveur déployé à la boucle locale) . 60
3.15 Architecture de VidShaper pour effectuer la limitation dynamique du trafic entre plusieurs lecteurs de streaming vidéo adaptatif 61
3.16 Le montage expérimental de l'évaluation de VidShaper 63
3.17 Variation du débit d'encodage pour le flux de streaming vidéo adaptatif avec et sans FlowQoS 64
3.18 Les performances en termes de débit de téléchargement de la vidéo avec et sans VidShaper 65

4.1 Forme extensive d'un jeu 69
4.2 Le cadre des jeux non-coopératifs pour l'allocation dynamique des ressources . . 73
4.3 Topologie de l'infrastructure physique partagée 79
4.4 Evolution au cours du temps de la fraction du nœud N_1 reçue par chaque FS . . 80
4.5 Le délai de traitement moyen des paquets d'un réseau virtuel déployé par FS_1 dans le nœud N_1 81
4.6 La fraction moyenne d'un nœud physique reçue par chaque FS 81
4.7 Evolution au cours du temps de la bande passante recommandée, requise et espérée pour chaque FS 85
4.8 Répartition de la disposition à payer au cours du temps 85
4.9 Bande passante recommandée, allouée et espérée pour chaque FS 86
4.10 Moyenne de la bande passante allouée pour chaque FS 86

5.1 Représentation d'un contrôle à boucle ouverte 91
5.2 Représentation d'un contrôle à boucle fermée 91
5.3 Architecture proposée du système de contrôleurs 93
5.4 Architecture du sous-contrôleur du VN 96
5.5 Scénario de partage de l'infrastructure physique entre les trois VNs 101
5.6 Scénario de capture de la trace simulée avec OPNET 102
5.7 Les délais mesurés et ciblés des paquets de $VN_{1,1}$ à chaque intervalle de contrôle 103
5.8 Evolution de la portion de la bande passante du lien $l_{1,1}$ demandée par $VN_{1,1}$ . . 104
5.9 Evolution de la bande passante du lien physique $l_{1,1}$ allouée aux trois VNs 104
5.10 L'impact du paramètre k du modèle des courbes *loss-load* dans l'allocation de la bande passante 105
5.11 La différence entre les portions moyennes de la bande passante demandée et celle allouée en utilisant les algorithmes PSA et FAA 105
5.12 Comparaison des variances de chaque ensemble de moyennes des algorithmes PSA et FAA 106

Liste des tableaux

3.1 Comparaison de l'utilisation du CPU et de la mémoire de OpenWrt sans et avec la topologie des deux OVSs 58
3.2 Comparaison de l'utilisation du CPU et de la mémoire de FlowQoS et d'un script de lissage du trafic 59

4.1 Forme stratégique du jeu du dilemme du prisonnier 70
4.2 Les différentes notations utilisées dans notre modélisation 74
4.3 La relation entre les gains et les stratégies du FS et du FIP 75

5.1 Les différentes notations utilisées dans le système à contrôleurs 95
5.2 Exemple de partage de la capacité d'un lien physique entre trois VNs 101

Résumé

Bien qu'Internet soit considéré comme le grand succès de ces dernières années, il est devenu une infrastructure critique à cause de l'absence de changements dans le réseau cœur et de la rigidité des équipements déployés. La mise en place et le déploiement des nouveaux services réseau sont devenus difficiles et coûteux. La virtualisation des réseaux a été présentée comme un nouveau paradigme pour palier aux problèmes de l'architecture actuelle de l'Internet.
Dans ce travail de thèse, nous présentons la virtualisation des réseaux et les réseaux définis par logiciels (SDN) comme solution avec laquelle les fournisseurs de services peuvent offrir, au travers des réseaux virtuels (VN), des nouveaux services aux utilisateurs avec une meilleure qualité de service, tout en optimisant l'utilisation des ressources réseaux physiques.
La première contribution consiste à démontrer le potentiel de SDN dans la gestion de la QoS dans le contexte d'un réseau domestique virtualisé. Nous proposons et implémentons le mécanisme "FlowQoS" qui peut être déployé par un fournisseur d'accès Internet au niveau de la boucle locale ou bien dans la passerelle domestique. Les mesures des performances montrent que cette solution permet de partager la bande passante entre plusieurs applications selon la configuration définie par l'utilisateur pour garantir la QoS pour chaque trafic actif.
La seconde contribution est une modélisation, par la théorie des jeux, de l'interaction entre les fournisseurs de services et les fournisseurs de l'infrastructure pour le partage dynamique de l'infrastructure physique entre plusieurs VN avec différents besoins en QoS. Il s'agit d'un ensemble de jeux non-coopératifs pour modéliser la phase de négociation et celle de l'allocation dynamique des nœuds et des liens physiques pour chaque VN déployé.
La troisième contribution porte sur une approche prédictive qui permet d'offrir un contrôle adaptatif de l'allocation de bande passante dans le but de réduire les délais des paquets d'un VN sur chaque lien physique.
Ces deux dernières contributions offrent des modèles de partage dynamique des ressources d'une infrastructure physique tout en garantissant la QoS pour chaque VN.

Mots-clés : Virtualisation des réseaux, Réseaux définis par logiciels, Qualité de service, Allocation des ressources.

Abstract

Internet has been successful in the recent years. The critical infrastructure of the internet has become stagnant due to the absence of changes in the core networks and stiffness of deployed equipment. It has become difficult and expensive to deploy new network services. Network virtualization is a new paradigm to overcome this problem.
In this thesis, we present network virtualization and Software Defined Networking (SDN) as a solution that can be used by service providers. It enables them to provide new services to users through virtual networks (VNs) with better quality of service while optimizing the use of physical network resources.
Firstly, we demonstrate the potential of SDN in the QoS management of a virtualized home network (VN). We propose and implement "FlowQoS", a mechanism that can be deployed by an Internet Service Provider in the last-mile hop or in the home gateway. Performance measurements show that this solution can share bandwidth between applications according to user-defined configuration to guarantee QoS for each active traffic.
The second contribution is modeling the interaction between service providers and infrastructure providers using game theoretic framework to offer dynamic sharing of physical infrastructure across multiple VN with different QoS requirements. We present a set of non-cooperative games to model the negotiation phase and the dynamic allocation of nodes and physical links for each deployed VN.
Finally we focus on a predictive approach that allows an adaptive control of bandwidth allocation in order to reduce the packet delays for a given VN on each physical link.
The last two contributions offer dynamic sharing models of physical infrastructure resources while guaranteeing the QoS for each VN.

Keywords : Network virtualization, Software Defined Networking, Quality of service, Resource allocation.

Liste des publications et des soumissions

Revues Internationales

[Seddiki et al., 2014a] M. S. Seddiki, M. Shahbaz, S. Donovan, S. Grover, M. Park, N. Feamster, Y. Song, FlowQoS : Providing Per-Flow Quality of Service for Broadband Access Networks, soumis au journal *Elsevier Computer Networks*.

[Seddiki et al., 2013a] M. S. Seddiki, M. Frikha, and Y. Song, A Non-cooperative Game-theoretic Framework for Resource Allocation in Network Virtualization, *The Journal of Telecommunication Systems* (Springer), à paraître dans le volume 76.

Conférences Internationales

[Seddiki et al., 2014b] M. S. Seddiki, M. Shahbaz, S. Donovan, S. Grover, M. Park, N. Feamster, Y. Song, FlowQoS : QoS for the Rest of Us, *ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN 2014)*, Chicago, Illinois, USA, pp. 207-208. *(Poster)*

[Seddiki et al., 2013b] M. S. Seddiki, M. Frikha, B. Nefzi, and Y. Song, Automated Controllers for Bandwidth Allocation in Network Virtualization, *The 32nd IEEE International Performance Computing and Communications Conference (IPCCC 2013)*, San Diego, California, USA, pp.1-7.

[Seddiki et al., 2013c] M. S. Seddiki, M. Frikha, and Y. Song, Dynamic Node Allocation in Network Virtualization, *The 2013 International Conference on High Performance Computing & Simulation (HPCS 2013)*, Helsinki, Finland, pp.635-639. *(Poster)*

[Seddiki et al., 2013d] M. S. Seddiki, B. Nefzi, Y. Song, M. Frikha, Queuing analysis of dynamic resource allocation for virtual routers, *The 2013 IEEE symposium on Computers and Communications (ISCC 2013)*, Split, Croatia, pp. 771-776.

[Seddiki et al., 2012a] M. S. Seddiki and M. Frikha, A Non-cooperative Game Theory Model for Bandwidth Allocation in Network Virtualization, *The 16th International Telecommunications Network Strategy and Planning Symposium (NETWORKS2012)*, Rome, Italy, pp. 1-6.

[Seddiki et al., 2012b] M. S. Seddiki and M. Frikha, Resource Allocation for Virtual Routers Through Non-cooperative games, *The 21st International Conference on Computer Communication Networks (ICCCN2012)*, Munich, Germany, pp. 1-6.

Glossaire

FS/SP : Fournisseur de services / *Service provider*
FIP/InP : Fournisseur de l'infrastructure physique / *Infrastructure provider*
VN : Réseau virtuel / *Virtual network*
SDN : *Software Defined Networking*
QoS : Qualité de Service / *Quality of Service*
QoE : Qualité d'éxperience /*Quality of Experience*
AS : Système autonome /*Autonomous System*
MPLS : *MultiProtocol Label Switching*
FAI : Fournisseur d'accès Internet
ADSL : *Asymmetric Digital Subscriber Line*
VNE : *Virtual Network Embedding*
DSLAM : *Digital Subscriber Line Access Multiplexe*r
TCP/IP : *Transmission Control Protocol / Internet Protocol*
ATM : *Asynchronous Transfer Mode*
HTTP : *HyperText Transfer Protocol*
VoIP : Voix sur IP /*Voice over IP*
VoD : Vidéo à la demande /*Video on demand*
DNS : *Domain Name System*
WFQ : *Weighted Fair Queuing*
IntServ : *Integrated Services*
RSVP : *Resource Reservation Protocol*
DiffServ : *Differentiated Services*
URL : *Uniform Resource Locator*
ICMP : *Internet Control Message Protocol*
CDN : *Content Delivery Network*
VPN : Réseau Privé Virtuel /*Virtual Private Network*
VLAN : Réseau local virtuel /*Virtual LAN*
OVS : *Open vSwitch*
CPU : *Central Processing Unit*
ASIC : *Application-Specific Integrated Circuit*
IETF : *Internet Engineering Task Force*

Chapitre 1

Introduction et problématique

Sommaire

1 Contexte et problématique générale 1
2 Problématique et contributions de la thèse 3
3 Structure de la thèse 6

1 Contexte et problématique générale

Un réseau de communication est par définition un ensemble de ressources matérielles et logicielles mis en place pour offrir aux usagers un ensemble de services. Avec l'évolution des services de télécommunications et des trafics de données multimédia, les opérateurs ont déployé plusieurs technologies dans le but d'augmenter la capacité et les fonctionnalités des réseaux. La gestion des ressources d'un réseau garantissant une qualité de service (QoS) adéquate est devenue un enjeu majeur lors de la planification du réseau.

Bien qu'Internet soit largement considéré comme le grand succès de ces dernières années, il est devenu une infrastructure critique en raison de son ossification [Turner and Taylor, 2005]. Cette ossification est principalement causée par l'absence de changement dans le réseau cœur et par la rigidité des équipements déployés. Elle rend la mise en place et le déploiement de nouveaux services réseaux difficiles et coûteux. D'autre part, avec l'émergence des services multimédia, la gestion de la qualité de service dans le réseau Internet actuel est devenue une tâche difficile. En effet ces nouveaux services comme la vidéo à la demande (VoD) ou la téléphonie sur IP (VoIP) nécessitent différents types de QoS que l'architecture actuelle d'Internet ne peut offrir. Cette QoS comporte divers paramètres tels que la fiabilité, le débit ou encore le délai de bout en bout. Le service "*best effort*" ne permet d'offrir une QoS que lorsqu'il n'y a aucune congestion dans aucun lien entre le serveur et l'utilisateur final. Puisqu'on ne peut pas prédire une congestion provoquant l'augmentation des délais ou des pertes de paquets, cette QoS ne peut pas être garantie tout le long de la session. Les services multimédia sont très sensibles à ces facteurs. Les problèmes de résistance au facteur d'échelle de IntServ et la gestion de la QoS par classe de service dans DiffServ ne permettent pas de garantir cette QoS.

La virtualisation est une technique bien qu'ancienne mais très efficace pour consolider les ressources offrant une abstraction qui masque les détails d'un équipement. La virtualisation des réseaux a été présentée comme un nouveau paradigme pour les nouvelles architectures réseaux et pour l'Internet du futur. Elle permet d'offrir une diversité de réseaux en masquant l'hétérogénéité de l'infrastructure physique et une flexibilité en contournant la rigidité des équipements réseaux.
La virtualisation des réseaux est définie comme une nouvelle technologie qui permet la segmentation logique des liens et des nœuds physiques d'un réseau en des nœuds et des liens virtuels. Ceci permet de créer des réseaux logiques appelés réseaux virtuels en interconnectant les nœuds virtuels par des liens virtuels. La virtualisation des réseaux fait preuve de beaucoup de promesses garantissant la qualité de service requise [Schaffrath et al., 2009, Carapinha and Jiménez, 2009]. Elle permet une gestion et un contrôle optimisé de l'infrastructure physique partagée par plusieurs services. En effet, un fournisseur de services peut déployer un mécanisme de gestion de la QoS spécifique pour chaque type de service déployé. Il peut aussi agir sur l'allocation de la bande passante de chaque lien physique utilisé par le réseau virtuel (VN) déployé. La flexibilité introduite par cette technologie permet aux FSs la gestion de la QoS de bout en bout.
Aujourd'hui les fournisseurs d'accès à Internet (FAIs) ont deux rôles à jouer. Le premier consiste à gérer leur infrastructure du réseau. Le second est dédié aux services sur Internet à offrir aux utilisateurs finaux.
La virtualisation des réseaux introduit un nouveau modèle d'affaires (*business model*) qui permet d'avoir deux acteurs distincts. Le premier acteur est le fournisseur de l'infrastructure physique (FIP) qui est le propriétaire de l'infrastructure réseau. Il est responsable du déploiement et du maintien des ressources physiques du réseau (routeurs, liens, etc). Le second acteur est le fournisseur de services (FS) qui déploie des protocoles en louant les ressources d'un ou de plusieurs fournisseurs d'infrastructure physique pour créer un VN. Ce dernier ne possède pas d'infrastructure réseau. Il a la responsabilité de délivrer des services de bout en bout aux différents utilisateurs. Le FIP permet aux différent FSs de partager son infrastructure grâce à des interfaces programmables. Il peut rejeter la demande d'un FS si un accord ne peut être conclu entre eux suite à une sur-utilisation des ressources. À titre d'exemple, imaginons qu'un fournisseur de service de vidéo à la demande (comme Netflix) déploie un équipement chez l'utilisateur pour lui permettre de visionner des vidéos à sa demande. Ce FS loue l'infrastructure physique de différents FIPs (comme SFR, Orange, etc) pour créer des réseaux virtuels et fournir ce service de bout en bout avec garantie de service. La figure 1.1 présente un exemple d'environnement réseau virtualisé où deux fournisseurs de services louent des ressources de deux fournisseurs d'infrastructure afin de créer à chacun son propre réseau virtuel sur lequel il déploie un service de bout en bout. La virtualisation des réseaux vise à offrir une meilleure flexibilité supportant plusieurs topologies et mécanismes de routage et de transfert de paquets. Elle permet à une configuration du VN d'être indépendante de celle des autres VNs. Son objectif est d'offrir une meilleure gestion du réseau tout en maximisant le nombre de VNs qui coexistent sur une même infrastructure physique. Elle permet également d'offrir une isolation des flux de chaque VN pouvant être créé sur plusieurs infrastructures réseaux hétérogènes.

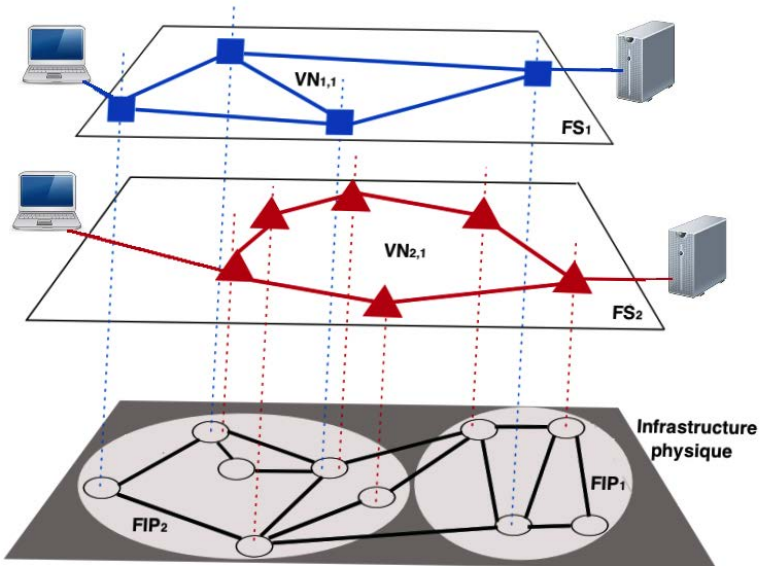


FIGURE 1.1 – Schéma d'un environnement réseau virtualisé

Les réseaux définis par logiciels (*Software-Defined Networking, (SDN)*) sont une nouvelle technologie qui permet d'introduire plus d'intelligence dans les réseaux. Elle offre aux FSs la possibilité de mieux contrôler les différentes composantes de leur réseau. Cette technologie permet d'augmenter l'efficacité du réseau physique pour que leur infrastructure logique soit plus fonctionnelle. L'utilisation de SDN permet un contrôle centralisé sur la programmation des différentes composantes du réseau. Dans une architecture SDN, le plan de contrôle et le plan de données d'un nœud physique sont découplés. En conséquence, cette flexibilité offre aux fournisseurs de services la possibilité de programmer et d'automatiser le contrôle du réseau. Ils peuvent ainsi répondre rapidement aux besoins changeants en QoS des services déployés sur un VN.

Dans ce travail de thèse, nous nous intéressons aux réseaux de commutation de paquets et à la QoS dans ce type de réseaux. Nous défendons l'idée que la virtualisation des réseaux et le SDN constituent des solutions efficaces gérant aux mieux la QoS dans les réseaux. En effet, avec ces deux technologies, les FSs peuvent offrir de nouveaux services aux utilisateurs avec une QoS meilleure que le "*best effort*" offert par l'Internet d'aujourd'hui.

2 Problématique et contributions de la thèse

Le réseau IP actuel présente plusieurs limitations liées au fait que les routeurs déployés sont plus ou moins programmables. Ses fonctionnalités sont parfois limitées en termes de gestion du trafic. De plus, ces entités présentent un comportement relativement statique, étant donné qu'ils n'effectuent que seulement les tâches pour lesquelles ils sont conçus. Par ailleurs, ces tâches qui diffèrent d'un constructeur à un autre engendrent des problèmes d'interopérabilité. On peut déduire à ce niveau, qu'avec l'évolution des réseaux à haut débit et la prolifération des terminaux fixes et mobiles, l'architecture actuelle ne peut plus répondre de façon efficace aux besoins des utilisateurs.

Deux problèmes majeurs, concernant le réseau Internet contemporain, sont mis en exergue. Le premier problème concerne la mise en place et le déploiement de nouveaux services réseaux. Ce type de déploiement s'avère très difficile et coûteux, et ne peut pas garantir la qualité de service requise. Cependant, avec l'apparition de nouveaux services que l'on peut déployer sur Internet, comme les services multimédia, les besoins en termes de QoS deviennent prépondérants et décisifs. En plus, Internet est constitué de nombreux systèmes autonomes (ASs) gérés par différentes organisations qui collaborent ensemble pour garantir la QoS de bout-en-bout. Ceci n'est pas une tâche facile parce que les méthodes de collaboration et la QoS à travers plusieurs ASs sont actuellement mal définies. À titre d'exemple, un fournisseur de vidéos à la demande (VoD) comme Canal+ doit s'assurer que les ASs par lesquels son trafic transite arrivent à subvenir à ses besoins en QoS.
Dans certaines situations, une demande importante en débit peut ne pas être satisfaite par ces ASs. Ceci engendre une détérioration du service au niveau de l'utilisateur final. D'une part, le fournisseur de VoD n'a pas le contrôle sur l'infrastructure pour pallier à ces problèmes. D'autre part, les mécanismes actuels de gestion de la QoS utilisés dans les ASs, comme DiffServ [Blake et al., 1998], traitent la QoS par classe de service et non par flux individuel de chaque utilisateur final. Ces déficiences liées à la QoS restent de nos jours une problématique ouverte. Le besoin d'offrir une QoS de bout en bout permettant aux fournisseurs de services de garantir une meilleure qualité d'expérience (QoE) pour les utilisateurs finaux se fait ressentir. Le travail présenté dans le cadre de cette thèse tente de trouver des solutions à ces problèmes.

Le second problème est celui de l'équilibrage du coût et de la tarification [Frieden, 2014] entre les fournisseurs de services et les opérateurs réseaux. Cette situation entraine plusieurs conflits mutuels. Les différends qui opposent Free à Youtube en France et Verizon à Netflix aux Etats-Unis sont une illustration de ce conflit. Pour mieux expliquer ceci, revenons en détail sur l'exemple du conflit Free-Youtube. Le problème provient du fait que Free est appelé à fournir le service Youtube à ses utilisateurs finaux, cependant, Youtube exige de ces derniers d'utiliser son service de streaming vidéo. Pourtant, dans ce scénario Free paye toujours Youtube pour raccorder ses utilisateurs. Lorsque Free a décidé d'arrêter le payement du raccordement de ses utilisateurs en exigeant une rémunération de l'excès de trafic qu'il génère, le fournisseur de services vidéo a sanctionné les utilisateurs de Free en provoquant une latence significative lors de l'accès à la plateforme.

Le travail effectué dans le cadre de cette thèse concerne la virtualisation des réseaux comme solution à ce type de problèmes rencontrés souvent dans le réseau IP actuel. Nous nous proposons dans cette thèse d'amener quelques éléments de réponse aux déficiences qui apparaissent, de plus en plus, de façon hétérogène dans les réseaux.
La thématique autour de laquelle s'articule cette étude concerne la virtualisation des réseaux comme solution possible palliant le manque d'efficacité dans les réseaux actuels. Pour cela, on se propose d'introduire de nouveaux mécanismes tenant compte des spécificités des services dans le but d'atteindre le niveau de performance requis. Dans ce travail nous traitons les problèmes de gestion de la QoS dans les réseaux de paquets. SDN et la virtualisation des réseaux sont des technologies conçues pour répondre à cette problématique.

Comme nous l'avons expliqué précédemment, le fournisseur de services est responsable des protocoles créant plusieurs réseaux virtuels (VNs) consolidant les infrastructures physiques de plusieurs FIPs. Les FS offrent des services de bout en bout aux différents utilisateurs. Chaque service s'exécute sur un réseau virtuel où le fournisseur de services définit ses besoins en termes de QoS.
La segmentation logique des nœuds et des liens physiques est une tâche complexe. Cette opération est couramment appelée approvisionnement des réseaux virtuels (*Virtual Network Embedding VNE*). Elle comprend généralement trois étapes [Esposito et al., 2013]. La première est la découverte des ressources du réseau physique. Chaque FIP surveille la charge de son infrastructure physique et informe les FSs sur l'usage et les performances de chaque lien et nœud. La seconde étape est le mappage du réseau virtuel. Cette étape est effectuée par le FS qui définit le chemin virtuel par lequel transitent tous les paquets d'un réseau virtuel. Il s'agit de l'étape qui choisit les nœuds et les liens physiques qui vont êtres utilisés par le VN. Ce choix dépend de la disponibilité des ressources du réseau physique. Cette étape est considérée comme la plus difficile, car il est nécessaire de combiner les deux contraintes de nœud et du lien. La troisième et dernière étape est l'allocation des ressources physiques pour le réseau virtuel. Cette étape est effectuée par le FIP lors de la réception de toutes les demandes des FSs. De nombreux travaux de recherche ont considéré l'allocation statique où le FIP alloue une portion de chaque nœud et de chaque lien pour un VN déployé par un FS [Haider et al., 2009]. Le manque d'adaptation des ressources du réseau physique offertes pour chaque VN à la dynamique de son flux engendre une sous-utilisation de ces ressources. À titre d'exemple, pour un fournisseur de vidéos à la demande, le nombre d'utilisateurs finaux qui utilisent leur service augmente le soir et diminue pendant la journée. Donc il est nécessaire de libérer les ressources et les réutiliser lorsqu'il y a un besoin.

Nous défendons la thèse de rendre cette allocation dynamique pour offrir une meilleure QoS pour les réseaux virtuels tout en maximisant l'utilisation des ressources. En effet, cette approche permet d'augmenter les bénéfices du FIP et d'offrir un meilleur service sur ses réseaux virtuels déployés au profit du FS.

Ce travail vise à mettre en place plusieurs solutions à la fois théoriques et techniques dans le but de : (i) allouer dynamiquement des ressources pour les réseaux virtuels qui coexistent sur une même infrastructure physique, (ii) optimiser les performances de l'infrastructure physique garantissant un niveau de QoS adéquat pour les réseaux virtuels. Nous avons découvert que la même idée apparaît aussi dans le *Framework* ACTN pour l'abstraction et le contrôle des réseaux de transport, présenté récemment dans le travail en cours de l'IETF [Ceccarelli et al., 2014]. Dans ce *draft*, les auteurs ont montré la faisabilité et la mise en œuvre d'un cadre qui permet le partage dynamique d'une ou de plusieurs infrastructures réseaux entre les FSs à l'aide de SDN. Ils présentent plusieurs types de contrôleurs qui permettent une granularité fine lors de l'allocation des nœuds et des liens physiques entre plusieurs VNs. Le *Framework* proposé adopte le même *business model* que le nôtre, où différents fournisseurs de services louent une partie du réseau physique de plusieurs fournisseurs d'infrastructures, afin de délivrer un ou plusieurs services à des utilisateurs finaux.
Nous partons d'un exemple simple où plusieurs fournisseurs de services déploient leurs serveurs

dans un centre de données (*data center*) directement connecté à plusieurs opérateurs réseaux comme Orange ou Free. Ces derniers sont à leurs tours connectés aux utilisateurs finaux. Ces utilisateurs sont soit des utilisateurs résidentiels fixes ou bien des utilisateurs mobiles qui payent les différents FSs pour un service donné (par exemple la VoD). Ces opérateurs réseaux se concentrent seulement sur la simple tâche de gestion de l'infrastructure physique du réseau. Quant aux FSs, ils déploient plusieurs protocoles afin de créer des réseaux virtuels dédiés à l'acheminement des paquets pour un service donné. La question critique est comment partager les ressources physiques limitées (nœuds et liens) d'un fournisseur d'infrastructure entre plusieurs fournisseurs de services qui ont des contraintes de qualité de service. Chacun cherche à consommer toutes les ressources disponibles pour délivrer au mieux un service aux utilisateurs finaux. Ce partage doit tenir compte du comportement dynamique de chaque flux de données tout en étant équitable.

Nos contributions dans le cadre de la thématique de la virtualisation des réseaux peuvent être résumées comme suit :

— une solution implémentée qui démontre la faisabilité technique de SDN et de la virtualisation des réseaux dans la gestion de la QoS dans un réseau domestique connecté à Internet haut débit. L'idée est de déléguer les fonctionnalités de la classification des flux et la limitation du trafic à un contrôleur SDN. Ce contrôleur peut être déployé par le FAI au niveau du réseau d'accès (au niveau de la boucle locale) ou bien connecté directement à la passerelle du réseau domestique comme la boxe ADSL. [Seddiki et al., 2014a, Seddiki et al., 2014b]

— une modélisation en deux étapes de l'interaction entre le fournisseur de services et le fournisseur de l'infrastructure physique pour le partage des ressources (nœuds et liens). La première est la négociation des ressources. La seconde étape est l'approvisionnement de la bande passante au niveau du lien et des cycles du processeur et des mémoires au niveau du nœud. À travers cette modélisation, nous pouvons partager équitablement et efficacement l'infrastructure physique entre plusieurs VNs. Cette solution peut être implémentée comme un système d'aide à la décision pour un FS et un système de facturation pour le FIP lors du partage des ressources entre plusieurs FSs. [Seddiki et al., 2013a, Seddiki et al., 2012a, Seddiki et al., 2012b]

— une approche prédictive qui permet de trouver le besoin en termes de bande passante pour chaque VN lors du partage du lien physique. Cette dernière dépend de l'estimation des performances de chaque VN et des allocations actuelles et passées de la bande passante. L'objectif est d'offrir un contrôle adaptatif de l'allocation de bande passante tout en maintenant la QoS offerte à chaque VN. Cette solution peut être considérée comme un système de monitoring et de contrôle des performances de chaque VN. [Seddiki et al., 2013b]

3 Structure de la thèse

Ce manuscrit présente nos contributions dans le cadre de l'allocation dynamique des ressources et la gestion de la qualité de service dans la virtualisation des réseaux. Ces contributions sont structurées en quatre chapitres.

Le chapitre 2 est consacré à la présentation des notions préliminaires et de l'état de l'art sur la virtualisation des réseaux et SDN. Nous présentons les technologies de la virtualisation des réseaux ainsi que les travaux existants pour l'approvisionnement des réseaux virtuels. La virtualisation du réseau offre plus de flexibilité et une meilleure gestion des réseaux à commutation de paquets. Elle permet de créer plusieurs réseaux virtuels hétérogènes (VN) qui coexistent sur la même infrastructure physique. Le défi majeur de cette technologie est lié au problème d'approvisionnement des VNs qui traite la cartographie et le partage efficace des ressources physiques.

Le chapitre 3 est dédié à la présentation d'une solution pour les réseaux domestiques connectés à Internet haut débit. Cette solution est basée sur la virtualisation des réseaux. Le but de cette contribution est d'offrir la possibilité à un utilisateur de gérer la QoS et contourner les limitations des réseaux domestiques en termes de ressources. Ce dernier peut spécifier pour chaque application la fraction de la bande passante qu'il veut allouer. L'objectif de ce travail est de démontrer la faisabilité de l'utilisation de SDN et la virtualisation des réseaux pour la gestion dynamique de la QoS dans une infrastructure de petite échelle.

Nous proposons, dans le chapitre 4, un cadre basé sur la théorie des jeux afin d'allouer dynamiquement les ressources pour plusieurs FSs et d'assurer la gestion et l'approvisionnement de ces ressources dans une infrastructure réseau virtualisée. Nous proposons une approche en deux étapes, fondée sur la théorie des jeux non-coopératifs. La première étape concerne la phase de négociation pour le partage des ressources où le FS demande à plusieurs FIPs une fraction de chaque lien et de chaque nœud. Chaque FIP décide d'accepter ou de refuser la demande lorsque le FS entraîne une congestion du nœud ou du lien. La seconde étape concerne le partage des ressources. Cette étape est composée de deux jeux non-coopératifs, le premier pour l'allocation de nœud et le second pour l'allocation de lien. Nous avons évalué notre approche par des simulations numériques étant donné qu'il n'existe aucun environnement d'expérimentation (*testbed*) de la virtualisation des réseaux, comme GENI ou PlanetLab, supportant cette fine granularité lors du partage des ressources physiques. L'implémentation sur une plateforme réelle sera nécessaire pour démontrer son intérêt pratique. Nous pensons que ceci est réalisable en utilisant SDN de façon similaire à ce qu'on a réalisé pour le réseau domestique.

Dans le chapitre 5 nous présentons un système de contrôleurs à deux niveaux qui s'adapte à la charge dynamique de chaque flux réseau. Le système utilise une approche prédictive afin de trouver la meilleure requête de partage pour chaque VN. Cette requête repose sur l'estimation de la relation entre la performance d'un VN en termes de délais de paquets et des allocations de bande passante actuelles et passées. Ensuite, en raison de la contrainte de capacité de la liaison physique, le système ajuste la portion de la bande passante offerte pour chacun d'eux.

Le chapitre 6 conclut ce mémoire par une synthèse de nos contributions. Nous clôturons ce travail par présenter plusieurs perspectives et extensions possibles de ce travail.

Chapitre 2

Notions préliminaires et état de l'art

Sommaire

1 Introduction **10**
2 Qualité de service dans les réseaux **10**
2.1 Mécanismes de gestion de la QoS 10
2.1.1 La classification et la régulation du trafic 10
2.1.2 La politique d'ordonnancement des paquets 10
2.1.3 Le contrôle de congestion du trafic 12
2.1.4 Le contrôle d'admission des flux 14
2.2 Les métriques de QoS 14
2.3 Les modèles de QoS dans les réseaux IP 15
2.3.1 Le modèle "Integrated Services" 15
2.3.2 Le modèle "Differentiated Services" 15
2.3.3 Utilisation des modèles de QoS dans l'Internet actuel 16
3 La virtualisation des réseaux **16**
3.1 Les réseaux Overlays 16
3.2 Les techniques de la virtualisation des réseaux 17
3.2.1 Techniques basées sur la virtualisation des protocoles 17
3.2.2 Techniques basées sur la virtualisation des machines 20
3.3 Les projets de recherche sur la virtualisation des réseaux 22
4 Les réseaux programmables **23**
4.1 Les réseaux définis par logiciels (SDN) 23
4.1.1 OpenFlow 25
4.1.2 FlowVisor 26
4.2 Les routeurs programmables 27
5 L'approvisionnement des réseaux virtuels **28**
5.1 Formulation du problème 31
5.2 La découverte de l'infrastructure physique 32
5.2.1 La description des ressources 32
5.2.2 Le regroupement des ressources 33
5.3 Le mappage du réseau virtuel 34
5.4 Allocation des ressources physiques pour le réseau virtuel 36
6 Conclusion **39**

1 Introduction

La virtualisation des réseaux fournit une nouvelle approche qui permet d'exécuter plusieurs réseaux virtuels sur une infrastructure de réseau physique partagée. Elle offre un moyen technique répondant aux besoins en QoS des nouveaux services dans les réseaux à commutation de paquets. Dans ce chapitre, nous rappelons brièvement les notions de la qualité de service des réseaux, les techniques de virtualisation et les réseaux programmables. Nous finirons cette partie par présenter les travaux existants remédiant au problème d'approvisionnement des réseaux virtuels.

2 Qualité de service dans les réseaux

Durant ces dernières décennies, la gestion de la qualité de service dans les réseaux informatiques a constitué un grand défi pour les opérateurs de réseau suite aux exigences croissantes des applications très variées. La QoS est définie comme étant la capacité d'un réseau à transporter le flux de paquets de manière satisfaisante selon les exigences des usagers. En effet, à toute application est associée une exigence liée la manipulation du trafic dans le réseau. Un ensemble de mécanismes de qualité de service a été spécifié pour répondre à cette demande [Aurrecoechea et al., 1998]. Ces mécanismes reposent sur l'idée de la mesure, de l'amélioration et de la limite de garantie préalable des performances du réseau. Cette propriété est particulièrement indispensable à la transmission du contenu multimédia.

2.1 Mécanismes de gestion de la QoS

Pour répondre à la demande des flux en qualité de service, plusieurs mécanismes de gestion ont été proposés dans la littérature [Armitage, 2000, Ferguson and Huston, 1998]. Dans ce qui suit, nous allons présenter brièvement les principaux mécanismes.

2.1.1 La classification et la régulation du trafic

La classification du trafic est le processus automatisé qui ordonne le transport de flux de paquets dans le réseau en fonction de divers paramètres dans un certain nombre de groupes. Ce processus est utilisé pour mettre en place un service différencié qui associe des ressources à certains types de mécanisme de gestion. À titre d'exemple, des règles de classification du trafic sont définies en utilisant les bits de priorité ToS (*Type of Service*), redéfinie en DSCP par DiffServ [Pujolle, 2014], dans l'entête du paquet IP pour distinguer les flux [Nichols et al., 1998a]. La régulation des paquets (*traffic shaping*) vise à organiser le temps de passage et la circulation des paquets dans un routeur. Ce mécanisme de lissage des flux est généralement utilisé pour contrôler le débit et éviter la congestion d'un réseau dans le but d'assurer une meilleure performance et une meilleure qualité de service.

2.1.2 La politique d'ordonnancement des paquets

La politique d'ordonnancement des paquets consiste à décider de l'ordre du traitement des paquets dans un routeur. Pour garantir une bonne QoS, il est indispensable à ce niveau, de dé-

finir une méthodologie d'attribution des différentes priorités accordées aux flux. Le mécanisme d'ordonnancement le plus courant, qui assure le partage le plus équitable des ressources, est le *Weighted Fair Queuing (WFQ)* [Demers et al., 1989]. Il repose sur le principe de séparation du trafic en plusieurs flots au niveau d'un routeur et d'allocation de fraction de la bande passante à chaque flot [Stiliadis and Varma, 1998]. L'ordonnancement *Weighted Fair Queuing* est une amélioration de l'algorithme *Fair Queuing*. Il utilise un système de poids pour chaque file d'attente permettant de délimiter le nombre de paquets nécessaire à ordonnancer à chaque cycle. Les paquets, qui sont placés en tête de file, sont servis selon le poids qui leur est affecté basé sur le principe de l'algorithme *Generalized Processor Sharing (GPS)* [Parekh and Gallager, 1993]. Ce dernier est un modèle théorique qui ne peut pas être implémenté en pratique. L'ordonnancement WFQ rajoute un processus de calcul temporel pour déterminer le "*finish-time*" qui correspond à l'instant de transmission du dernier bit du paquet par l'algorithme GPS. Ce paramètre prend en considération la date d'arrivée du paquet et définit l'ordre dans lequel les paquets vont être servis. En effet, l'ordonnanceur distingue le paquet ayant la plus petite valeur de finish time pour être traité. Le finish-time $F_i(n,t)$ d'un paquet n du flux i arrivant à l'instant t est déduit de l'équation suivante [Aidarous and Plevyak, 2003] :

$$F_i(n,t) = S_i(n,t) + \frac{P_i(n,t)}{w_i}, \tag{2.1}$$

$$S_i(n,t) = max\{F_i(n-1,t), R(t)\}, \tag{2.2}$$

où $P_i(n,t)$ est la taille en bits du paquet n du flux i arrivant à l'instant t, w_i le poids du flux i et $R(t)$ la valeur du paramètre incrémenté en fonction du débit de sortie et du nombre des flux actifs associée au numéro du cycle à l'instant t.

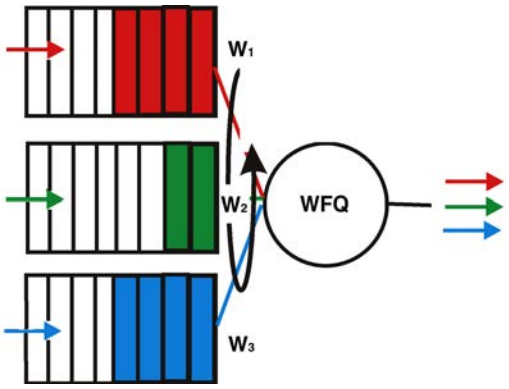


FIGURE 2.1 – Schéma du mécanisme d'ordonnancement WFQ

L'algorithme d'ordonnancement Weighted Fair Queuing est largement utilisé dans les réseaux pour partager la bande passante disponible pour différents flux quelle indépendamment de la taille de leurs paquets. Il garantit différents débits entre chaque flux et en assurant une équité en termes de satisfaction des contraintes de QoS. La figure 2.1 schématise le fonctionnement du mécanisme d'ordonnancement adopté par WFQ.

2.1.3 Le contrôle de congestion du trafic

Le contrôle de congestion est une opération qui vise à éviter la saturation du réseau en contrôlant la charge introduite du trafic [Nagle, 1984, Floyd, 2000]. La cause principale de la congestion est la nature sporadique des trafics. Il y a deux approches possibles pour le contrôle de congestion du trafic ; l'approche boucle ouverte et l'approche boucle fermée. L'approche boucle ouverte cherche à se prévenir de la congestion. Quant à l'approche boucle fermée, elle cherche plutôt à surveiller le réseau, à détecter la congestion et à prendre les mesures adéquates pour pallier ces défauts.

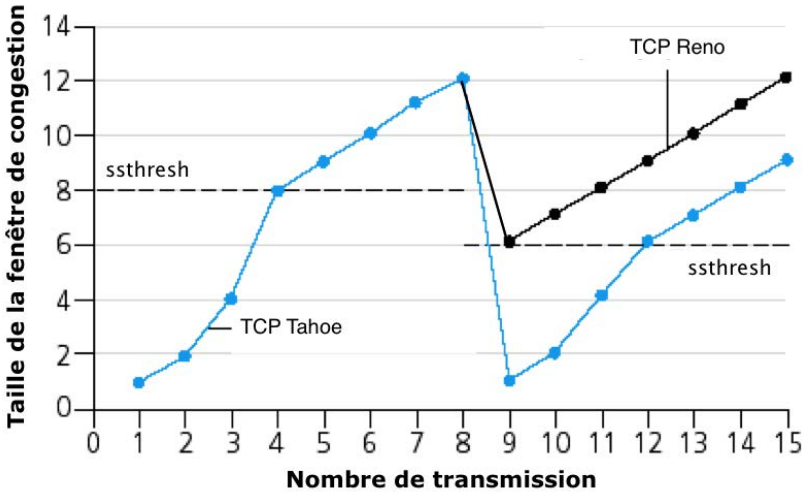


FIGURE 2.2 – Evolution de la fenêtre de congestion avec TCP Tahoe et TCP Reno [Kurose and Ross, 2009]

Le contrôle de congestion de TCP adapte le débit d'émission à la charge du réseau en utilisant une méthode d'augmentation additive/retrait multiplicatif (AIMD) [Allman et al., 2009]. L'émetteur définit une fenêtre de congestion (cwnd) qui représente la quantité maximale de données que l'émetteur peut émettre avant d'attendre un acquittement. Le récepteur définit la fenêtre de réception (rwnd) qui décrit la quantité maximale de données que le récepteur peut accepter. La taille maximale d'un segment est désignée par le MSS. Le paramètre *Slow Start threshold(ssthresh)* définit le seuil qui est le débit au-dessus duquel la congestion risque de se produire [Kurose and Ross, 2009]. La fenêtre d'émission est alors égale à la valeur minimale entre le cwnd et rwnd.
À titre d'exemple, dans le TCP Tahoe [Jacobson, 1988], la fenêtre de congestion croit en exponentiel durant la phase de *Slow Start* quand l'émetteur reçoit des acquittements pour les paquets envoyés.
Lorsqu'il y a une perte de paquets la valeur de ssthresh devient celle de cwnd sur deux on remet la valeur de cwnd à 1.Lorsque ssthresh est atteint, on passe à la phase de *Congestion Avoidance* et la valeur de cwnd augmente de façon linéaire d'un MSS. Dans TCP Reno [Jacobson, 1990], lorsque l'émetteur reçoit trois acquittements dupliqués, il passe à la phase de *Fast Recovery* où la nouvelle valeur de cwnd est égale à ssthresh plus trois MSS. Pour chaque acquittement reçu,

il incrémente le cwnd d'un MSS jusqu'au *timeout*. S'il y a perte de paquet la valeur de cwnd sera égale à 1. La figure 2.2 illustre l'évolution de la fenêtre de congestion (cwnd) avec les deux différents algorithmes TCP.

Un autre mécanisme de contrôle de congestion qui nous semble intéressant et sera utilisé dans ce travail. Il s'agit du congestion avec l'algorithme des courbes *loss-load*. Ce mécanisme a été proposé par Williamson et Cheriton [Williamson and Cheriton, 1991] comme mécanisme de rétroaction pour contrôler le débit d'émission dans les réseaux ATM. La courbe de perte de charge exprime la probabilité de perte de paquets en fonction de la bande passante offerte à un moment donné et d'un état de charge dans le réseau. Cette courbe fournit une information sur le niveau de service prévisible sans exiger une réservation de ressources pour chaque flux. Le contrôle de congestion se base sur une gestion dynamique de la bande passante [Williamson, 1996] pour chaque flux lorsque l'émetteur du trafic choisit un débit moyen de transmission en fonction des besoins en QoS. Périodiquement, le mécanisme met à jour des courbes de perte de charge en fonction de la charge actuelle du réseau afin de répartir les capacités offertes pour les différents trafics. L'algorithme cherche à trouver un compromis entre le débit et la perte de paquets pour chaque flux. Si la somme des demandes de bande passante des émetteurs est inférieure ou égale à la capacité d'un lien, chaque émetteur recevra la bande passante demandée. Sinon, l'allocation se fait de telle sorte que toute la capacité d'un lien est utilisée et tout excès de trafic est rejeté. L'excès de trafic est éliminé par l'affectation d'une probabilité de perte de paquets à chaque émetteur à partir de laquelle la portion du lien offerte sera calculée. Le mécanisme pénalise les émetteurs avides en leur attribuant moins de bande passante que celle reçue lorsqu'ils sont moins gourmands. Nous présentons en détail dans le chapitre 5 l'algorithme de l'allocation dynamique de la bande passante sur lequel se base ce mécanisme.

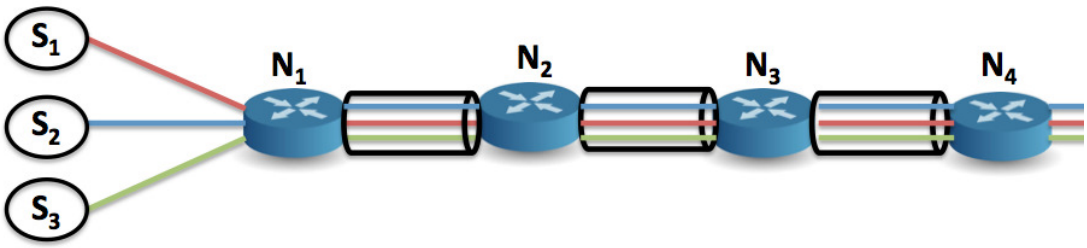


FIGURE 2.3 – Scénario de partage de la capacité de plusieurs liens physiques entre trois sources utilisant l'algorithme des courbes loss-load

La figure 2.3 illustre un scenario où trois sources se partagent la bande passante de plusieurs liens physiques d'une infrastructure réseau. Avec le contrôle de congestion des courbes de perte de charge, chaque nœud physique notifie les sources de la bande passante offerte sur son interface de sortie. Il est important de noter qu'il n'y a aucune coopération entre les sources faisant appel à une bande passante exprimée en bits par seconde (ou paquets par seconde).
Pour chaque source, l'algorithme des courbes loss-load exige que chaque nœud calcule la proba-

bilité de perte des paquets tout en gardant une trace du débit d'émission de chaque source traversant ses liens. Il exige aussi de chaque nœud de fournir la courbe loss-load chaque fois qu'une source change son débit d'émission. Chaque source cherche alors le lien du goulot d'étranglement et adapte ensuite son débit moyen à partir de la courbe loss-load qui a été fournit par le nœud physique. Ce goulot d'étranglement est le lien où la portion offerte à une source est le minimum de toutes les autre portions offertes. Les sources ont la responsabilité de choisir leur propre débit de transmission moyen.

2.1.4 Le contrôle d'admission des flux

L'objectif du mécanisme du contrôle d'admission est de décider, pour un trafic donné, s'il sera ou non admis dans le réseau. Il vise également à assurer la qualité de service requise pour un trafic sans affecter d'autres trafics actifs [Aurrecoechea et al., 1998]. Ceci est généralement effectué par la réservation des ressources disponibles lors de l'exécution du trafic.
À titre d'exemple, le contrôle d'admission dans ATM *Call Admission Control (CAC)* permet de limiter le nombre de connexions afin de garantir la QoS et le maintien du partage des ressources à des connexions déjà en cours. Il calcule le débit nécessaire à une demande à partir d'un descripteur de trafic et ensuite décide d'accepter ou non la demande. En cas d'acceptation, un chemin possédant les ressources disponibles pour supporter le débit est attribué entrainant une vérification de la conformité durant la connexion. Cette vérification se fait à l'aide d'un mécanisme de seau percé pour la régulation des rafales. En cas de refus, un message est renvoyé à la source et un autre chemin peut être cherché.

2.2 Les métriques de QoS

La gestion efficace de la QoS, comme enjeu majeur pour les fournisseurs de services, s'appuie sur un ensemble de paramètres qualitatifs et quantitatifs. Les paramètres qualitatifs donnent une description de la qualité perçue par l'utilisateur final, telle que la qualité de la voix ou de la vidéo. Les paramètres quantitatifs expriment le niveau de qualité relatif au réseau, tel que le temps d'acheminement des paquets de bout en bout.
Les métriques les plus importantes de la QoS dans les réseaux IP sont comme suit :

— **le délai de transit** [Almes et al., 1999a, Almes et al., 1999c] : il représente l'intervalle de temps nécessaire à la transmission d'un paquet de données depuis la source jusqu'à la destination.

— **le débit** [Chimento and Ishac, 2008] : ce paramètre désigne la quantité d'informations que l'application véhicule pendant un intervalle de temps donné.

— **le taux de perte de paquets** [Almes et al., 1999b] : ce paramètre indique la probabilité pour laquelle les données n'arrivent pas à destination.

— **la gigue** [Demichelis and Chimento, 2002] : ce paramètre désigne la variation de latence des paquets. Le réseau doit respecter ce paramètre lors de la transmission de la voix et la vidéo-conférence,.

2.3 Les modèles de QoS dans les réseaux IP

2.3.1 Le modèle "Integrated Services"

Le modèle *Integrated Services* (IntServ) [Braden et al., 1994] est une architecture qui spécifie les éléments garantissant la QoS de bout en bout dans les réseaux à commutation de paquets. Il est proposé pour assurer la QoS à l'architecture Internet sans perturber le bon fonctionnement du protocole IP.
Les composantes clés de l'architecture offrent deux nouvelles classes de service supplémentaire par rapport au service traditionnel du Best-effort. Cependant, ce dernier ne garantit aucun critère de QoS. IntServ définit deux types de service. Le premier est la charge contrôlée qui n'offre aucune garantie sur le délai de bout en bout et la perte mais il garantit l'absence de congestion des trafics du réseau. Le second type est le service garanti la bande passante et le délai d'acheminement limité. Ce modèle repose sur le protocole de signalisation RSVP d'acheminement des informations routeurs répondant aux demandes et aux exigences des flux [Zhang et al., 1997]. Les routeurs assurent les fonctionnalités de contrôle d'admission de flux et de réservation de ressources. RSVP est le protocole qu'utilisent les applications pour réserver des ressources du réseau. À grande échelle, le modèle IntServ admet beaucoup de limitations. En effet, en utilisant le protocole RSVP, IntServ doit maintenir les états de réservation de chaque flux traversant les routeurs. Lorsque le nombre de flux augmente, la gestion des états devient une tache difficile que IntServ ne peut pas supporter. Il en découle une détérioration des performances dans le réseau. IntServ nécessite également un grand échange de messages de rafraichissement entrainant ainsi un accroissement de la charge du réseau et par conséquent une augmentation de la probabilité de suppression des paquets. RSVP souffre de problèmes de mise à l'échelle, il convient seulement aux réseaux de petite taille et il n'est pas très performant pour les grands réseaux.

2.3.2 Le modèle "Differentiated Services"

Le modèle *Differentiated Services* (DiffServ) [Nichols et al., 1998b] a été développé pour répondre à la limitation du modèle IntServ. L'objectif est de fournir, d'un côté, une QoS par classe de paquets IP et non pas par flux. D'un autre côté, il pousse la surcharge du traitement aux extrémités du réseau cœur. Cette classification se base sur le champ DiffServ (DSCP) dans l'entête d'un paquet IP visant à diviser le réseau en sous-domaines. Cette architecture définit un domaine comme un ensemble de nœuds soumis aux mêmes politiques d'allocation de service. Chaque domaine est constitué de deux types d'éléments ; les éléments de bordure et les éléments du cœur du réseau. Le modèle DiffServ cherche, d'une part, à solliciter des routeurs situés aux frontières du réseau de gérer toutes les fonctions de classification de paquets et de mise en forme de trafic. D'autre part, il exige des routeurs du cœur du réseau d'appliquer des comportements prédéfinis par classe de flux. DiffServ définit quatre classes de service ou *Per-Hop Behavior (PHB)* qui caractérisent les comportements des routeurs [Jacobson et al., 1999, Heinanen et al., 1999] :

— *Default PHB* : ce comportement désigne le comportement par défaut (*Best Effort*). Il est utilisé pour les trafics qui ne sont pas temps réel.

— *Class Selector PHB* : ce comportement a été proposé pour maintenir une compatibilité avec les anciennes utilisations du champ "Priorité" dans les paquets IP.

— *Expedited Forwarding PHB* (EF) : ce comportement permet de garantir une bande passante avec de faibles taux de perte de paquets, de délai et de gigue. Il nécessite un contrôle d'accès. Ce comportement est destiné aux trafics nécessitant une haute disponibilité comme la visioconférence.

— *Assured Forwarding PHB* (AF) : ce comportement offre quatre classes pour chaque flux. Chaque classe jouit d'une quantité différente de ressources dans les routeurs. Chaque classe comprenant 3 niveaux de priorité pour faire du rejet sélectif. Ce comportement permet d'assigner une classe qui définit différents niveaux de service aux clients et aux applications. À l'intérieur de chaque classe, les paquets d'un flux jouissent d'une priorité qui décrit le comportement des routeurs lors de la détection d'une congestion.

2.3.3 Utilisation des modèles de QoS dans l'Internet actuel

En raison des problèmes de résistance au facteur d'échelle, le modèle IntServ est rarement mis en œuvre pour garantir la QoS de bout en bout. Lorsque ce modèle est implémenté, ceci se produit généralement au niveau du réseau d'accès. DiffServ a été largement mis en œuvre dans le cœur du réseau d'Internet. Ce modèle permet d'assurer la QoS en gérant la priorité pour divers trafics qui transitent entre plusieurs ASs. En proposant différentes classes de service (Platinum, Gold, etc.) pour divers tarifs, les fournisseurs de services ont généré beaucoup de revenus. Cette politique s'avère problématique lorsque le trafic transite par des chemins disposant de ressources insuffisantes. Dans ce cas précis, les exigences des caractéristiques de performance telle que la gigue ou la latence ne seront pas respectées. Ce problème est résolu grâce à l'architecture MPLS (*Multi-Protocol Label Switching*) qui permet de faire de l'ingénierie de trafic. Cette architecture permet de transporter davantage de trafic IP à grande vitesse, en réduisant les délais de traitement au niveau des routeurs. De nouveaux mécanismes comme *MPLS DiffServ-aware Traffic Engineering* ont vu le jour pour permettre le déploiement efficace de DiffServ. Ces mécanismes permettent de garantir la QoS dans un seul AS et par conséquent lorsque les flux traversent plusieurs ASs lors de l'acheminement vers l'utilisateur final, le problème persiste.

3 La virtualisation des réseaux

3.1 Les réseaux Overlays

La conception des réseaux overlay est fondée sur la création d'une topologie virtuelle sur une topologie physique existante. Son objectif est de fournir des services personnalisés à l'utilisateur [Han et al., 2005]. Il s'agit d'un réseau informatique construit au-dessus d'un second réseau dont le but est d'optimiser la distribution des données. Les optimisations réalisées concernent le délai, la bande passante et l'accroissement de la disponibilité des services sur ce réseau. Internet a commencé comme un réseau overlay superposé sur un réseau de télécommunications.
Cette implémentation se fait, généralement, au niveau de la couche application du modèle OSI

bien qu'actuellement elle s'effectue au niveau des couches inférieures.
Les réseaux overlays ne nécessitent pas une modification dans l'infrastructure physique d'un réseau. Ils sont par conséquent très utilisés comme moyens simples et peu coûteux pour déployer de nouveaux services réseaux. Il est important de noter que ces réseaux nécessitent beaucoup d'entretien pour permettre le déploiement et la gestion des réseaux sous-jacents. La plupart des réseaux overlays ont été implémentés au-dessus de la couche IP, cette solution ne peut pas aller au-delà des limites d'Internet. Bien que les notions des réseaux Overlays et de la virtualisation des réseaux soient liées, ils ne s'agissent pas des mêmes réseaux. Les réseaux overlays sont proposés comme solution aux problèmes liés à Internet tels que la gestion de la QoS et le multicast mais la virtualisation des réseaux se présente comme une nouvelle architecture d'Internet palliant tous les problèmes existants.

3.2 Les techniques de la virtualisation des réseaux

La virtualisation est une technique utilisée pour modifier les propriétés d'un service de réseau sans introduire de modifications au niveau des clients et des serveurs. Elle permet la coexistence de multiples réseaux hétérogènes dans une seule infrastructure. Pour cela, cette technologie doit assurer un niveau adéquat d'isolation afin de permettre l'utilisation des ressources physiques du réseau en temps réel et à grande échelle.

Au cours de ces dernières années, plusieurs techniques ont été utilisées pour créer des réseaux virtuels comme les VLANs (réseaux locaux virtuels) et les VPNs (réseaux privés virtuels).
Récemment, les approches de virtualisation des serveurs ont été utilisées pour créer des routeurs et des liens virtuels sur des équipements physiques et des canaux de communication. Dans ce qui suit nous présentons une brève discussion des approches de virtualisation des réseaux.

3.2.1 Techniques basées sur la virtualisation des protocoles

Les approches basées sur le protocole mettent en œuvre un protocole permettant l'identification et l'isolation des réseaux virtuels [Chowdhury and Boutaba, 2009]. Ce type d'approche exige que l'équipement physique soit capable de supporter le protocole choisi. Un exemple de la virtualisation de réseau à base de protocole est les réseaux locaux virtuels (VLAN). L'idée est de diviser logiquement un réseau local en plusieurs réseaux virtuels. Les hôtes dans un même VLAN communiquent entre eux si elles sont sur le même réseau local, quel que soit l'emplacement physique.
La création de réseaux privés virtuels est une autre approche couramment utilisée. Les VPN sont généralement utilisés pour fournir un canal de communication sécurisé entre plusieurs sites géographiquement distants. Des protocoles pour assurer la confidentialité des données et l'authentification de l'utilisateur sont utilisés dans ce genre de réseau virtuel.
Dans ce qui suit nous allons présenter en détail les techniques utilisées pour la virtualisation basées sur les protocoles.

3.2.1.1 Les réseaux locaux virtuels (VLANs)

Un VLAN permet le regroupement de plusieurs hôtes, de façon logique et non physique

indépendamment de leur connectivité physique [Pillou and Lemainque, 2012]. Il permet de créer des domaines de diffusion gérés logiquement sans se soucier de l'emplacement de ses hôtes. Plusieurs VLANs peuvent coexister sur un même commutateur réseau et ils peuvent être locaux à un commutateur ou s'étendre sur un ensemble de commutateurs reliés entre eux. L'objectif est de contourner les limitations de l'architecture physique. Ceci conduit à l'amélioration de la gestion du réseau et de l'optimisation de la bande passante tout en séparant les flux de chaque ensemble d'hôtes. Chaque trame possède un identifiant du VLAN dans l'en-tête de contrôle d'accès au support (*Media Access Control*, (MAC)). Les réseaux locaux virtuels fonctionnent au niveau des couches liaison de données et réseau du modèle OSI. Ils sont définis par le standard IEEE 802.1Q.
La figure 2.4 illustre le regroupement de plusieurs PCs en VLANs indépendamment de leur connectivité physique.

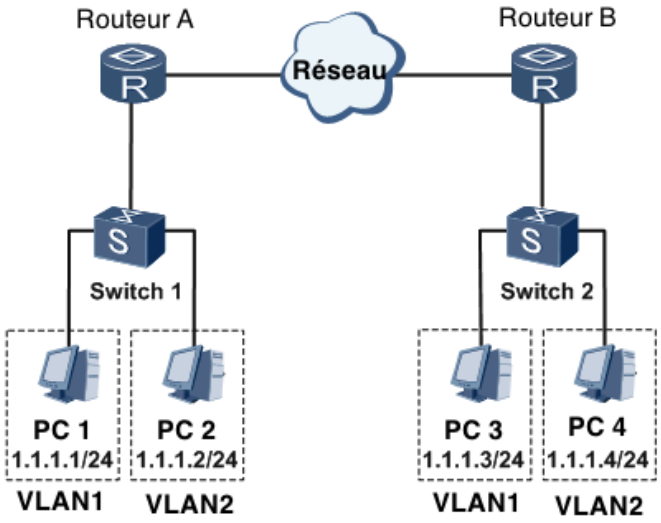


FIGURE 2.4 – Regroupement des PCs en VLANs

On distingue trois types de VLAN [Rajaravivarma, 1997] :

— **VLAN de niveau 1 ou VLAN par port** (*Port-Based VLAN*)
Il définit un réseau virtuel en fonction des ports de raccordement au commutateur. On associe un port physique de ce commutateur à un numéro de VLAN.

— **VLAN de niveau 2 ou VLAN MAC** (*MAC Address-Based VLAN*)
Il définit un réseau virtuel en fonction des adresses MAC des hôtes. Le déploiement de ce type de VLAN est plus souple que le VLAN de niveau 1 puisque la machine, peu importe le port sur lequel elle sera connectée, elle fera toujours partie du VLAN dans lequel son adresse MAC est configurée.

— **VLAN de niveau 3**
On distingue deux types de VLAN de niveau 3.

  — **VLAN par sous-réseau** (*Network Address-Based VLAN*)
  Il a le même principe que pour les VLAN de niveau 2 mais en indiquant les adresses IP

source des datagrammes et en associant des sous-réseaux à différents VLANs. Ceci permet de modifier automatiquement la configuration des commutateurs lors d'un changement ou d'un déplacement de la hôte.

— **VLAN par protocole** (*Protocol-Based VLAN*)
Il définit un réseau virtuel en fonction des protocoles. Par exemple, il définit un VLAN pour TCP/IP. Dans ce cas, tous les hôtes, qui utilisent ce protocole dans un même VLAN, sont regroupées.

3.2.1.2 Les réseaux privés virtuels (VPN)

Un réseau privé virtuel (*Virtual Private Network* (VPN)), est une technologie utilisée pour connecter en toute sécurité deux réseaux physiques géographiquement distants. Elle permet de créer une liaison virtuelle, sur un réseau public comme Internet, entre des ordinateurs distants. Cette technologie repose sur un protocole de tunneling qui permet de faire circuler les informations de façon cryptée d'un bout à l'autre du tunnel. Son objectif est de masquer la distance entre deux ou plusieurs sites. Les VPNs ne sont pas conçus pour créer plusieurs réseaux virtuels sur une même infrastructure physique.
La figure 2.5 montre un schéma du déploiement d'un VPN entre plusieurs sites distants d'une organisation.

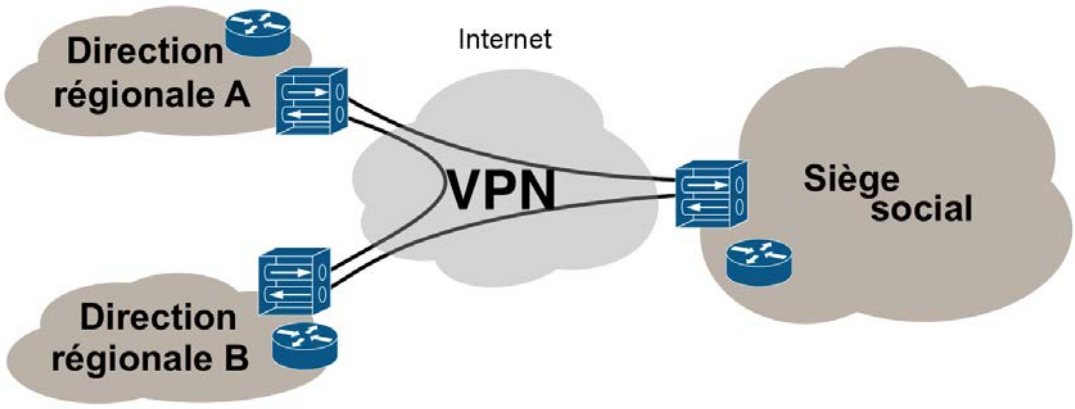


FIGURE 2.5 – VPN entre plusieurs sites distants d'une entreprise donnée

On distingue trois types de VPN :

— **VPN de couche 1 (L1VPN)**
Il permet d'offrir un service par la couche physique du modèle OSI. Il fournit la connectivité entre deux ou plusieurs sites clients [Fedyk et al., 2008]. Il offre aux clients un certain contrôle sur la création et le type de la connectivité du réseau virtuel.

— **VPN de couche 2 (L2VPN)**
Il permet le transport des trames de la liaison de données des différents sites participants. Il s'agit d'une solution simple pour des connexions point à point via des tunnels pour permettre le transport de paquets de n'importe quel protocole de la couche réseau [Andersson and Rosen, 2006].

— **VPN de couche 3 (L3VPN)**
Il permet de connecter plusieurs sites géographiquement séparés à travers le réseau commuté par paquets. Il permet de garantir à chaque trafic utilisateur d'être séparé des autres trafics au niveau du réseau cœur. Ceci est réalisé en déployant des équipements client de bord (*Customer Edge-CE*) qui sont à leur tour connectés aux équipements fournisseur de bord (*Provider Edge-PE*) dans le réseau cœur [El Mghazli et al., 2005]. L'objectif est de faire transiter le trafic entre les PE en utilisant ses routeurs internes. Le L3VPN garantit, pour chaque utilisateur, un trafic privé et séparé des autres utilisateurs connectés au réseau cœur.

3.2.2 Techniques basées sur la virtualisation des machines

Les approches basées sur la virtualisation des machines consistent à isoler les ressources informatiques de façon à pouvoir exécuter plusieurs instances de réseaux virtuels par un moyen de groupes de machines virtuelles interconnectées (VMs) [Sahoo et al., 2010]. Ces machines sont utilisées pour créer des routeurs virtuels et des liens virtuels pour servir de liaison. Cette technique est relativement souple, car elle permet l'utilisation d'un logiciel personnalisé, pas cher, pour la création d'un commutateur ou d'un routeur virtuel.
Dans ce qui suit nous allons présenter en détail les techniques basées sur la virtualisation de machines qui sont le cloisonnement, la virtualisation complète et la para-virtualisation.

3.2.2.1 Le cloisonnement

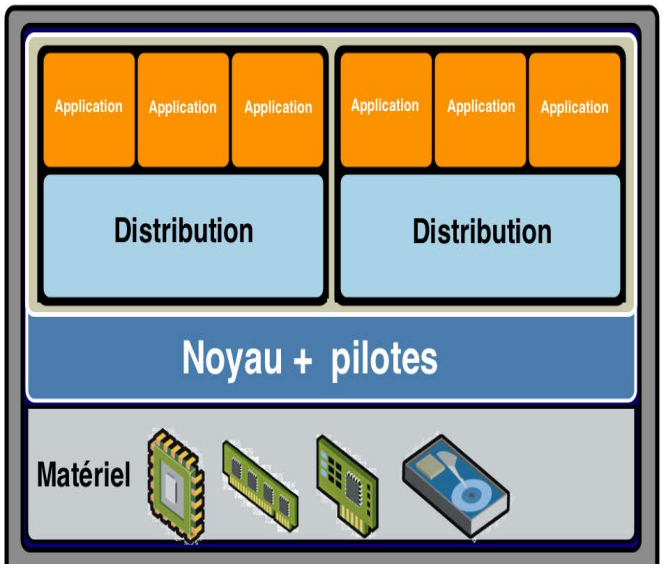


FIGURE 2.6 – La technique du cloisonnement

Le cloisonnement est la technique de virtualisation la plus légère qu'on peut mettre en œuvre au sein d'un même système d'exploitation. Pour cela on le divise en plusieurs espaces ou environnements. Chaque environnement est géré par le système d'exploitation hôte comme

un processus isolé dans un conteneur partageant le même noyau. Le conteneur apporte une virtualisation de l'environnement d'exécution. Il permet aux programmes de chaque contexte de communiquer seulement avec les processus et les ressources qui leur sont associés. L'espace noyau n'est pas différencié, il est unique et partagé entre les différents contextes. Il fournit la virtualisation, l'isolement et la gestion des ressources. Ce partage du noyau limite cette technique aux environnements de mêmes types.

3.2.2.2 La virtualisation complète

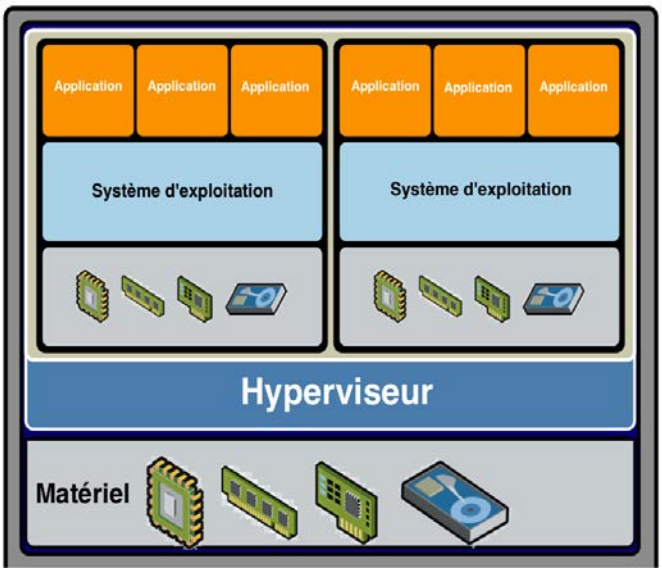


FIGURE 2.7 – Schématisation de la virtualisation complète

La virtualisation complète (*full virtualization*) est une technique de virtualisation qui offre une réplique virtuelle du matériel du système de sorte que les systèmes d'exploitation et logiciels peuvent fonctionner sur le matériel virtuel exactement comme sur le matériel d'origine. L'hyperviseur constitue la couche d'abstraction entre les systèmes d'exploitation invités et le matériel. Son rôle est de créer un environnement virtuel complet simulant globalement un nouvel ordinateur. Au moment de l'exécution, les instructions du système d'exploitation invité ne donnent accès qu'au matériel virtuel présenté par l'hyperviseur. La virtualisation complète offre une meilleure isolation et plus de sécurité pour les machines virtuelles [Chiueh and Brook, 2005] en simplifiant la migration et la portabilité.
La figure 2.7 illustre la création de deux machines virtuelles sur une seule machine physique tout en isolant les ressources. Les programmes qui s'exécutent sur l'espace utilisateur d'un système d'exploitation invité n'ont pas un accès direct au matériel, mais uniquement à la couche d'abstraction. La machine virtuelle émule le matériel pour faciliter l'accès du système d'exploitation aux ressources physiques.

3.2.2.3 La para-virtualisation

La para-virtualisation est très proche du concept de la virtualisation complète. Les deux types de virtualisation reposent sur un hyperviseur qui gère l'interfaçage avec les ressources matérielles [Menon et al., 2006]. Excepté le fait que la para-virtualisation a des fonctionnalités différentes dans chaque technique de virtualisation, elle permet une coopération entre l'hyperviseur et le système d'exploitation invité. En effet, lors de l'exécution du système d'exploitation invité, l'hyperviseur capture les appels système de l'invité et les transmet au matériel. L'hyperviseur gère l'interface qui va permettre à plusieurs systèmes d'exploitation invités d'accéder de manière concurrente aux ressources [Zhang et al., 2010]. Le système d'exploitation invité est conscient de l'exécution sur une machine virtuelle (VM). Cette opération nécessite certaines modifications logicielles non seulement au niveau du système d'exploitation hôte mais également au niveau du système d'exploitation invité. Ce dernier doit être muni des pilotes permettant d'adresser des commandes au matériel.
La figure 2.8 schématise la technique de para-virtualisation.

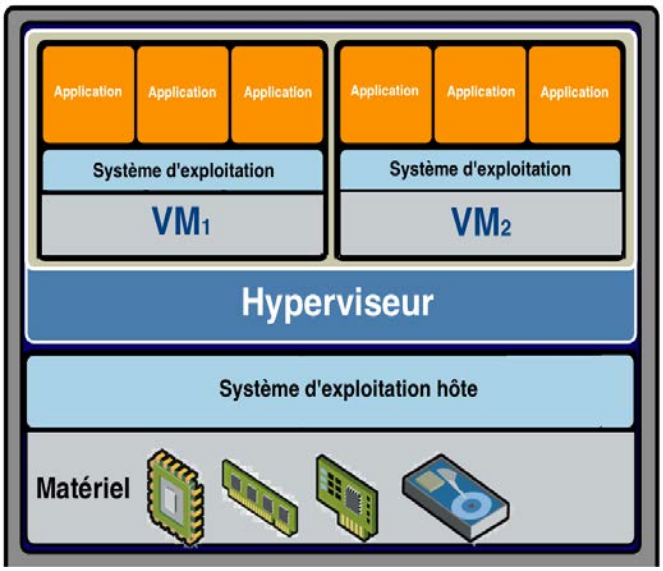


FIGURE 2.8 – La technique de la para-virtualisation

3.3 Les projets de recherche sur la virtualisation des réseaux

Récemment, la virtualisation des réseaux a reçu une attention particulière. En effet cette technologie est considérée comme le concept clé de l'Internet du futur. Différents projets ont vu le jour suite à de nombreuses initiatives internationales telles que PlanetLab [Peterson et al., 2003] et GENI [Berman et al., 2014] aux États Unis, Akari [Harai, 2009] en Asie et 4ward [Bauck and Görg, 2008] en Europe.
Ces initiatives cherchent à fournir une infrastructure réseau virtualisée pour aider les chercheurs à déployer et évaluer de nouveaux services sur des réseaux virtuels. Ces projets sont classés selon la technologie réseau utilisée ou selon le niveau de virtualisation [Chowdhury and Boutaba,

2009]. Parallèlement à ces projets, plusieurs prototypes d'environnement de virtualisation des réseaux ont été développés dans le but d'exploiter les caractéristiques d'une ou de plusieurs technologies réseau. À titre d'exemple le projet VIOLIN [Ruth et al., 2005], X-Bone [Touch, 2001], AGAVE [Boucadair et al., 2007] se sont focalisés seulement sur les réseaux IP. Quant à l'implémentation dans les projets CABO [Feamster et al., 2007] et GENI elle supporte plusieurs réseaux hétérogènes. Le niveau de la virtualisation se réfère à la granularité correspondante à l'exécution de chaque VN. Par exemple les projets VNET [Sundararaj and Dinda, 2004], VIOLIN et NetScript [Da Silva et al., 2001] offrent seulement une virtualisation du nœud. De surcroît, ils permettent de créer un réseau virtuel en connectant plusieurs nœuds virtuels. Le projet CABO permet en particulier une virtualisation totale de l'infrastructure du réseau dont le but d'offrir plus d'isolation au niveau de chaque VN.
Il est important de noter que le projet CABO est le seul projet de virtualisation des réseaux qui offre une séparation entre les fournisseurs de l'infrastructure physique et les fournisseurs de services. Il permet aux FSs d'exécuter simultanément plusieurs services de bout en bout sur une infrastructure appartenant à différents FIPs. L'allocation de l'infrastructure physique se fait d'une façon statique. En plus, CABO permet aussi aux FSs de contrôler directement les protocoles et les services qui s'exécutent sur les nœuds et les liens virtuels avec une découverte et une gestion de l'infrastructure physique par les FIPs.
Les possibilités offertes par le projet CABO justifient l'intérêt que nous accordons à cette idée dans notre travail de recherche. Par conséquent, nous nous focalisons sur l'allocation dynamique des nœuds et des liens physiques entre plusieurs FSs.

4 Les réseaux programmables

4.1 Les réseaux définis par logiciels (SDN)

Les réseaux définis par logiciels (SDN) et la virtualisation des fonctions réseau *Network Function Virtualization (NFV)* sont de nouvelles façons de concevoir, construire et exploiter les réseaux. Dans ce travail nous n'allons pas nous intéresser pas à la technologie de virtualisation des fonctions réseau qui sont exécutées sur des serveurs standards. Nous considérons que toutes les fonctions du réseau telles que le routage ou le traitement des paquets sont effectuées par des routeurs programmables virtualisés.
Les routeurs sont des équipements du réseau qui permettent de transférer les paquets d'un réseau à un autre. Sur la base de leur fonctionnement, on peut diviser un routeur en deux parties ou plans : le plan de contrôle et le plan de données. Le plan de données est responsable du transport des données de la source à la destination. Il récupère les paquets de données au niveau de l'interface d'entrée afin d'exécuter des fonctions de commutation [Shin et al., 2012]. Les tables de routage sont alors consultées pour déterminer l'interface de sortie des paquets. Tous les paquets qui ont la même destination suivent le même chemin. Le plan de contrôle est en charge de la construction et du maintien des tables de routage. Cette configuration est effectuée soit manuellement par les administrateurs du réseau soit à l'aide des informations distribuées collectées par des protocoles de routage comme BGP ou OSPF. Actuellement, dans un routeur

ou un commutateur classique les tables de routage sont programmées localement. Les nœuds du réseau choisissent librement la meilleure façon de traiter un flux. Le plan de données et le plan de contrôle sont co-localisés sur le même équipement. Les réseaux définis par logiciels (*Software-Defined Network* (SDN)) ont introduit une séparation entre le plan de données et le plan de contrôle pour rendre le réseau programmable [Das et al., 2012]. Le plan de contrôle est placé dans un contrôleur centralisé qui a une vision sur la topologie du réseau. Le plan de données réside encore sur le commutateur ou le routeur.
L'objectif de SDN est d'offrir une flexibilité et une programmabilité des réseaux afin de rendre sa gestion simple. La figure 2.9 illustre les différentes couches des réseaux SDN.

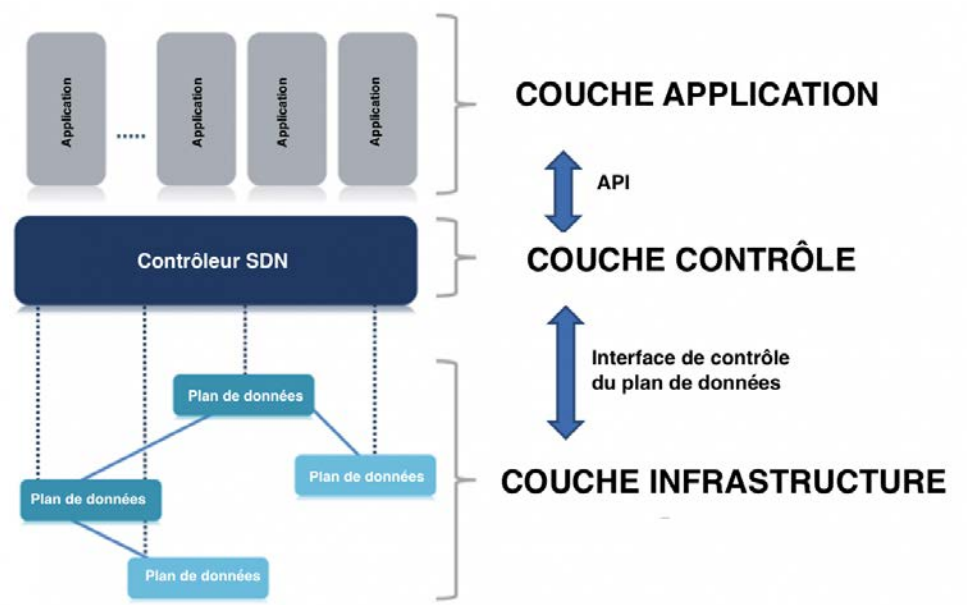


FIGURE 2.9 – Les différentes couches des réseaux définis par logiciels (SDN)

Les réseaux SDN sont le résultat de plusieurs travaux de recherche antérieurs visant à rendre programmable les réseaux afin qu'ils soient plus flexibles et innovants [Nunes et al., 2014]. Deux groupes de recherche ont proposé de séparer le plan de contrôle logiciel du matériel sous-jacent et d'offrir des interfaces pour la gestion. Le but est de pouvoir déployer rapidement des services personnalisés et offrir une configuration dynamique des réseaux au moment de l'exécution. Ces groupes sont celui du travail *OpenSig* et celui de l'initiative *Active Networking* [Feamster et al., 2014].
OpenSig est apparu en 1995 en proposant une application du concept de programmation dans les réseaux ATM. L'idée principale était la séparation du plan de contrôle des réseaux de données, avec un protocole de signalisation entre les deux plans. L'objectif de cette contribution est la programmation et le contrôle à distance des commutateurs ATM.
L'initiative *Active Networking* est apparue au milieu des années 90, dont l'idée est de gérer les ressources des nœuds du réseau par une API (*Application Programming Interface*). Ceci permet aux opérateurs des réseaux de contrôler aisément et activement les nœuds en exécutant un code souhaité.

4.1.1 OpenFlow

OpenFlow est un protocole de communication entre le contrôleur et le commutateur dans un réseau SDN [McKeown et al., 2008]. Il décrit l'interfaçage entre le plan de données et le plan de contrôle dans un équipement réseau. OpenFlow permet l'exécution des opérations du plan de contrôle dans un contrôleur OpenFlow. Il permet également d'ajuster les règles de routage selon les besoins dynamiques du réseau. Il s'agit d'un nouveau standard dans les réseaux SDN qui donne plus de pouvoir aux administrateurs dans la gestion du réseau. Par exemple, dans un *data center* des milliers de serveurs physiques et virtuels sont interconnectés à travers des VLANs, avec SDN et OpenFlow comme protocole, la gestion devient plus simple. Les administrateurs auront seulement à utiliser un logiciel de contrôleur pour programmer VLAN et suivre l'ensemble du réseau OpenFlow.
OpenFlow régit l'échange de l'information entre le contrôleur et le commutateur. Il est composé d'une table de flux qui contient des informations sur le traitement d'un paquet et un canal sécurisé qui relie le commutateur à la commande par une connexion SSL/TLS et OpenFlow. Cette connexion est utilisée pour transmettre les modifications des tables du routage, du contrôleur au commutateur. Elle transporte des paquets pour permettre au contrôleur de les analyser s'il n'y a pas une entrée correspondante pour le flux dans la table de routage. Par la suite, le contrôleur OpenFlow envoie la règle de routage adéquate pour ce flux au commutateur OpenFlow. Chaque entrée dans la table de flux contient un ensemble de champs pour faire la correspondance entre un paquet et une action comme modifier ou envoyer ce paquet. Quand un commutateur OpenFlow reçoit un nouveau paquet qu'il n'a pas vu auparavant et pour lequel il n'a aucune entrée de flux, il l'envoie au contrôleur. Le contrôleur prend, alors, une décision sur la façon de traiter ce paquet. Il peut soit le détruire soit ajouter une entrée qui correspond à la façon de transmettre les paquets similaires à l'avenir. Le commutateur OpenFlow et le contrôleur communiquent via le protocole OpenFlow, qui définit des messages comme l'envoi, la réception, la modification des paquets, la modification de la table de routage et la gestion des statistiques pour chaque flux.
La virtualisation des réseaux n'a pas besoin de SDN pour fournir une abstraction du réseau physique en plusieurs réseaux logiques. De même que, SDN permet la séparation d'un plan de contrôle logiquement centralisé du plan de données sous-jacentes d'un équipement réseau physique ou virtuel. SDN n'est pas indispensable à la virtualisation de réseau, il fournit juste une solution qui facilite la gestion et le déploiement d'un environnement de virtualisation des réseaux. Toutefois, il y a une relation étroite entre la virtualisation des réseaux et SDN ; les deux technologies sont complémentaires afin de faire une abstraction des couches du réseau et de centraliser et simplifier la gestion. SDN fournit un cadre pour le déploiement de la virtualisation, à titre d'exemple, c'est un moyen pour permettre aux clients de plusieurs fournisseurs de Cloud de partager la même infrastructure réseau. La solution est SDN pour la mise en œuvre d'un contrôleur logique centralisé qui installe les règles dans ces commutateurs virtuels et permet de contrôler la façon dont les paquets sont encapsulés. Ce contrôleur doit aussi mettre à jour ces règles lorsque les machines virtuelles se déplacent. La virtualisation des réseaux est plus simple avec SDN, car il s'agit d'un outil très performant utilisé pour la gestion des réseaux virtuels. On peut l'exécuter pour contrôler les différents équipements dans le réseau, tandis que la virtuali-

sation des réseaux ajoute une abstraction logique sur le réseau physique. SDN modifie le réseau et offre un moyen pour l'approvisionner et le gérer.

4.1.2 FlowVisor

FlowVisor [Sherwood et al., 2009] est un contrôleur OpenFlow particulier proposé par Sherwood et al. qui permet de virtualiser un réseau. Il agit comme un *proxy* transparent entre les commutateurs et les contrôleurs OpenFlow. Ce type de contrôleur permet de créer plusieurs réseaux virtuels appelés "*slices*" assurant à chacun d'eux une utilisation des commutateurs et des ressources attribuées. FlowVisor définit un "*slice*" comme un ensemble de flux s'exécutant sur une topologie de commutateurs OpenFlow. Il permet en outre de séparer la table de flux au niveau de chaque commutateur. Concernant la bande passante de chaque lien physique, il assure le partage en affectant un débit minimal pour l'ensemble des flux qui constituent un *slice*. FlowVisor agit comme un hyperviseur réseau qui se situe entre le plan de données et le plan de contrôle d'un commutateur OpenFlow. Il permet ainsi de garantir l'isolation entre plusieurs contrôleurs en charge d'un ou de plusieurs commutateurs OpenFlow.

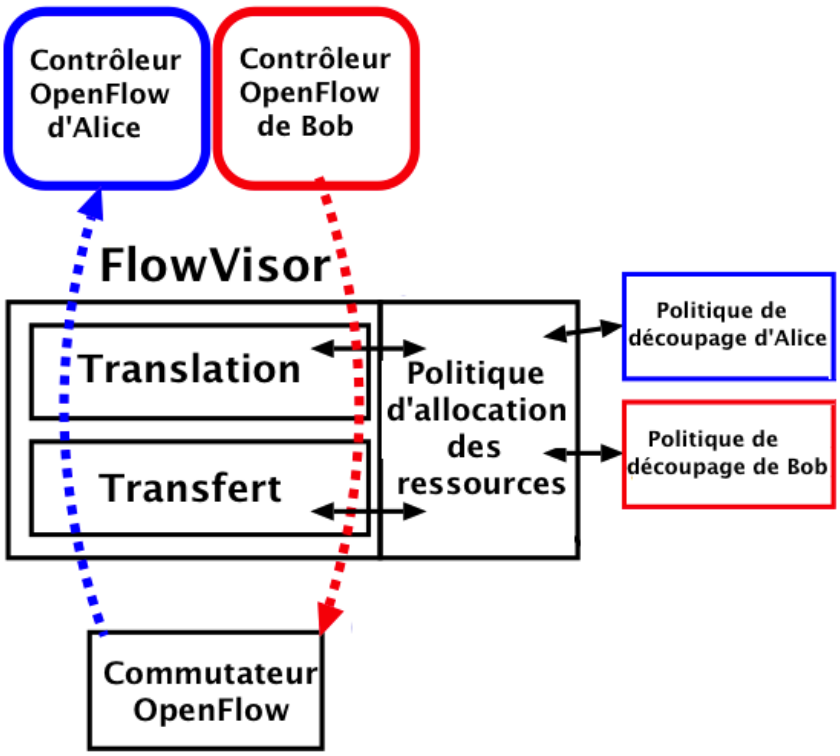


FIGURE 2.10 – Architecture de FlowVisor [Sherwood et al., 2009]

La figure 2.10 illustre un exemple de fonctionnement de FlowVisor. Deux utilisateurs, Bob et Alice, ont créé plusieurs *slices*. Chacun d'eux a déployé un contrôleur OpenFlow dont le but de contrôler un ou plusieurs commutateurs. FlowVisor intercepte tous les messages OpenFlow dans les deux sens de la communication (du commutateur vers le contrôleur et vice-versa). À

l'aide de la politique de découpage de chaque utilisateur (*slicing policy*), il réécrit de manière transparente ces messages afin de contrôler seulement une portion du réseau physique. Les messages des commutateurs ne sont transmis que lorsqu'ils correspondent à leur politique de découpage. Chaque fois qu'un nouveau flux, correspondant à un des *slices*, arrive au niveau d'un commutateur, FlowVisor envoie les messages OpenFlow au contrôleur concerné.

4.2 Les routeurs programmables

Habituellement, les routeurs hautes performances ont été conçus en utilisant des ASICs (*Application-Specific Integrated Circuit*) permettant de traiter les paquets à grande vitesse. Généralement, ces ASICs regroupent un grand nombre de fonctionnalités mais n'offrent aucune flexibilité. Ils ne peuvent présenter que les fonctionnalités préalablement prédéfinies par les concepteurs.
Récemment, une autre approche utilisant les processeurs réseau (*Network Processor*) pour la conception de routeurs hautes performances et programmables a gagnée en popularité [Chao and Liu, 2007]. L'objectif est d'offrir une solution intégrant des fonctions applicatives aux routeurs. Les processeurs réseau sont programmables et offrent des fonctions génériques qui peuvent être utilisées dans différents types de réseaux [Anwer and Feamster, 2009]. L'objectif est d'offrir une flexibilité dans l'aiguillage des paquets arrivant au port d'entrée. Les processeurs réseau sont entièrement programmables et offrent la possibilité de modification des instructions de traitement des paquets afin d'atteindre la vitesse des ASICs.. Les routeurs programmables sont des routeurs qui peuvent exécuter un code personnalisé tout en continuant à utiliser le format standard de paquets. Par conséquent, ils facilitent le développement et le déploiement rapide de nouveaux services dans les réseaux. En outre, Ils créent une excellente plate-forme pour tester de nouvelles architectures de service réseau.
Afin d'expliquer le fonctionnement et le rôle de chaque composante d'un routeur programmable, nous présentons dans la figure 2.11 un schéma d'une architecture classique d'un routeur programmable.

Un routeur programmable contient plusieurs cartes réseaux (*line cards*). Ces dernières représentent le point d'entrée et de sortie des paquets. Lorsqu'un paquet arrive au niveau de la carte d'entrée (*ingress line card*), le processeur réseau traite le paquet et effectue la recherche dans la table de routage afin de trouver la carte de sortie (*egress line card*) [Thiele et al., 2002]. C'est l'opération de recherche de route (*IP lookup*) qui vise à trouver la meilleure route pour un paquet à partir de son adresse de destination. Cette opération nécessite des mémoires telles que la TCAM (*Ternary content-addressable memory*), la SRAM (*static random-access memory*), la DRAM (*Dynamic random-access memory*). Le paquet transite par la suite par le *Switch Fabric*. Les composants essentiels dans une carte ligne sont l'émetteur-récepteur (*Transceiver*), le dispositif de tramage (*Framer*), le CPU, le gestionnaire de trafic (*Traffic Manager*) et plusieurs autres types de mémoire. L'émetteur-récepteur permet la conversion des signaux comme la conversion d'un signal optique vers un signal électrique. Le dispositif de tramage est responsable de la délimitation des trames selon les exigences de la signalisation physique. Il garantit la bonne réception du paquet. Le gestionnaire de trafic est responsable de toutes les fonctionnalités du

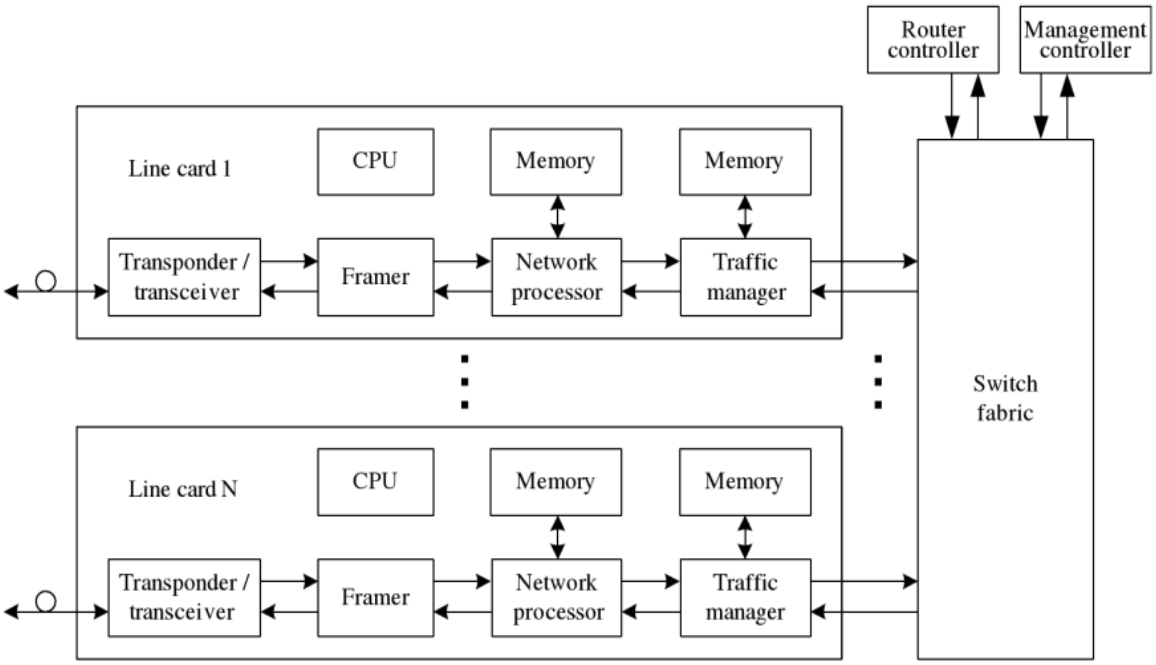


FIGURE 2.11 – Architecture d'un routeur programmable [Chao and Liu, 2007]

contrôle de flux. Il assure aussi la gestion des mémoires tampons. Le CPU est responsable des fonctionnalités nécessaires au traitement des paquets comme la mise à jour des tables de routage. La TCAM est une mémoire spéciale à très haute vitesse qui permet de rechercher l'ensemble de contenus dans un seul cycle d'horloge. Elle est très utilisée dans les routeurs hautes performances pour augmenter la vitesse du processus de recherche de route et de la transmission de paquets. Chaque adresse dans la TCAM est composée de l'adresse du réseau et celle de la machine. La SRAM permet d'enregistrer les informations sur le transfert des paquets à titre d'exemple le saut suivant (*next hop*) ou le port de sortie. Elle utilise le résultat de la recherche dans la TCAM pour s'acquérir des informations de renvoi du paquet à partir de son en-tête. La DRAM est utilisée comme une mémoire tampon lorsque les paquets sont traités par le processeur réseau.

5 L'approvisionnement des réseaux virtuels

La virtualisation des réseaux consiste à partager les ressources d'un réseau physiques (routeurs, commutateurs, etc.) entre plusieurs instances logiques d'un réseau constitué par des nœuds virtuels interconnectés par des liens virtuels. Cet ensemble forme un réseau virtuel sur lequel un fournisseur de services déploie un service de bout en bout pour ses utilisateurs finaux. Les éléments de base d'un environnement de virtualisation des réseaux sont représentés dans la figure 2.12. À partir d'une simple architecture réseau physique on peut créer plusieurs réseaux virtuels partageant les ressources disponibles de chaque nœud ou lien.

Le problème crucial dans la virtualisation des réseaux est l'approvisionnement des réseaux virtuels. Il s'agit de trouver la meilleure cartographie de nœuds et de liens virtuels sur une in-

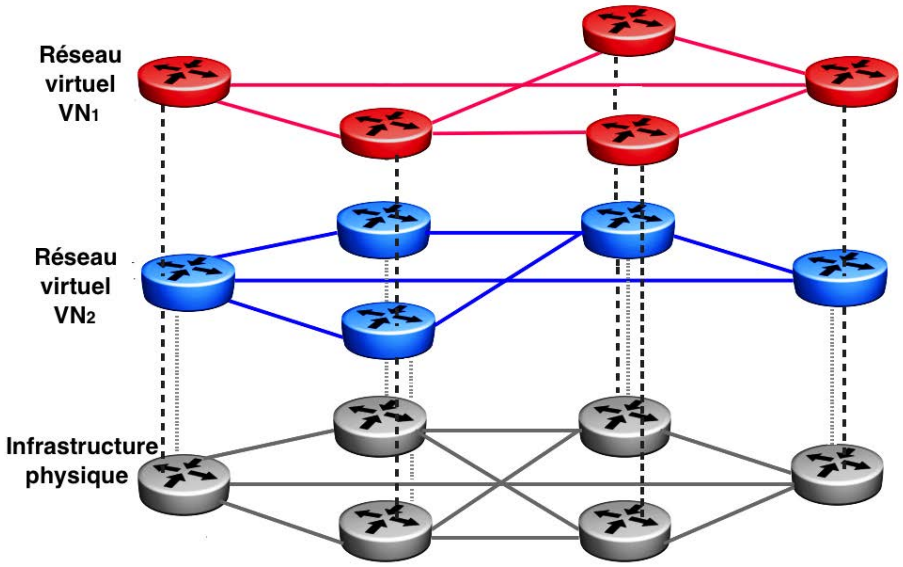


FIGURE 2.12 – Schéma d'une architecture réseau virtualisée

frastructure de ressources physiques dans le but d'incorporer un ou plusieurs réseaux virtuels. L'approvisionnement des réseaux virtuels suit les trois phases suivantes : la découverte des ressources, le mappage du réseau virtuel, et l'allocation des ressources [Esposito et al., 2013].
La découverte de ressources est le processus de surveillance de l'état de l'infrastructure physique à l'aide de plusieurs outils de monitorage. C'est le fournisseur de services qui est en charge de cette opération. Il observe, d'un côté, l'utilisation du processeur et de la mémoire au niveau des nœuds, et d'un autre côté, l'utilisation de la bande passante au niveau des liens.
Le mappage du réseau virtuel consiste à faire correspondre les demandes des réseaux virtuels aux ressources disponibles. Ces demandes peuvent inclure l'emplacement physique d'un nœud et la vitesse de traitement de son processeur réseau ou éventuellement la bande passante au niveau du lien. L'objectif de cette phase est la sélection du sous-ensemble des ressources qui peuvent potentiellement héberger ces VNs. Cette étape est considérée comme un problème NP-difficile car elle nécessite de combiner à la fois les contraintes des ressources des nœuds et des liens physiques et également les demandes de chaque VNs.
L'allocation des ressources consiste à attribuer une portion de chaque nœud et lien physique à un VN. Cette étape doit être répétée périodiquement afin de s'adapter au comportement dynamique des flux de chaque VN.
Plusieurs contraintes ont été considérées lors de l'approvisionnement des réseaux virtuels. Les longueurs des chemins entre deux nœuds virtuels interconnectés ont, par exemple, un effet probable sur les délais. D'autres contraintes comme l'utilisation des ressources et le niveau de stress doivent également être prises en compte. Le niveau de stress reflète le nombre d'entités virtuelles qui sont mappées sur un lien ou un nœud physique. À titre d'exemple une valeur élevée du niveau de stress d'un lien physique peut entrainer une hausse des taux de perte de paquets et des délais de transit. L'utilisation quant à elle évalue le rapport entre les ressources physiques dépensées

pour chaque VN et la capacité totale de ces ressources. Elle fournit un moyen de mesure plus précis que le niveau de stress, car elle prend en compte la disponibilité des ressources.
Les solutions existantes dans la littérature tentent de résoudre soit une tâche spécifique soit deux tâches combinées du problème d'approvisionnement des VNs.
Oppenheimer et al. [Oppenheimer et al., 2004] ont présenté SWORD, une approche qui combine la phase de découverte et celle du mappage pour les plates-formes étendues et partagées telles que PlanetLab [Peterson et al., 2003]. Les VNs qui sollicitent des nœuds physiques pour leur application doivent soumettre une demande de ressource sous forme d'une topologie de groupes interconnectés. Chaque groupe est considéré comme une classe d'équivalence de nœuds avec les mêmes contraintes que celles d'un nœud (mémoire, CPU) et d'un lien (bande passante, délais). Chaque VN spécifie une plage de valeurs nécessaires au niveau du nœud et du lien pour l'exécution du service. Il spécifie aussi son degré de tolérance pour chaque attribut des ressources physiques. Par exemple, il peut mentionner qu'il a besoin de 30 ms de latence ou moins entre deux nœuds au sein de chaque groupe. Cependant, il indique que le service pourrait être satisfait avec des latences jusqu'à 50 ms. La requête d'un VN, exprimée en XML, est acheminée d'abord avec des sockets TCP pour tous les nœuds exécutant SWORD. Ensuite, elle est reçue par le gestionnaire de requête de SWORD, qui émet à son tour une requête spéciale à tous les groupes concernés. Il reçoit en conséquence les réponses qu'il regroupe et qu'il optimise afin de trouver le meilleur chemin pour un VN. L'approche SWORD se concentre sur la sélection de l'ensemble des nœuds qui répondent aux exigences des VNs sans tenir compte des effets de la sélection sur la performance des VNs. En outre, SWORD n'offre pas la possibilité de spécifier les besoins de chaque VN en termes de bande passante entre deux nœuds. Il tient compte de performances de bout en bout sans se soucier qu'il peut exister un goulot d'étranglement au niveau d'un lien physique.
D'autres approches comme celle proposée par Yu et al. [Yu et al., 2008] introduit la coordination entre la phase de mappage et celle d'allocation des ressources. La solution proposée collecte les requêtes des VNs pendant un intervalle de temps. Elles sont injectées dans la file d'attente des demandes. Ces requêtes risquent d'être abandonnées si jamais aucune chance d'être servi durant un certain temps n'est accordée. L'approche introduit la notion de revenu qui est défini lors de l'accord entre les FSs et les FIPs. La solution traite les demandes dans l'ordre décroissant des revenus. L'approche utilise un algorithme glouton pour combiner l'opération du mappage de chaque nœud et chaque lien séparément et celle de l'allocation des ressources de l'infrastructure physique. L'algorithme surveille aussi les capacités disponibles au niveau de chaque ressource physique afin d'optimiser le processus d'allocation. La solution proposée effectue un contrôle d'admission tout en allouant statiquement les ressources des nœuds et des liens aux différents VNs. Elle ne permet pas le redimensionnement dynamique des ressources allouées. À titre d'exemple elle ne permet pas l'ajout ou la suppression des nœuds ou des liens virtuels ou même l'augmentation ou la diminution des capacités allouées d'un lien physique à un VN.
Dans ce qui suit nous allons présenter une formulation du problème d'approvisionnement des réseaux virtuels et un état de l'art des solutions existantes, traitant chaque phase du problème.

5.1 Formulation du problème

Dans la littérature, il existe plusieurs formulations du problème d'approvisionnement des réseaux virtuels. Belbekkouche et al. [Belbekkouche et al., 2012] et Rahman et al. [Rahman et al., 2010] ont proposé une formulation qui tient compte des contraintes de localisation géographique des nœuds virtuels, de la bande passante et des délais de transmission des paquets des liens virtuels. Fischer et al. [Fischer et al., 2013] ont inclus deux nouvelles fonctions dans leur formulation. La première concerne le mappage des nœuds et la seconde concerne le mappage des liens. Les différentes formulations existantes permettent de décrire les infrastructures physiques et virtuelles sans se soucier ni de l'interaction entre le FIP et le FS ni du rôle de chacun de ses acteurs dans le processus d'approvisionnement des réseaux virtuels. Nous proposons dans le cadre de ce travail d'utiliser la formulation basée sur celle proposée dans [Sun et al., 2013] en tenant compte de l'interaction entre les FIPs et les FSs, les contraintes de capacité et les demandes de partage des nœuds et des liens avec une fine granularité. On définit un réseau comme un graphe non orienté $G = (N, L, C^N, C^L)$, où N est un ensemble de nœuds et L est un ensemble de liens. On associe à chaque nœud $n \in N$ une contrainte $C^N(n)$ et à chaque lien $l \in L$ une contrainte $C^L(l)$.

Le fournisseur d'infrastructure physique FIP_k est en charge du monitoring des nœuds et des liens physiques. Son rôle est de déployer des protocoles et des mécanismes qui permettent de vérifier la disponibilité de chaque nœud et lien physique. Le réseau physique du fournisseur d'infrastructure physique FIP_k est noté $G_p^k = (N_p^k, L_p^k, CN_p^k, CL_p^k)$. On associe des contraintes à chaque lien et à chaque nœud virtuel. Au niveau du nœud, ce sont des contraintes sur la portion des ressources offertes pour traiter les paquets et sur les liens les contraintes de délais de transit des paquets.

Ce FIP notifie à chaque fournisseur de services la disponibilité de son infrastructure physique avec un vecteur $A_k(N_k, L_k)$ qui comprend les liens et les nœuds. Le FS collecte les vecteurs de tous les FIPs dans le but de les analyser pour décider sur quels nœuds et liens physiques il veut déployer et incorporer ses réseaux virtuels.

Le FS_i soumet une requête composée de $\alpha_i \in N$ nœuds virtuels, $\beta_i \in L$ liens virtuels et deux vecteurs de demande de capacité $CN_i(n)$ et $CL_i(l)$. Ces vecteurs définissent les besoins du fournisseur de services en termes de capacité des ressources sur chaque lien et nœud physique. Le fournisseur d'infrastructure physique s'assure que les capacités demandées de chaque nœud et lien ne dépassent pas la capacité totale disponible au niveau du réseau physique. Il peut soit accepter la requête du FS soit la rejeter en cas d'une détérioration des performances des réseaux virtuels en cours d'exécution. Lorsqu'il y a un rejet, le FS doit reprendre la phase du mappage afin de trouver le meilleur chemin pour chaque VN. Lorsque le FIP accepte la requête de partage du FS, il doit garantir le partage équitable et efficace de chaque nœud et lien de son infrastructure physique.

Le réseau virtuel VN_j déployé par FS_i est noté $G_v^{i,j} = (N_v^{i,j}, L_v^{i,j}, CN_v^{i,j}, CL_v^{i,j})$. Les contraintes $CN_v^{i,j}$ au niveau d'un nœud $n_v^{i,j}$ incluent les contraintes d'emplacement $loc(n_v^{i,j})$, la capacité du processeur réseau $C_{Processor}(N_v^{i,j})$ et la capacité des espaces mémoires $C_{Memory}(N_v^{i,j})$. Les contraintes $CL_v^{i,j}$ au niveau du lien incluent la bande passante $B(l_v^{i,j})$ et le délai $D(l_v^{i,j})$.

5.2 La découverte de l'infrastructure physique

Cette étape consiste à distinguer les ressources disponibles dans l'infrastructure physique d'un FIP. Elle nécessite la surveillance continue des éléments du réseau. Malgré l'importance de cette phase dans le problème d'approvisionnement des réseaux, il n'existe pas beaucoup de travaux de recherche qui s'intéressent à ce problème. Les quelques travaux existants ne cherchent pas à analyser la coopération entre plusieurs fournisseurs d'infrastructure physique durant cette phase. Afin de faciliter la phase de mappage des réseaux virtuels il est intéressant de combiner les résultats de la découverte de chaque FIP et les inciter à coopérer pour choisir le lien et le nœud adéquat garantissant la QoS pour chaque VN.
D'après Belbekkouche et al. [Belbekkouche et al., 2012] la découverte des ressources comporte deux parties : la description des ressources et le regroupement des ressources.

5.2.1 La description des ressources

La composante de base de la description des ressources est l'élément du réseau, qui est le nœud ou le lien. Chaque élément est décrit par des attributs fonctionnels et d'autres non fonctionnels. Les attributs fonctionnels comprennent le type d'élément, l'outil de virtualisation utilisé, le système d'exploitation de l'élément et la pile réseau. Les attributs non fonctionnels comprennent des paramètres de rendement, la capacité, l'utilisation et l'emplacement physique. Chaque FIP maintient un référentiel qui regroupe les attributs fonctionnels de ses ressources. Les attributs non fonctionnels sont mis à jour périodiquement. Les FIPs sont responsables de la notification liée à la disponibilité de leurs ressources. Chaque FS doit décrire les attributs de chaque élément et leurs besoins en ressources.
Dans la littérature, plusieurs spécifications et langages permettent de décrire les composantes et la topologie d'un réseau physique dans un environnement réseau virtualisé. NNDL (*Network Node Description Language*) [Dobrilovic et al., 2012] a été proposé dans le but de décrire la configuration d'un nœud avec suffisamment de détail pour permettre sa configuration dans un scénario de virtualisation des réseaux. Il a été conçu comme un langage basé sur XML et indépendant de la plate-forme qui prend en charge tous les concepts de base des réseaux. NNDL offre une structure flexible qui permet une extension facile pour un autre usage de ce langage.
VNUML [Galán et al., 2009] est un langage de description des ressources basé sur XML. Il permet de décrire la topologie du réseau dans un environnement virtualisé. Il fournit aussi un interpréteur de langage qui facilite la création et le déploiement des réseaux virtuels sur un nœud ou un lien physique du réseau.
VXDL [Koslovski et al., 2009] est un langage pour la spécification et la modélisation des réseaux d'interconnexion des ressources virtuelles. Il permet de décrire la topologie de façon individuelle ou regroupée d'une infrastructure réseau virtualisée. Il est aussi conçu sur XML pour la spécification des ressources dans les infrastructures virtuelles distribuées. La grammaire de ce langage permet de faire une description complète de l'environnement, en précisant les paramètres pour la classification des ressources.
VN-SLA [Fajjari et al., 2010] a été proposé pour fournir une spécification des ressources offertes par le fournisseur d'infrastructure pour l'approvisionnement de réseaux virtuels. Il permet de

décrire les propriétés de chaque élément du réseau d'une façon flexible. De nouveaux paramètres peuvent être ajoutés en fonction des exigences. VN-SLA permet aussi de spécifier les contraintes de capacité et d'emplacement des liens et des nœuds.
Les différents langages de description des ressources permettent d'aider un FS à découvrir et à choisir quelles ressources physiques disponibles, gérée par un FIP, vont lui permettre de remplir sa demande de partage et de maintenir les performances du VN déployé. Ceci est fait par la communication et la coopération avec les FIPs. En effet, la découverte de ressources est un processus continu qui est particulièrement compliqué lorsque plusieurs FIPs offrent leurs infrastructures pour être partagées. Le choix du langage de description des ressources dépend des exigences des VNs et des contraintes de capacité de l'infrastructure physique.

5.2.2 Le regroupement des ressources

Le regroupement des ressources consiste à disposer des informations sur les ressources physiques en arbre hiérarchique [Belbekkouche et al., 2012]. Cela facilite le processus de recherche en utilisant simplement un algorithme de correspondance qui commence à la racine de l'arbre. Si la racine ne satisfait pas l'exigence du fournisseur de services, la demande est rejetée. Si l'exigence de ressources est satisfaite au niveau de la racine, le processus de recherche passe à travers les branches de l'arbre. Le but de cette phase est de regrouper et de classifier les ressources physiques dans plusieurs catégories afin de minimiser la complexité de la phase de découverte.

Houidi et al. [Houidi et al., 2009] ont proposé un *Framework* pour la description et le regroupement de ressources dans un environnement réseau virtualisé. Ce travail permet de décrire et de classifier les ressources offertes par un fournisseur d'infrastructure. Les auteurs ont présenté une technique de *clustering* conceptuel classant les ressources ayant des caractéristiques similaires dans une hiérarchie de groupe dans un arbre hiérarchique. Ils présentent aussi un algorithme de correspondance dans l'arbre dans le but de faciliter le processus d'approvisionnement des réseaux virtuels. Le même *Framework* proposé a été aussi utilisé dans [Lv et al., 2010] pour modéliser le problème d'apprivoisement des réseaux virtuels dans une infrastructure réseau gérée par plusieurs FIPs.
Medhioub et al. [Medhioub et al., 2011] ont présenté un nouvel algorithme de *clustering* qui permet de classer les ressources en fonction de leur importance selon un score obtenu grâce à l'analyse statistique des requêtes des réseaux virtuels. L'approche cherche à classer et à construire des arbres à l'aide des attributs et leurs valeurs. Le but de l'algorithme proposé est de faciliter le processus d'approvisionnement des réseaux virtuels. L'algorithme permet de réduire la complexité de l'ordre $O(2^n)$ en ordre $O(m)$ (où n est le nombre de nœuds et m est les différents attributs) en mettant l'accent sur les attributs de ressources virtuelles plutôt que la correspondance entre les nœuds.
Amarasinghe et al. [Amarasinghe et al., 2012] ont proposé ADVNE, une approche basée sur l'agrégation qui permet d'utiliser les informations des attributs non fonctionnels pour améliorer l'efficacité du processus de découverte. L'approche cherche à éviter la surveillance continue du réseau physique pour chaque FIP. Elle minimise le nombre de messages échangés entre les nœuds physiques.

5.3 Le mappage du réseau virtuel

Le mappage des réseaux virtuels est considéré comme la phase la plus critique lors de l'approvisionnement des VNs puisqu'il s'agit d'un problème NP-difficile. Plusieurs approches ont été proposées pour trouver le meilleur chemin pour un VN dans une infrastructure réseau virtualisée. On peut classifier ces solutions selon l'arrivée des requêtes des VNs au cours du temps. Les approches comme celles de Houidi et al. [Houidi et al., 2008], Zhu et Ammar [Zhu and Ammar, 2006] et Lu et Turner [Lu and Turner, 2006] supposent que les requêtes des VNs sont parfaitement connues à l'avance. Zhu et Ammar [Zhu and Ammar, 2006] ont présenté un ensemble d'algorithmes heuristiques pour l'affectation des ressources du réseau physique à des réseaux virtuels. L'idée principale de ce travail est d'identifier la ressource et leur niveau de stress tout en supposant que les ressources ont une capacité infinie. Le niveau de stress reflète le nombre de réseaux virtuels qui utilisent ces ressources. Les algorithmes proposés considèrent cette métrique comme une contrainte lors du mappage et cherchent à la minimiser. Les auteurs présentent deux approches. La première offre un mappage fixe et la seconde un mappage qui peut évoluer durant l'exécution du VN. Pour le mappage fixe, les auteurs proposent d'identifier en premier lieu le nœud qui a le plus de nœuds et de liens voisins disponibles. Ensuite, ils proposent d'identifier le reste des nœuds qui vont être utilisés par le VN en fonction leurs niveaux du stress et de la distance entre eux. Enfin, une fois tous les nœuds sont identifiés, l'algorithme du plus court chemin est utilisé pour le mappage des liens virtuels. La seconde approche, qui concerne le mappage avec une reconfiguration des nœuds et des liens, se fait périodiquement en deux étapes. La première étape identifie les VNs critiques qui nécessitent une reconfiguration du mappage des ressources en fonction du niveau de stress de chaque ressource. La seconde étape choisit les nouveaux nœuds et les nouveaux liens qui vont être utilisés par le VN en minimisant la fonction de coût de reconfiguration.
Lu et Turner [Lu and Turner, 2006] sont à l'origine d'une nouvelle méthode de mappage réseau virtuel qui permet d'accommoder plusieurs schémas de trafic (*traffic pattern*). Chaque VN peut réserver plusieurs ressources physiques en définissant les contraintes de son trafic. L'approche vise à trouver le meilleur mappage dans une famille de topologies d'épine dorsale en étoiles (*backbone-star*) composée de plusieurs sous-ensembles de nœuds. La solution proposée cherche à obtenir suffisamment de capacités au niveau des ressources accueillant tout modèle de trafic avec contraintes. Elle considère le comportement du trafic comme une métrique pour trouver les besoins en termes de bande passante pour chaque VN sans se soucier des contraintes imposées au niveau des nœuds.
Houidi et al. [Houidi et al., 2008] ont présenté une approche distribuée pour le mappage des réseaux virtuels. Ils présentent une solution pour l'auto-organisation des VNs chaque fois qu'une nouvelle requête de partage arrive à FIP. Le FIP déploie des agents autonomes et intelligents au niveau de chaque nœud physique. Les agents coopérèrent en échangeant cinq types de messages entre les différents nœuds de l'infrastructure physique. Le protocole suppose que toutes les requêtes arrivent au niveau d'un groupe de nœuds appelés *hub and spoke cluster* en charge du mappage des VNs. Chaque nœud physique exécute trois algorithmes de mappage dont l'algorithme du plus court chemin afin de mettre à jour les informations sur les capacités des nœuds et

des liens physiques. Les auteurs proposent l'utilisation d'un réseau de signalisation et de contrôle externe pour atténuer le problème de la surcharge des ressources.
D'autres approches comme celles proposées dans [Butt et al., 2010, Chowdhury et al., 2010, Yu et al., 2008, Alkmim et al., 2013, Chowdhury et al., 2012] traitent à la volée les différentes requêtes des VNs. Ils supposent que les requêtes des réseaux virtuels arrivent simultanément. Chowdhury et al. [Chowdhury et al., 2012] présentent une solution qui introduit une corrélation entre le mappage des nœuds et celui des liens. Ils proposent deux algorithmes, l'un déterministe et l'autre aléatoire. Ces algorithmes traitent d'abord la bande passante de chaque lien comme un problème de multi-flots. Il s'agit de faire passer un flux de matière à travers un graphe réseau en respectant les capacités des arcs et celles des nœuds. Ensuite ces algorithmes cherchent à trouver le meilleur chemin pour un VN à l'aide d'une formulation en programmation mixte en nombres entiers.
Alkmim et al. [Alkmim et al., 2013] ont proposé une formulation mathématique du problème de mappage des réseaux virtuels. Ils présentent plusieurs algorithmes basés sur la programmation linéaire en nombres entiers cherchant à minimiser la portion de la bande passante offerte pour chaque VN. Ces algorithmes visent aussi à réduire le temps requis pour instancier un routeur virtuel sans se soucier de sa charge actuelle. La solution ne tient pas compte de la connaissance antérieure des requêtes de réseaux virtuels en supposant que les ressources de l'infrastructure physique ont une capacité limitée.
On peut également considérer la décomposition du problème du mappage comme un critère de classification des approches. Les solutions comme celles proposées dans [Houidi et al., 2008, Yu et al., 2008, Butt et al., 2010, Razzaq et al., 2012] décomposent le problème en deux parties : le mappage des liens et le mappage des nœuds. Butt et al. [Butt et al., 2010] ont présenté plusieurs mécanismes de mappage des liens et de nœuds virtuels en fonction de l'impact sur la connectivité et du taux d'utilisation de l'infrastructure physique. Les auteurs ont introduit deux nouvelles métriques lors du mappage. La première, mesure la probabilité sur les capacités restantes dans un réseau lorsqu'une ressource physique est indisponible. La seconde, mesure combien de VNs sont affectés par l'indisponibilité des ressources. L'algorithme proposé choisit séparément les liens et les nœuds les moins critiques lors du mappage.
Lu et al. [Lu et al., 2013], Melo et al. [Melo et al., 2012], Chowdhury et al. [Chowdhury et al., 2012], Wang et al. [Wang et al., 2012b] et Inführ et Raidl [Inführ and Raidl, 2011] cherchent la meilleure topologie d'un VN sans faire de distinction entre nœud et lien. Leurs approches se basent sur la programmation linéaire en nombres entiers. Lu et al. [Lu et al., 2013] ont formulé cette phase comme un problème d'optimisation linéaire avec plusieurs contraintes. Les contraintes sont des contraintes de capacité, des contraintes de flux et des contraintes d'emplacement de la ressource. La solution construit d'abord un modèle de graphe à partir de l'infrastructure physique. Ensuite, en optimisant une fonction objective, la solution cherche le meilleur chemin qui maximise le revenu du mappage du VN et minimise son coût. Ce coût est exprimé et termes d'utilisation de CPU au niveau des nœuds et de la bande passante au niveau du lien.
Melo et al. [Melo et al., 2012] proposent de formuler le problème du mappage comme un programme linéaire avec une fonction objective qui cherche à réduire la charge maximale pour

chaque ressource physique. La solution assimile aussi les contraintes des capacités des ressources à un problème de multi-flots. Elle cherche à trouver la meilleure correspondance des liaisons virtuelles au même temps que les nœuds virtuels. L'heuristique proposée considère le mappage consommant le moins de liens et de nœuds physiques comme solution optimale à ce problème. Les auteurs de [Inführ and Raidl, 2011] ont adopté la même formulation mathématique du problème de mappage des VNs mais en considérant des contraintes supplémentaires comme le délai au niveau d'un lien, le routage et l'emplacement des ressources physiques.
Ivaturi et wolf [Ivaturi and Wolf, 2014] ont proposé une solution pour rendre la virtualisation des réseaux utile pour les applications temps réel sensibles aux délais. Ils présentent un algorithme de mappage qui assure un compromis entre un délai faible des paquets d'un VN et une utilisation efficace de l'infrastructure physique. L'approche proposée consiste à déterminer le meilleur chemin qui minimise les délais de bout en bout.

5.4 Allocation des ressources physiques pour le réseau virtuel

Dans les travaux de recherche traitant le problème de l'approvisionnement des réseaux virtuels, L'allocation des ressources n'a pas reçu une attention particulière. La majorité des travaux existants se focalisent sur la phase du mappage des VN, en supposant que l'allocation des ressources s'effectue implicitement une fois le meilleur chemin pour VN est identifié en allouant une portion fixe de chaque nœud et de chaque lien. Les approches d'allocation des ressources dans un environnement réseau virtualisé peuvent être classées en deux catégories : approches statiques et approches dynamiques.
Les approches statiques proposées dans [Lu et al., 2013, Melo et al., 2012, Zhu and Ammar, 2006, Lu and Turner, 2006] n'offrent aucune possibilité aux FSs de solliciter plus ou moins de ressources pour leurs VN déployés. Les travaux, dans le cadre de ces approches, prennent en compte les contraintes de capacité et d"utilisation de l'infrastructure physique lors du mappage sans considérer les besoins en QoS de chaque VN. Les algorithmes de mappage proposés cherchent à trouver la meilleure correspondance d'un VN sur un réseau physique géré par plusieurs FIPs. Cette opération engendre des problèmes de sous utilisation des ressources physiques et de non-équité lors du partage. En effet, un VN peut consommer toute la bande passante d'un lien et toute la capacité du CPU d'un nœud sans rencontrer un mécanisme d'empêchement.
Les approches dynamiques ont été développées pour palier ce défaut en offrant la possibilité de reconfigurer les portions des ressources disponibles dans chaque nœud et chaque lien.
Dans ce qui suit, nous présentons une revue bibliographique sur les différentes approches dynamiques proposées. Les approches développées dans les travaux antérieurs se focalisent sur l'allocation de la bande passante. Ces travaux [Zhou et al., 2010, Wang et al., 2011] utilisent la théorie des jeux et les approches économiques pour effectuer le partage de la bande passante d'un lien physique entre plusieurs VNs.
Zhu et al. [Zhou et al., 2010] ont proposé un modèle de jeux non-coopératifs où chaque VN essaie de maximiser sa fonction d'utilité lors du partage de l'infrastructure physique. La fonction d'utilité du VN dépend principalement de la bande passante offerte, de l'état du lien physique et du prix que doit payer le VN pour partager ce lien. Les auteurs proposent un algorithme

itératif qui permet d'atteindre l'équilibre de Nash lors du partage de la bande passante tenant compte des performances de chaque VN. L'approche vise à allouer dynamiquement une portion de la capacité de chaque lien physique pour éviter la congestion de l'infrastructure physique. Les auteurs ne discutent pas l'équité lors de l'allocation de bande passante. Ils supposent aussi que c'est le rôle du FIP d'inciter le FS à changer de stratégie pour s'assurer que les demandes des VNs ne dépassent pas la capacité du lien. Sans l'intervention du FIP la solution ne permet pas d'assurer le partage efficace et équitable du lien physique.

Wang et al. [Wang et al., 2011] présente un modèle similaire à [Zhou et al., 2010] avec une nouvelle fonction d'utilité que les différents VNs cherchent à maximiser durant le jeu. Les VNs soumettent un vecteur d'enchères pour exprimer leurs besoins en termes de bande passante. Les auteurs ont montré l'existence d'un état d'équilibre de Nash dans le jeu proposé mais n'ont pas considéré les contraintes de QoS pour chaque VN lors du choix de la meilleure stratégie dans le jeu. La solution proposée ne garantit pas de la bande passante aux différents VNs si l'un d'eux cherche à consommer toute la capacité disponible sur un lien.

Shun-li et al. [Shun-li et al., 2011] supposent que la portion des ressources demandées par un FS doit dépendre du temps et des revenus des FSs afin d'éviter une sous-utilisation. Ils présentent un modèle autonome d'allocation des ressources fondé sur les revenus des FSs au cours d'un intervalle de temps. En effet, leur approche utilise les prévisions de la demande de ressources, les données historiques et les informations sur l'allocation actuelle. Le système proposé collecte les demandes de FSs et tente d'allouer dynamiquement ces ressources à chaque intervalle de temps en maximisant le revenu du FS. Lors du partage, la solution proposée cherche à maximiser les revenus de chaque FIP sans se soucier des contraintes de QoS de chaque VNs déployé. En outre, ils ne présentent pas le mécanisme de tarification utilisé par les différents FSs lors du partage.

He et al. [He et al., 2008] ont proposé une approche dite DaVinci qui permet de réaffecter périodiquement une portion de la bande passante à plusieurs VN en tenant compte des niveaux de congestion de l'infrastructure physique. Au niveau de chaque lien physique existe un coordinateur de bande passante qui ajuste la portion de la capacité offerte à chaque VN. Cette solution se base sur une théorie d'optimisation afin de maximiser le rendement global de tous les VNs. Chaque réseau virtuel exécute un protocole distribué maximisant la fonction objective de performance. Cette dernière varie en fonction de la classe du trafic que transporte chaque VN.

Wei et al. [Wei et al., 2010] ont présenté un algorithme d'allocation dynamique de la bande passante dans un environnement réseau virtualisé. Les auteurs caractérisent l'allocation comme un problème de multi-flots avec l'utilisation de la bande passante par chaque VN comme critère d'optimisation. Leur approche tient compte du comportement On/Off des flux de chaque VN et permet de prédire leurs comportements dynamiques. La moyenne du débit des flux de chaque VN se calcule périodiquement. Elle est utilisée dans le but de trouver les besoins en bande passante pour chaque VN.

Garg et Saran [Garg and Saran, 2000] ont proposé une approche pour le partage dynamique de la capacité d'un lien physique. Ils considèrent deux types de messages échangés entre le FS et le FIP. Le premier type est pour demander plus de bande passante et le second pour demander moins. La capacité actuelle d'un lien à un VN peut être modifiée par l'envoi d'une demande

d'augmentation ou de réduction au FIP. Ce dernier déploie des ordonnanceurs de paquet SFS (*Stochastic Fair Sharing*) sur chaque lien physique en supposant que les paquets de chaque VN arrivent selon une loi de Poisson. Il traite les demandes des différents FSs et met à jour les poids attribués pour chaque VN lors du partage. En cas de surcharge, le FIP envoie un message pour demander la réduction de la bande passante allouée à un VN.

Botero et al. [He et al., 2008] considèrent le problème de l' allocation de bande passante comme un problème d'optimisation qui cherche à maximiser l'utilisation des ressources de l'infrastructure physique. Ils proposent un mécanisme équitable pour résoudre la répartition de la bande passante dans un environnement réseau virtualisé. Le travail des auteurs se concentre sur la gestion du problème lorsqu'un ou plusieurs VNs ont un goulot d'étranglement. Ils proposent une stratégie qui tente de répartir équitablement la bande passante entre les VNs en trouvant la valeur du minimum de VNs ayant reçu une capacité dépassant les besoins de chacun d'eux. Ainsi, l'algorithme proposé évite les goulots d'étranglement.

Wenzhi et al. [Wenzhi et al., 2012] ont abordé le problème de la répartition adaptative de la bande passante entre les VNs dans un environnement de virtualisation des réseaux. Ils considèrent un scénario de cinq types de VNs ayant différents besoins en QoS. Les auteurs utilisent un modèle hydraulique connecté à chaque VN et sur chaque lien physique. Le modèle repose sur la pression de la congestion interne du lien afin d'ajuster l'allocation de la bande passante entre les VNs. Un algorithme d'ordonnancement des paquets est proposé pour équilibrer la pression de chaque VN sur chaque lien. Le mécanisme d'allocation adaptatif de la bande passante se base essentiellement sur cette pression pour repartir la capacité du lien entre les différents VNs.

Wang et al. [Wang et al., 2012a] suggèrent l'utilisation des mesures de comportement du trafic de chaque VN et l'ajustement dynamique de la portion de chaque ressource offerte. Les auteurs ont développé une approche permettant de déterminer la portion exacte de chaque ressource offerte, à chaque VN, satisfaisant aussi plusieurs contraintes de QoS. Ils utilisent les statistiques de performance de TCP comme métrique de l'estimation de la QoS. En effet, l'approche considère deux services déployés sur les VNs. Le premier est un service web et le second est un service vidéo à la demande. La solution proposée cherche à analyser l'interaction entre le comportement de l'utilisateur et les performances du réseau. Les auteurs supposent que le FIP mesure les probabilités qu'un utilisateur quitte lorsqu'il n'est pas satisfait du service, le taux d'arrivée des paquets et le débit de la connexion TCP. À partir de ces valeurs le FS estime la bande passante nécessaire pour chaque utilisateur. Il peut, par la suite, ajuster dynamiquement sa demande en fonction de ses besoins.

Dans un contexte autre que la virtualisation des réseaux, Padala et al. [Padala et al., 2009] présentent un système de commande des ressources qui s'adapte automatiquement aux changements dynamiques dans une infrastructure *data center* virtualisée partagée pour satisfaire les besoins des applications. Les auteurs utilisent un modèle d'estimation de ligne qui sert à capturer la relation entre les performances de l'application et l'allocation des ressources des serveurs. Ensuite, ils utilisent plusieurs contrôleurs qui permettent de déterminer automatiquement la portion des ressources nécessaires pour satisfaire les besoins d'une application. La solution permet de détecter les goulots d'étranglement sur les serveurs partagés et de fournir une différenciation de

services pour les applications. Le principe nous parait intéressant et nous inspire pour l'allocation dynamiques des ressources physiques dans un environnement réseau virtualisé.
Dans ce travail de recherche nous adoptons l'idée de Wang et al. [Wang et al., 2012a] sur l'importance de la QoS offerte pour chaque VN comme critère dans l'allocation des ressources. La majorité des solutions existantes ne tiennent pas compte de cette contrainte. Ces solutions cherchent principalement à offrir un meilleur partage des capacités des ressources de l'infrastructure physique sans surveiller les performances des VN en termes de QoS.
Lors de l'allocation des ressources et de l'exécution d'un service sur chaque VN, il est nécessaire de surveiller les métriques de la QoS pour mesurer l'impact du partage de l'infrastructure physique. Ces métriques sont généralement le débit, le délai et la gigue.

— Le débit du trafic de chaque VN décrit la vitesse maximale de transmission des paquets entre deux nœuds physiques.

— Le délai décrit le temps nécessaire pour transmettre un paquet d'un nœud à un autre.

— La gigue mesure la variation de la latence au cours du temps. Dans un scénario de virtualisation de réseau, la gigue peut dépendre de l'état de l'infrastructure physique du réseau ou de l'outil de virtualisation employé.

6 Conclusion

Dans ce chapitre nous avons présenté une étude bibliographique détaillée des techniques de virtualisation des réseaux et un état de l'art de l'approvisionnement des réseaux virtuels. Nous avons soulevé le handicap des solutions existantes qui n'offrent pas de fine granularité dans l'allocation des nœuds et des liens physiques.
Tout le long de ce travail nous allons essayer d'amener des éléments de réponse pour pallier ces défauts. Nous partons d'abord de l'idée que l'infrastructure physique est un réseau SDN où chaque commutateur est attaché à un contrôleur facilitant la phase de découverte des ressources. Ensuite, chaque FS choisit l'algorithme de mappage qui satisfait au mieux ses contraintes en termes de routage ou de délai. Nous nous focalisons dans ce qui suit sur la phase d'allocation des ressources en offrant la possibilité à tout FS d'exprimer les besoins de chaque VN en cours d'exécution. Nous modélisons, enfin, l'interaction entre un FIP et un FS lors du partage. Chaque FS surveille les performances de ses VNs et décide de demander plus ou moins de ressources auprès d'un FIP. Ce dernier veille à son tour sur le fait que toutes les portions de la ressource physique allouée, pour différents FSs, ne dépassent pas la capacité totale de cette ressource comme étant un mécanisme de contrôle d'admission.

Chapitre 3

Gestion de la QoS dans un environnement domestique virtualisé

Sommaire

1 Introduction **41**
2 Le réseau Internet haut débit **42**
2.1 Internet haut débit par xDSL 43
2.2 Internet haut débit par câble 45
3 Virtualisation des passerelles domestiques **45**
4 SDN et réseaux domestiques connectés à Internet haut débit **46**
4.1 Gestion de la QoS dans les réseaux SDN 47
4.2 Gestion de la QoS dans les réseaux domestiques 49
5 FlowQoS : Gestion de la QoS dans un environnement domestique virtualisé . . **50**
5.1 Architecture de FlowQoS 51
5.2 Le classificateur de flux 52
5.3 Le lissage du trafic 53
5.4 Evaluation de FlowQoS 54
6 VidShaper : Allocation dynamique de la bande passante pour les flux de streaming vidéo adaptatif **60**
6.1 Architecture de VidShaper 61
6.2 Evaluation de VidShaper 62
7 Conclusion **63**

1 Introduction

Dans les réseaux domestiques connectés à Internet haut débit, une application peut consommer toute la bande passante disponible au détriment d'autres applications, dégradant ainsi la performance globale de ces applications. Une solution à ce problème consiste à allouer une bande passante à différents flux en fonction du type d'application à l'entrée du réseau domestique. Malheureusement, ce mécanisme de gestion de la qualité de service (QoS) n'est pas facile à déployer

pour plusieurs raisons. L'une des raisons liée à ceci réside dans le fait que les routeurs domestiques ne sont pas assez puissants pour effectuer la classification des flux. A ceci s'ajoute, le fait que les utilisateurs n'ont pas assez de connaissance pour configurer les paramètres de QoS.
Dans ce chapitre nous présentons FlowQoS, une approche fondée sur SDN pour l'allocation de la bande passante en fonction de l'application. FlowQoS est composé de deux modules : un classificateur de flux et un limiteur de la bande passante. Les utilisateurs peuvent allouer une fraction de la bande passante pour chaque application différente. Le classificateur de flux permet d'abord d'identifier les applications et ensuite cette information est reliée à un contrôleur SDN qui installe dynamiquement des règles de limitation de la bande passante du trafic des flux réseaux de chaque application. Nous proposons un premier classificateur à base de DNS qui permet l'identification des applications s'exécutant sur les ports HTTP/HTTPS. Un second classificateur effectue une inspection légère (*lightweight packet inspection*) des paquets pour identifier les flux non-HTTP.
Par la suite, nous présentons,VidShaper, une technique basée sur le lissage du trafic qui assure une meilleure qualité d'expérience (*Quality of Experience* (QoE)) quand plusieurs lecteurs de streaming vidéo adaptatif sont actifs dans le même réseau domestique.
Dans ce travail, nous montrons la faisabilité de l'allocation de la bande passante pour plusieurs flux de données sur un nœud physique virtualisé à l'aide de SDN. Dans un scénario de large échelle, un fournisseur de services peut contrôler la charge offerte pour chacun de ses VNs en déployant un mécanisme de partage de la bande passante sur le contrôleur SDN qui gère un ou plusieurs nœuds virtualisés.

2 Le réseau Internet haut débit

Un réseau domestique consiste à interconnecter des équipements domestiques hétérogènes. Par le biais d'une connexion Internet haut débit, ces équipements peuvent accéder à nombreux services offerts sur la grande toile. Les fournisseurs d'accès Internet (FAI) rivalisent pour offrir aux utilisateurs des réseaux domestiques une connexion Internet de plus en plus rapide et de plus en plus efficace. Sur le marché actuel, deux technologies sont disponibles pour fournir de l'Internet haut débit ; la xDSL et le câble.
La gestion de la QoS dans les réseaux domestiques est une tâche ardue. Les utilisateurs du réseau s'attendent à une bonne performance pour les données, la voix et la vidéo, le tout avec une QoS raisonnable. Garantir une bonne qualité de service pour ces applications implique une configuration des priorités par flux. Le but est d'empêcher une application Internet de dégrader la performance globale lors du partage de la bande passante. Dans ce contexte, le trafic des applications gourmandes en bande passante (vidéo streaming) et des applications interactives (VoIP, jeux) se disputent relativement des ressources limitées de la bande passante. Une approche commune pour faire face aux contraintes de limitation de la bande passante consiste à configurer le routeur domestique pour offrir une priorité à certains flux avant d'autres. De nombreux mécanismes de qualité de service ont été proposés, normalisés et mis en place [Gozdecki et al., 2003, Newman et al., 1996, Aurrecoechea et al., 1998, McDysan, 1999]. Ces mécanismes

n'ont pas été déployés dans les routeurs domestiques d'accès à Internet pour plusieurs raisons. La première est due au fait que les utilisateurs ont une connaissance limitée de la méthodologie de gestion de la qualité de service. Les dispositifs existants ne peuvent pas être facilement configurés pour exécuter des fonctions de QoS ne reposant pas sur des applications spécifiques, des équipements ou des utilisateurs. À titre d'exemple, un utilisateur ne saura pas quel port spécifier lors de la configuration de la priorité pour les flux VoIP. La seconde raison est liée au fait que la plupart des passerelles domestiques, mis à part les nouvelles box ADSL (Freebox, Livebox, etc.), ont des ressources mémoire et des ressources de calcul limitées. Ils n'arrivent pas à effectuer de la classification à la volée (*on-the-fly*). Par conséquent, un équipement spécial dans la maison doit être déployé par les FAIs.

En effet les nouvelles applications disponibles sur Internet comme le streaming vidéo ont des besoins de QoS que l'architecture actuelle d'Internet doit supporter. L'introduction de nouvelles technologies comme streaming vidéo adaptatif via HTTP a permis de réduire le coût en bande passante pour le FAI et d'optimiser l'expérience utilisateur quel que soit son débit. Des études [Wang and Zink, 2014] ont montré que Netflix et Youtube sont responsables de plus de 50% du trafic Internet aux USA. D'ici trois ans, ces deux fournisseurs de vidéo à la demande seront responsables d'au moins 80% du trafic.
Le streaming vidéo adaptatif via HTTP est normalisé par le standard MPEG-DASH (ISO/IEC 23009-1). Dans ce qui suit, nous allons expliquer brièvement le fonctionnement de cette technologie. Un fournisseur de VoD comme Netflix ou Hulu dispose d'une essence audiovisuelle appelée vidéo brute (*raw video*) constituée de plusieurs composantes : vidéo, pistes audio, sous-titrage. La première étape consiste à redimensionner la taille la vidéo en fonction de la taille du terminal du streaming. Généralement, il y a trois tailles de vidéos ; la première est dédiée aux téléphones mobiles, la seconde aux tablettes et la derrière aux ordinateurs et aux télévisions. L'étape suivante est l'encodage de ces différents formats de vidéos en plusieurs versions avec différents débits d'encodage (*bitrate*). Chaque vidéo est segmentée en plusieurs morceaux de vidéos de taille fixe (par exemple 4 secondes) appelés « chunks ». Le fournisseur de VoD crée un fichier descripteur appelé *manifest file* sous un format XML qui comprend l'adresse URL des différents chunks et le débit d'encodage de la vidéo. Le descripteur est exploité par le lecteur, côté client, pour accéder à la vidéo. Le tout est par la suite chargé sur un serveur Web. Sur la base du fichier descripteur et de l'estimation de la bande passante disponible du réseau du client, le lecteur de vidéo sélectionne le format et le débit approprié puis accède aux chunks en envoyant des `HTTP GET`. Le lecteur dispose d'un buffer qui lui permet de se libérer des variations de débit et de garantir un affichage fluide. En cours d'affichage, le lecteur de vidéos continue à analyser la bande passante disponible afin de sélectionner dynamiquement les chunks suivants. La figure 3.1 illustre les différentes étapes du fonctionnement du streaming vidéo adaptatif via le protocole HTTP.

2.1 Internet haut débit par xDSL

La technologie xDSL permet de faire passer du haut débit sur la paire de cuivre utilisée pour les lignes téléphoniques. On distingue deux familles de xDSL. L'une utilise une transmission

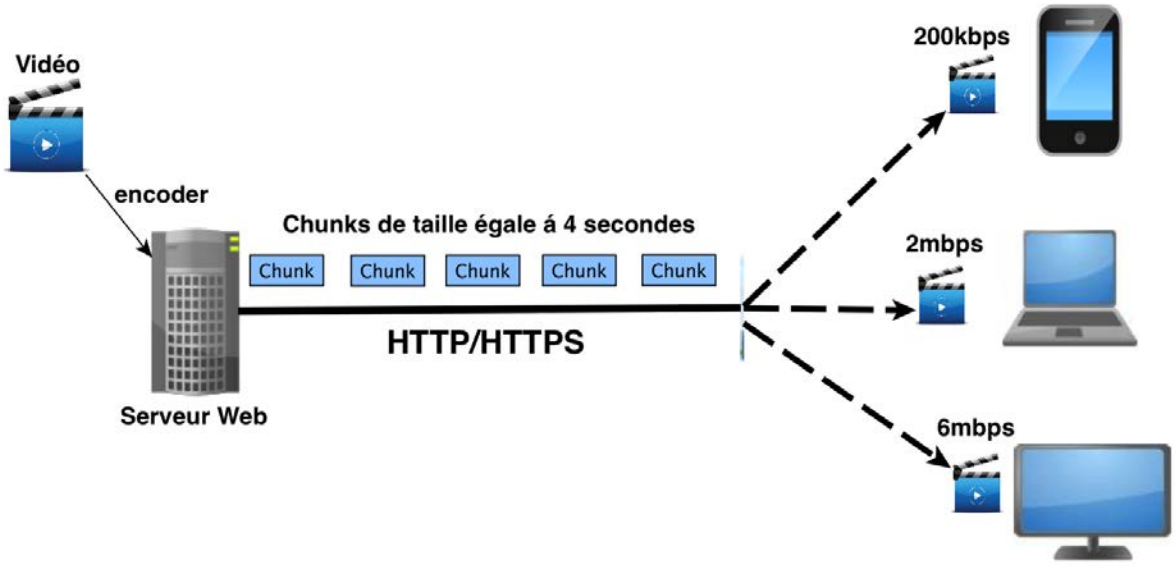


FIGURE 3.1 – Le fonctionnement du streaming vidéo adaptatif via HTTP

symétrique et l'autre utilise une connexion asymétrique. L'idée est de se servir des fréquences supra vocales qui ne sont pas utilisées par le téléphone. Cette technique nécessite l'installation d'équipements de communication dédiés aux deux extrémités de la ligne téléphonique. Les trafics de l'ADSL et du téléphone transitent par le même fil jusqu'à la centrale téléphonique, mais les deux fonctionnent indépendamment. Cette centrale, généralement appelée nœud de raccordement d'abonnés (NRA), envoie les basses fréquences, sur lesquelles passe la voix, vers le réseau téléphonique commuté et les hautes fréquences, sur lesquelles passent les données vers le réseau du FAI. Elle joue le rôle d'un répartiteur en charge du tri entre les lignes des abonnés et qui par la suite, les transfère vers les multiplexeurs d'accès DSL respectifs des FAIs correspondant à chaque ligne. Ce multiplexeur, appelé DSLAM (*Digital Subscriber Line Access Multiplexer*), est l'équipement central auquel se connectent les utilisateurs DSL. Il est composé d'un châssis sur lequel sont installées les cartes modem et les cartes filtres. Le DSLAM est l'équipement qui fait la liaison entre les lignes des abonnés et le réseau du FAI. Il sert à gérer les différents trafics. Cette étape se fait, en premier lieu, par une répartition du trafic par type voix et ADSL. Par la suite, le DSLAM les envoie sur le réseau de fibre optique du fournisseur pour être acheminé vers son infrastructure. Le trafic de la voix est envoyé par la carte filtre vers le réseau commuté. Par contre celui de l'ADSL est envoyé vers la carte modem. Cette carte modem rassemble les flux numériques de chaque abonné et les fait converger par multiplexage temporel où les données sont transportées en IP ou en ATM sur un seul lien à fort débit vers le Broadband Access Server (BAS). Il s'agit du serveur d'authentification du client qui est en charge, aussi, de la transmission des paramètres IP. La figure 3.2 illustre les différentes composantes de l'architecture de l'ADSL.

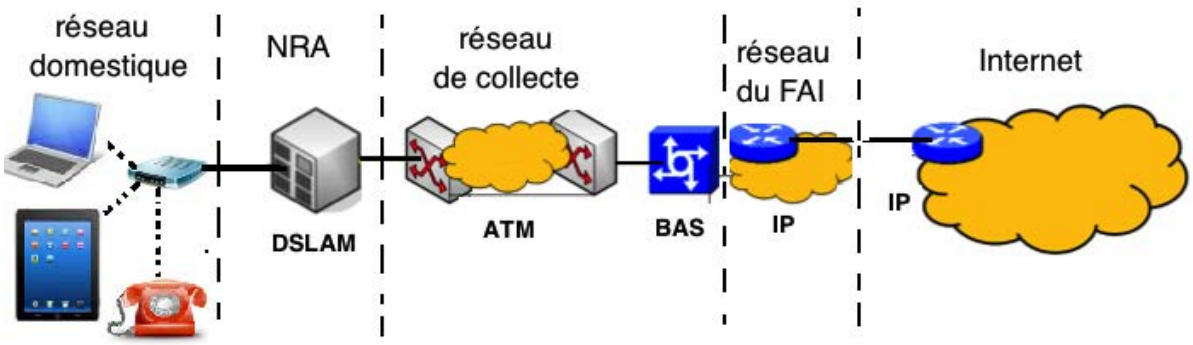


FIGURE 3.2 – Architecture de L'ADSL

2.2 Internet haut débit par câble

Il existe plusieurs architectures fournissant une connexion Internet haut débit par câble. La différence entre elles réside dans la proportion respective des segments fibres optiques et des segments câbles coaxiaux. Ces architectures ont été proposées pour la diffusion de télévision en mode *broadcast* et par la suite elles ont été adaptées pour l'Internet haut débit.

On peut distinguer trois architectures principales :

— L'architecture totalement coaxiale qui ne supporte pas l'acheminement des flux hauts débits,

— L'architecture HFC (*Hybrid fibre-coaxial*), qui comporte la fibre optique et le câble coaxial. Elle est généralement adaptée au haut débit. Elle utilise une architecture en étoile de câbles coaxiaux qui desservent des groupes d'abonnés. Pour chaque service, on utilise des bandes de fréquences spécifique dans les deux sens de la transmission.

— Les architectures FTTx (*Fiber to the x*), dans laquelle la fibre optique est déployée de bout en bout jusqu'à l'abonné. Cette architecture, permet d'atteindre de très haut débit car elle remplace le câble coaxial de la ligne vers l'abonné par la fibre optique.

3 Virtualisation des passerelles domestiques

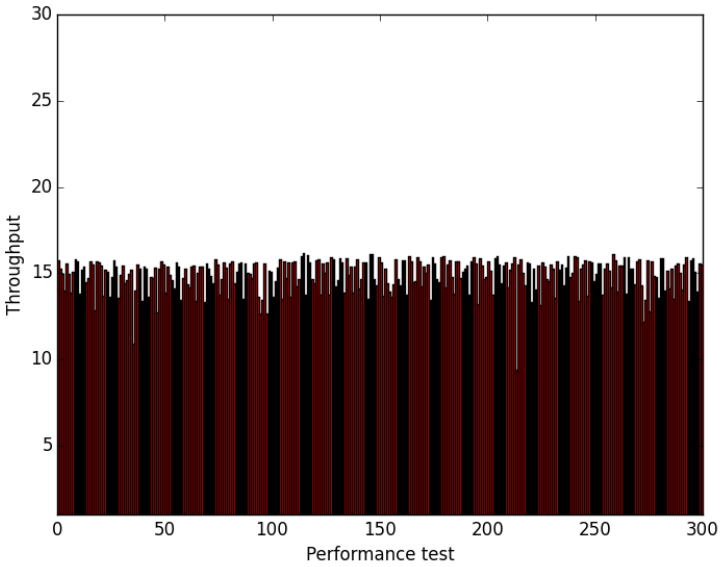
La passerelle domestique (*home gateway*) est le dispositif permettant de relier le réseau domestique à Internet. Elle comprend généralement un modem pour la conversion des signaux, un routeur, un commutateur et un point d'accès sans fil. La passerelle effectue des fonctionnalités de routage des paquets, de translation d'adresse et de pare-feu. En France, la plupart des FAIs offrent à leurs utilisateurs une passerelle domestique assez sophistiquée appelée box ADSL qui leur permet de bénéficier d'un bouquet de services annexes à l'accès à Internet comme la téléphonie et la télévision sur IP. Cet équipement est en charge de la gestion des données entrantes et sortantes du réseau domestique pour être acheminées vers les différents équipements dans le réseau.

Les solutions de la virtualisation de la passerelle du réseau domestique sont moins répandues. Cette technologie offre une flexibilité de gestion de la connexion Internet haut débit. La seule solution disponible à notre connaissance est la solution "Pantou" qui est développée par Yiakoumis [1]. Pantou est une implémentation de la version 1.0 de OpenFlow qui peut être utilisée avec une variété de routeurs qui supporte un système d'exploitation OpenWrt. OpenWrt est une distribution GNU/Linux à embarquer dans des routeurs [2]. Il s'agit d'un système d'exploitation très complet et facilement modifiable pour les routeurs. La solution Pantou permet de transformer la passerelle en un routeur OpenFlow. Nous avons évalué les performances de cette solution en termes de débit. La figure 3.3 illustre les débits maximaux de Pantou en réseau filaire et en réseau sans fil. Pour cette évaluation nous avons utilisé l'outil de mesure de la bande passante, `Iperf`, afin de déterminer le débit maximal supporté par Pantou. Nous avons remarqué que le débit maximal supporté en filière est de 15 Mbps et en sans fil est de 13 Mbps. Cette solution présente de médiocres performances. Selon l'étude Visual Networking Index de Cisco [3] les connexions Internet haut débit des réseaux domestiques en Europe ont un débit moyen de 19 Mbps. À l'aube de 2018, le débit moyen d'une connexion Internet haut débit passera à 49 Mbps. En conclusion, la solution Pantou n'est pas adaptée aux besoins en termes de bande passante pour les réseaux domestiques.
Pour remédier aux problèmes de limitation de Pantou, nous avons intégré un module noyau du *switch* Open vSwitch (OVS) [4] avec un system d'exploitation OpenWrt. Open vSwitch est une implémentation logicielle d'un commutateur Ethernet qui permet la création d'un commutateur virtuel sur la passerelle domestique. Il s'agit d'un commutateur logiciel open-source conçu pour être utilisé comme un commutateur virtuel ayant les mêmes fonctionnalités qu'un vrai commutateur administrable. Il permet de supporter OpenFlow 1.0. La figure 3.4 illustre les performances de OVS en termes de débit maximal de téléchargement supporté. Il permet également d'avoir un débit maximal de 80 Mbps en sans fil et de 93 Mbps en filière.

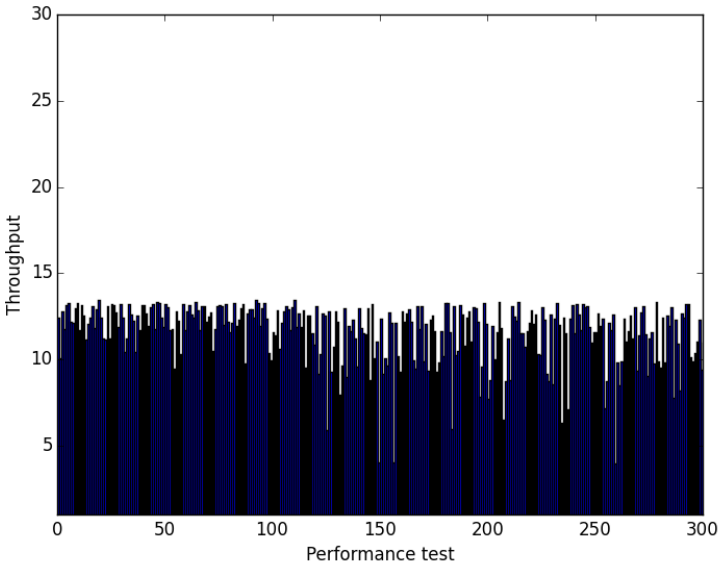
4 SDN et réseaux domestiques connectés à Internet haut débit

La flexibilité introduite par SDN rend la gestion de la QoS plus facile. Auparavant, la plupart des solutions classiques sont généralement proposées par des constructeurs d'équipements réseaux comme Cisco [5] ou Blue Coat [6]. D'autres travaux ont été proposés pour effectuer l'identification de l'application et la classification du trafic IP afin d'offrir un traitement spécifique pour chaque flux [Roughan et al., 2004, Karagiannis et al., 2005]. De plus en plus d'environnements d'expérimentation ont commencé à fournir la possibilité de gérer la QoS avec SDN [Sonkoly et al., 2012]. Dans cette partie nous allons présenter l'état de l'art des mécanismes de QoS proposées avec SDN.

1. http://archive.openflow.org/wk/index.php/Pantou_:_OpenFlow_1.0_for_OpenWRT
2. http://wiki.openwrt.org/fr/start
3. http://www.cisco.com/visual-networking-index-vni
4. http://www.openvswitch.org
5. http://meraki.cisco.com
6. http://www.bluecoat.com/products/packetshaper



(a) Débit maximal en filière

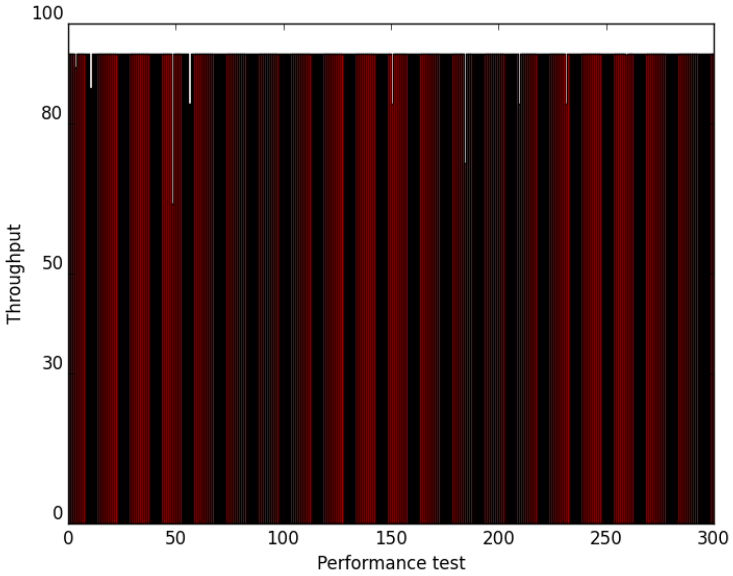


(b) Débit maximal en sans fil

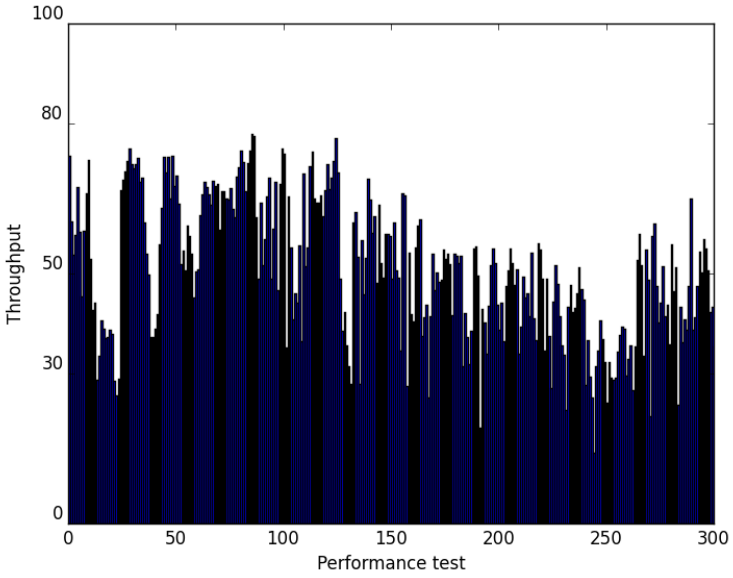
FIGURE 3.3 – Les performances de Pantou en filaire et en sans fil.

4.1 Gestion de la QoS dans les réseaux SDN

Kim et al. [Kim et al., 2010] ont présenté un cadre de contrôle automatisé de la qualité de service du réseau OpenFlow. Il s'agit d'une extension de OpenFlow qui permet la gestion de la QoS au niveau du contrôleur. L'idée des auteurs est de placer des limiteurs de flux au niveau du routeur d'accès. Ils ont aussi proposé un cadre de contrôle à granularité fine de la qualité de service pour automatiser la gestion de multiples commutateurs dans les réseaux OpenFlow.



(a) Débit maximal en filaire



(b) Débit maximal en sans fil

FIGURE 3.4 – Le débit maximal de OVS en filaire et en sans fil

Quand un nouveau flux arrive au niveau du commutateur OpenFlow, les premiers paquets de ce flux sont envoyés au contrôleur. Ce dernier détermine, d'abord, si oui ou non le flux est sensible à la QoS, ensuite, il s'assure du rôle des commutateurs d'accès à satisfaire les exigences et à garantir les performances de ce flux. Ishimori et al. [Ishimori et al., 2013] ont développé QoS-Flow, un système qui permet de fournir la QoS dans les réseaux OpenFlow. L'idée est d'utiliser plusieurs stratégies d'ordonnancement des paquets au niveau d'un commutateur OpenFlow, au

lieu de la stratégie FIFO existante. L'objectif est d'offrir aux applications le contrôle de la bande passante dont elles ont besoin. Leur système n'offre aucune classification. Il présente seulement une solution sur la façon d'ajouter des fonctionnalités à OpenFlow pour garantir la QoS.
Ko et al. [Ko et al., 2013] ont proposé une nouvelle architecture SDN, qui est une variante évolutive et flexible de OpenFlow, pour la gestion de la QoS dans les réseaux. Les auteurs ont aussi proposé un cadre à deux niveaux de gestion de QoS pour chaque flux réseau. L'idée est de diviser la table de routage en trois ; la première pour l'état du flux, la seconde pour les règles de transmission (*forwarding rule*) et la troisième est une table de gestion de QoS où plusieurs flux peuvent partager un même profil de QoS. La solution proposée exige une architecture microprocesseur multi-cœur qui n'est pas conçue pour la passerelle des réseaux domestiques.
Ferguson et al. [Ferguson et al., 2013] ont développé PANE, un système qui permet à un utilisateur de réserver de la bande passante avec une QoS garantie entre deux hôtes. PANE est proposé dans le but de trouver des solutions aux problèmes de configuration des réseaux dont on peut citer le contrôle d'accès à la configuration du lien physique. Il ne se concentre pas spécialement sur la qualité de service pour les réseaux Internet haut débit. Williams et Zander [Williams and Zander, 2011] ont développé un algorithme de classification du trafic IP basé sur les propriétés des flux. Ils ont évalué ses performances sur une passerelle domestique. Leur approche supporte un débit maximal de 28 Mbps. À l'aube de 2018, le débit moyen d'une connexion Internet pour un réseau domestique sera 49 Mbps. Ceci rend la solution de Williams et Zander non adaptée aux besoins des utilisateurs des réseaux domestiques connectés à Internet haut débit.
Risso et Cerrato [Risso and Cerrato, 2012] ont développé un mécanisme fondé sur OpenFlow pour la personnalisation du traitement des paquets au niveau de la passerelle domestique. Leur solution offre la possibilité d'installer des applications au niveau du plan de données de la passerelle du réseau domestique pour inspecter et modifier un trafic en transit. Ces applications sont utilisées par les fournisseurs de services ou les utilisateurs finaux pour contrôler le trafic réseau au niveau de cette passerelle domestique. L'architecture proposée est axée sur la modification du traitement des paquets et non sur la gestion de la QoS.

4.2 Gestion de la QoS dans les réseaux domestiques

Nombreuses autres approches ont été explorées pour la gestion de la QoS dans les réseaux domestiques. Yiakoumis et al. [Yiakoumis et al., 2012] ont proposé une solution qui permet aux utilisateurs d'aviser le fournisseur de services Internet à propos de leurs besoins en bande passante pour chaque service. L'approvisionnement se produit au niveau de la boucle locale située au dernier kilomètre du câble téléphonique raccordant les domiciles des abonnés. Cette solution implique un déploiement au niveau de l'infrastructure de fournisseur d'accès Internet rendant ainsi la tâche difficile à l'utilisateur pour sa mise en œuvre chez lui.
Georgopoulos et al. [Georgopoulos et al., 2013] ont proposé un *Framework* basé sur OpenFlow qui sert à améliorer la qualité de l'expérience des utilisateurs (QoE) dans les réseaux domestiques pour les flux multimédia. Le but recherché est la garantie de l'équité et de l'efficacité lors du partage de la bande passante. Le système identifie les applications par le bien du port de destination des flux de paquets. Cette solution n'étant pas efficace puisque, généralement,

les applications utilisent d'autres ports dynamiques que ceux affectés par défaut. Mortier et al. [Mortier et al., 2011] ont développé une solution nommée Homework. Il s'agit d'une plate-forme pour les réseaux domestiques qui offre la mesure et la gestion des capacités par flux. La solution permet aux utilisateurs de surveiller et de contrôler l'usage de la bande passante par périphérique et par protocole. Cependant, il ne fournit pas un soutien de qualité de service ou de classification de l'application.

5 FlowQoS : Gestion de la QoS dans un environnement domestique virtualisé

Une approche pour le déploiement de la QoS dans les réseaux domestiques consiste à déléguer les fonctions qui gèrent la QoS, l'identification de l'application et la configuration au niveau du routeur à une entité logique séparée. Ceci permet à un utilisateur de configurer des politiques de qualité de service à des niveaux d'abstraction plus élevés (par exemple par application). L'émergence de SDN et en particulier OpenFlow spécification [McKeown et al., 2008] rend cela possible. Un contrôleur pourrait exécuter directement un programme sur le routeur (dans le cas où le routeur est assez puissant comme la Freebox), ou sur un dispositif séparé, soit à l'intérieur de la maison soit à un emplacement distant. D'autres travaux antérieurs ont indiqué que les routeurs domestiques sont de plus en plus puissants [Valancius et al., 2009, Whiteaker et al., 2012], et peuvent finalement permettre d'embarquer les fonctionnalités de notre approche dans ces routeurs.
Nous présentons FlowQoS, comme un système pour effectuer la gestion de la QoS par flux basée sur des applications en déléguant à un contrôleur SDN les fonctions de la classification des flux et la limitation du trafic. Dans FlowQoS, l'utilisateur du réseau d'Internet spécifie simplement la portion maximale de la bande passante pour chaque application. Le contrôleur effectue l'identification de l'application appropriée et la configuration de la QoS pour le trafic en amont selon les préférences de l'utilisateur. FlowQoS identifie, en temps réel, le type d'application pour chaque flux et installe des règles dans le plan de données en fonction des priorités définies par l'utilisateur pour ces applications. FlowQoS facilite pour un utilisateur la configuration des priorités et l'implémentation de QoS par flux. Parfois, la classification du trafic se produit à un emplacement distant, ce qui augmente la latence. Pour atténuer l'impact de ce problème, nous proposons une approche de classification basée sur les requêtes DNS. Cet algorithme de classification permet à FlowQoS d'effectuer une identification précoce du type d'application pour chaque flux. D'un autre côté, FlowQoS permet le transfert des premiers paquets de chaque flux avec une QoS par défaut avant d'installer les règles appropriées. Cette approche qui a été également appliquée avant dans les réseaux backbone IP [Newman et al., 1996] permet des latences très faibles. À l'heure actuelle, les routeurs restent encore limités en puissance et c'est pour cette raison que le prototype de FlowQoS est installé actuellement sur un dispositif distinct.

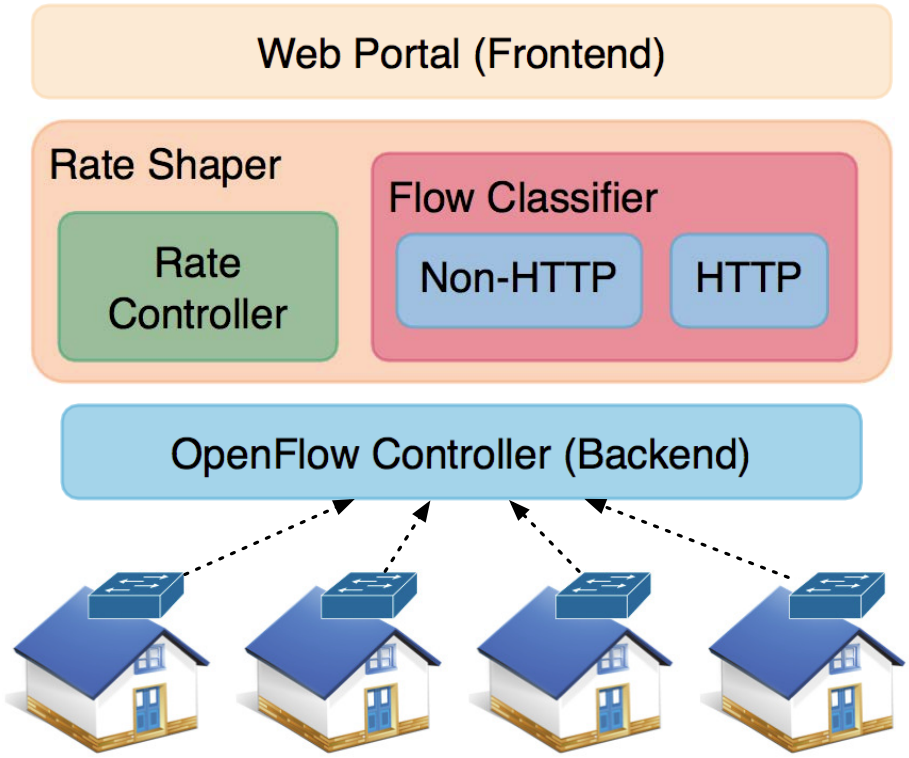


FIGURE 3.5 – Architecture de FlowQoS

5.1 Architecture de FlowQoS

La figure 3.5 illustre l'architecture de FlowQoS. Le contrôleur est déployé à l'extérieur de la maison selon le prototype mis en œuvre. Chaque routeur peut faire fonctionner son propre contrôleur. Les utilisateurs configurent le maximum de la bande passante allouée pour chaque application à l'aide d'un portail Web. Un fichier de configuration est alors généré. Le Shaper utilise ce fichier pour mettre en œuvre la limitation de la bande passante associée à chaque application. Les deux principales composantes de notre système sont le limiteur de bande passante basé sur SDN et le classificateur de flux. Le classificateur de flux est composé de deux modules : le premier effectue une identification anticipée du trafic des applications HTTP et HTTPS en utilisant les informations DNS. Le second module effectue l'identification de l'application pour les autres flux.

Lorsque le premier paquet d'un flux arrive au niveau de la passerelle domestique, une copie de ce paquet est transmise au contrôleur. Le *switch* OpenFlow continue à effectuer le renvoi du trafic dans le chemin par défaut jusqu'à ce que l'identification de l'application soit réalisée. Le contrôleur détermine quel classificateur classe le flux réseau selon le type d'application. Pour les ports HTTP et HTTPS, le classificateur basé sur DNS effectue une classification fine selon les nombreuses applications s'exécutant sur ces ports en utilisant les réponses DNS, que le *switch* envoie au contrôleur. Pour les autres flux, on utilise une version modifiée de la libraire "Libpro-

toident" pour effectuer la classification. Ce module a besoin de cinq champs et d'informations supplémentaires (les quatre premiers octets du contenu du premier paquet envoyé et du premier paquet reçu, la taille de ces contenus dans les deux sens de la communication et le protocole de transport utilisé). Il classifie le trafic en utilisant un ensemble de classificateurs. Chaque classificateur effectue une brève inspection pour rendre sa décision. Sur la base des résultats de classification, le contrôleur installe les règles de routage pour chaque flux au niveau du chemin approprié et par conséquent, la file d'attente dans le commutateur.

5.2 Le classificateur de flux

FlowQoS utilise deux modules différents pour classifier le trafic. Il identifie le trafic des applications s'exécutant sur les ports 80 et 443 (HTTP et HTTPS) en utilisant le classificateur de DNS. Ce dernier effectue une plus fine classification. De nombreux classificateurs identifient le trafic de ces applications comme un trafic web. Pour les autres applications non-HTTP le classificateur utilise une version modifiée de la libraire "Libprotoident" [Alcock and Nelson, 2012].

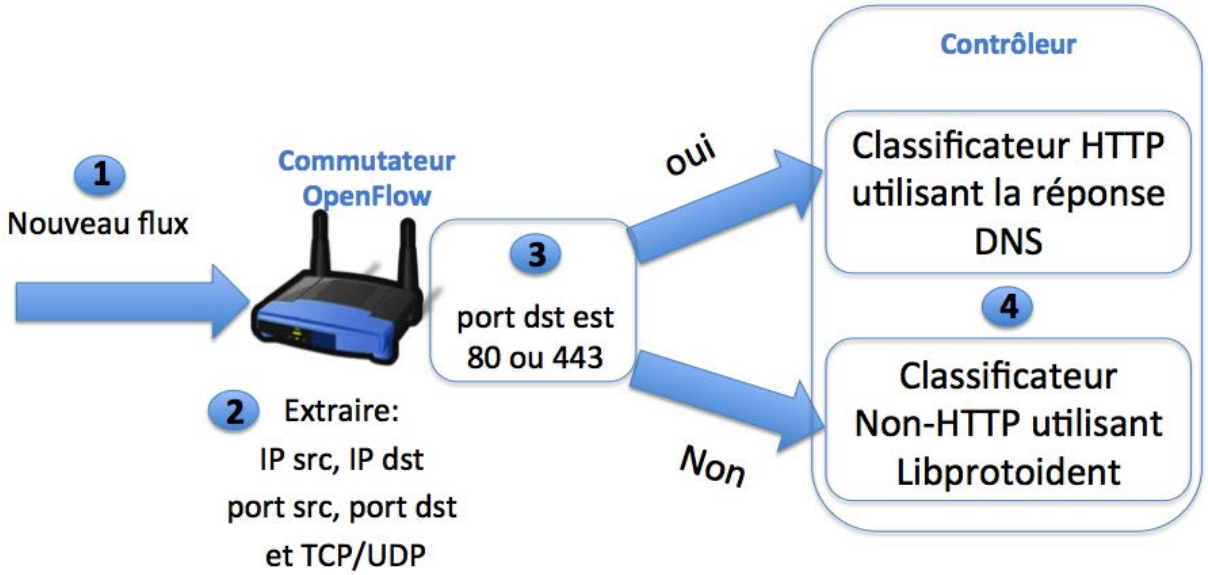


FIGURE 3.6 – La classification des paquets avec FlowQoS

La figure 3.6 illustre les différentes étapes de classification des paquets par FlowQoS. Quand un nouveau flux arrive au niveau de la passerelle, les cinq champs dans les premiers paquets permettent d'identifier le flux. Invoquant la libraire Libprotoident nécessite la connaissance des quatre premiers octets envoyés, des quatre premiers octets reçus, de la taille des données envoyées et de la taille des données reçues pour des premiers paquets dans les deux sens de la transmission. Lorsque le contrôleur commence l'analyse de ces champs, les paquets du flux sont transmis dans une file d'attente par défaut.

Le classificateur DNS maintient une table qu'il construit à l'aide des réponses DNS. La table

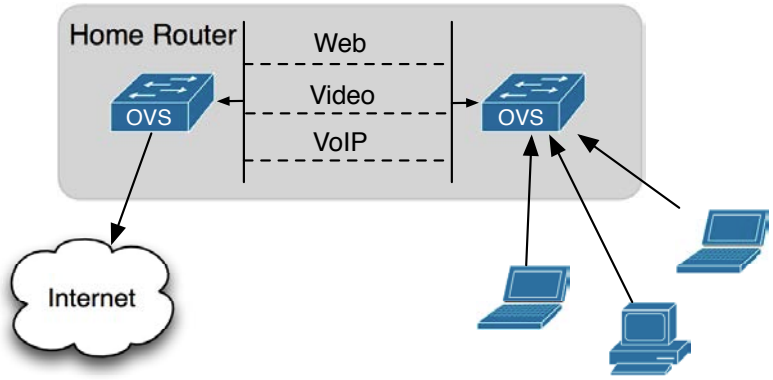


FIGURE 3.7 – Topologie des commutateurs virtuels effectuant le lissage du trafic à l'intérieur du routeur domestique

comprend les champs ANAME ou CNAME, l'adresse IP correspondante et la durée de vie de l'enregistrement. Pour classer les flux sur la base de cette information, le classificateur vérifie le ANAME ou CNAME avec une liste d'expressions régulières. À titre d'exemple l'expression régulière "`*. amazonaws.com`" permet d'identifier les flux vidéo de Netflix. Puisque l'utilisateur lance une requête DNS avant la connexion TCP correspondante, le classificateur associe un flux à une application avant même le début du flux. Dans le cas des flux HTTP ou HTTPS, le contrôleur installe de façon proactive une règle qui décrit la façon de traitement des paquets basée sur l'adresse IP de la source et l'adresse IP de destination et les ports associés. Cette règle mentionne que pour un flux donné, l'action consiste à le transférer vers un port du *switch* virtuel. Le classificateur par flux peut également établir des priorités dans les flux distincts de la même page Web. Par exemple, supposons qu'un utilisateur visionne un contenu sur `youtube.com`, la plupart des flux sont classés comme "web". Le contenu de "`*. Googleuservideo.com`" sera uniquement considéré comme "vidéo". Un utilisateur peut ajouter d'autres expressions régulières dans le classificateur, lorsqu'un fournisseur de vidéo à la demande utilise un nouveau CDN (*Content Delivery Network*).

Le classificateur de flux maintient une table de correspondance (*lookup table*) dans laquelle la clé est constituée de l'adresse IP source, l'adresse IP de destination, le protocole, le port source et le port de destination. Lorsqu'un nouveau flux est initié, la passerelle envoie une copie du premier paquet du flux au contrôleur. Le classificateur de flux vérifie ensuite si le champ correspond bien à une entrée dans la table de correspondance. Par la suite il retourne le type d'application, tel que vidéo, VoIP, P2P, ou Web.

5.3 Le lissage du trafic

Sur la base de l'identification de l'application par classificateur, FlowQoS crée des associations entre les flux et les applications, et contrôle le débit à base de SDN pour tout le trafic. Dans les systèmes existants, l'affectation d'une telle priorité à partir d'une configuration de l'utilisateur

est compliquée à cause des limitations des routeurs domestiques actuels. En effet, ces derniers ne supportent pas le contrôle par flux, ils offrent seulement des mécanismes sur "*Linux traffic control* (`tc`))" ou encore sur un marquage par *IPtables*. La version actuelle de Open vSwitch ne supporte pas encore les spécifications de OpenFlow 1.3 [Pfaff et al., 2012] afin de permettre la gestion de la QoS par flux.

Pour surmonter les limitations citées précédemment, FlowQoS permet la gestion de la QoS par flux à l'aide de l'instanciation d'une topologie virtuelle à deux *switchs* virtuels au niveau de la passerelle domestique comme le montre la figure 3.7. Chaque lien virtuel, entre deux *switch*, correspond à un groupe différent d'application (vidéo, web, jeux). Pour mettre en œuvre la limitation de débit, chaque lien a un régulateur de trafic spécifié par l'utilisateur (mis en œuvre avec l'utilitaire de Linux `tc`). Quand un nouveau flux arrive au niveau du *switch*, il est envoyé vers un lien approprié défini par le classificateur. Sur la base des résultats de classification, le contrôleur installe les règles OpenFlow dans les composants Open vSwitch. La configuration de `tc` sur des liens virtuels *inter-switch* est similaire à celle de OpenFlow 1.3 [7]. Open vSwitch peut supporter les fonctions de QoS pour chaque flux de données comme décrit dans OpenFlow 1.3. Notre topologie est une solution permettant de contourner les limitations dans la mise en œuvre de Open vSwitch.

Pour limiter le débit, une fois le classificateur a identifié le type de flux, le limiteur de trafic se réfère à des règles existantes déterminant la connexion *inter-switch* correspondant à celle de la classe du trafic. Chaque type de flux sera limité d'une manière différente définie par l'utilisateur. Le contrôleur de débit gère également l'installation d'autres règles, telles que les règles associées aux réponses DNS. Une copie des paquets initiaux pour chaque nouveau flux est transmise au contrôleur. Le *switch* à son tour continue d'acheminer le flux par le lien par défaut jusqu'à ce que la classification de l'application soit effectuée. Cette opération s'arrête lorsqu'une règle de transfert des paquets au lien approprié est créée.

5.4 Evaluation de FlowQoS

Pour la mise en œuvre du prototype de FlowQoS, nous utilisons un routeur basé sur OpenWrt. Nous avons, d'abord, intégré Open vSwitch avec OpenWrt pour permettre le contrôle d'un *switch* utilisant OpenFlow. Nous avons ensuite utilisé une Raspberry Pi [Upton and Halfacree, 2012] comme équipement du contrôleur. Enfin nous avons implémenté FlowQoS au-dessus de Pox [8] considéré comme un contrôleur OpenFlow populaire open source.

La figure 3.8 montre le montage expérimental. Nous avons configuré la connexion Internet à 12 Mbps en *download* et 6 Mbps en *upload*. Nous avons alloué une bande passante de 7 Mbps pour la vidéo, 3 Mbps pour la VoIP et 2 Mbps pour les applications web. Lors des expérimentations, la hôte 1 est soit en train de regarder une vidéo soit d'effectuer un appel VoIP, en fonction du scénario expérimental. La hôte 2 génère un trafic en téléchargeant un grand fichier. FlowQoS sépare les flux traversant la passerelle domestique et les envoi aux chemins correspondants entre les deux commutateurs OVS en installant la règle de correspondance adéquate.

7. https://www.opennetworking.org/openflow/openflow-spec-v1.3.0.pdf
8. http://www.noxrepo.org/pox/about-pox

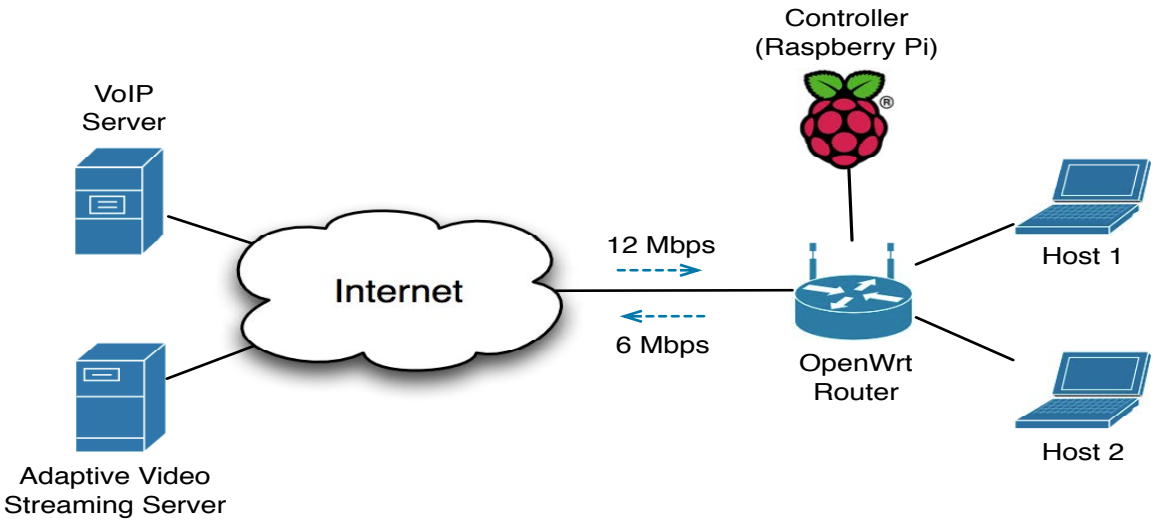


FIGURE 3.8 – Montage expérimental d'évaluation de FlowQoS

Nous avons évalué les capacités de FlowQoS pour la gestion de la QoS pour améliorer la performance de streaming vidéo adaptatif qui repose sur TCP. Ce streaming vidéo s'adapte à la bande passante disponible chez l'utilisateur. Nous évaluons FlowQoS pour le streaming vidéo adaptatif sur HTTP. Nous effectuons ces expériences en utilisant le Dataset DASH [Lederer et al., 2012]. Nous utilisons le benchmark vidéo Big Buck Bunny avec une longueur de segment vidéo de quatre secondes avec dix débits différents avec *Dash-JS video player*.

Les figures 3.9 et 3.10 illustrent les performances, en termes de débit d'encodage et de téléchargement de la vidéo avec et sans FlowQoS. La figure 3.11 représente l'évolution du trafic de téléchargement au cours du temps de la hôte 2. Les résultats obtenus montrent que FlowQoS permet au système de converger rapidement à un débit d'encodage plus élevé. Ainsi, on peut conclure que FlowQoS améliore la qualité du streaming vidéo adaptatif par réduction de l'oscillation entre les différents débits utilisant la totalité de la bande passante allouée pour les flux vidéo. Les oscillations observées entre différents débits d'encodage dans la figure 3.9 sont dues au comportement non agressif de l'algorithme de sélection du bitrate du lecteur de streaming vidéo [Miller et al., 2012]. Dès qu'il y a un changement lors de l'estimation de la bande passante disponible, l'algorithme choisit un nouveau chunk de la vidéo avec un différent débit d'encodage.

Nous avons également évalué FlowQoS dans le cadre de l'application VoIP. La voix sur IP est très sensible à la gigue et aux délais de transmission des paquets. Nous surveillons les délais des paquets et la gigue de l'application de VoIP utilisant les outils `ping` et `iperf` au cours de l'expérience. L'utilisation de ces deux outils permet de mesurer la latence et la gigue car nous ne disposons pas d'un serveur et d'un client dont les horloges sont parfaitement synchronisées. Nous avons constaté que même en utilisant le protocole NTP (*Network Time Protocol*) le temps de transit des paquets vers le serveur de synchronisation n'est pas comptabilisé. Ceci entraine une différence de quelques millisecondes entre les deux horloges, ce qui ne satisfait pas nos besoins d'expérimentation.

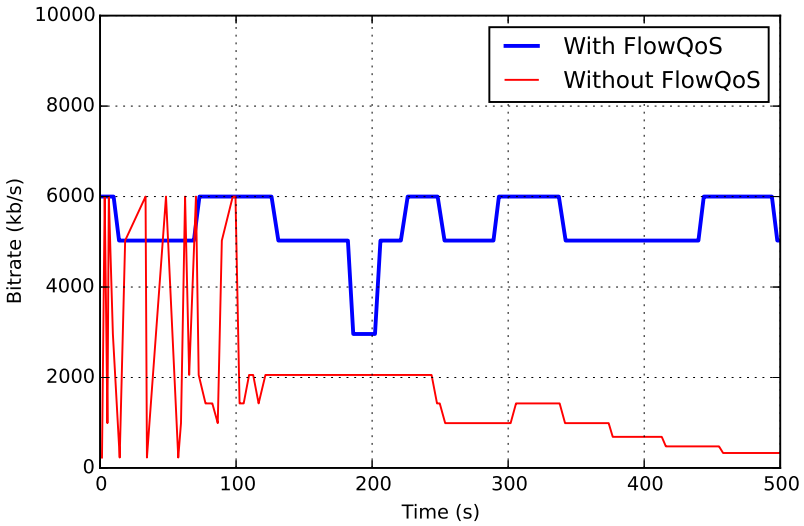


FIGURE 3.9 – Comparaison de la sélection du débit d'encodage du flux de streaming vidéo adaptatif au cours du temps avec et sans FlowQoS

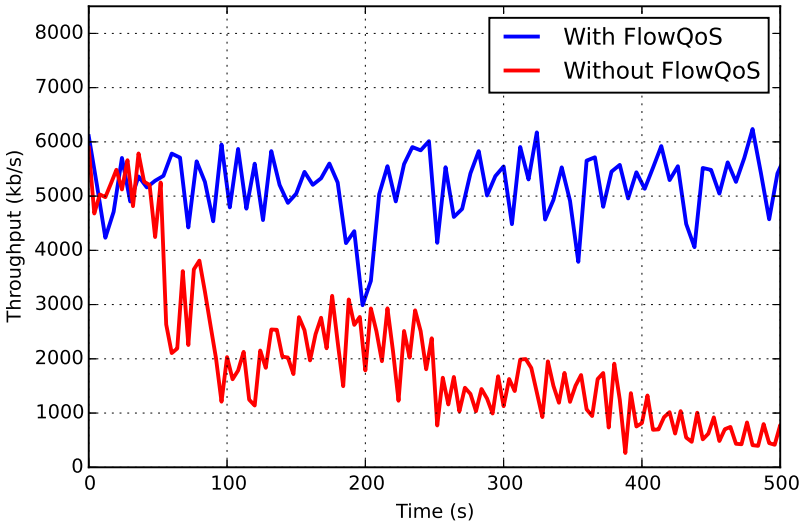


FIGURE 3.10 – Performance du débit de téléchargement streaming vidéo adaptatif avec et sans FlowQoS

Les figures 3.12 et 3.13 montrent l'évolution de la latence d'aller-retour et la gigue avec et sans FlowQoS pour une application VoIP de la hôte 1 en concurrence avec un flux TCP de téléchargement de fichier de la hôte 2 pendant 1000 secondes sans et avec FlowQoS. Il est important de noter que les grandes valeurs du délai (par exemple à l'instant t = 940s) sont dues aux délais de réponse aux paquets *ICMP Echo request* envoyés.
L'évaluation montre que FlowQoS garantit les délais et réduit la gigue pour ce type d'application. Notre système garantit les exigences de la latence et de la gigue pour les flux VoIP sans se soucier des autres flux gourmands en bande passante.

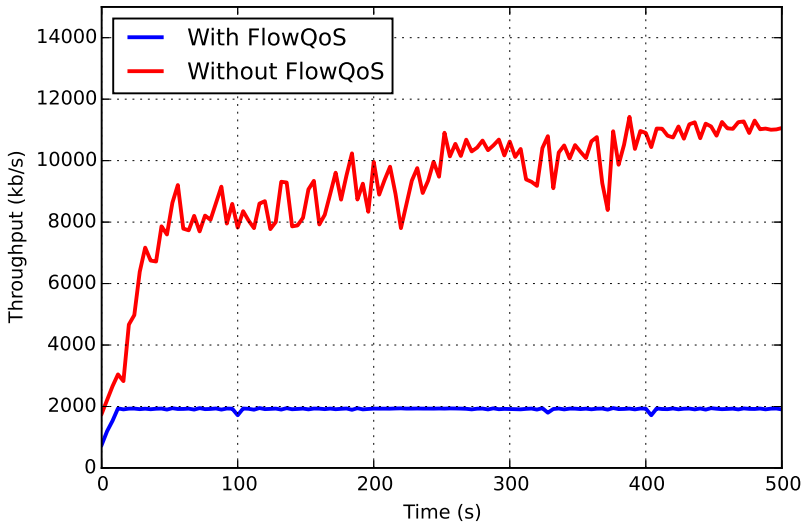
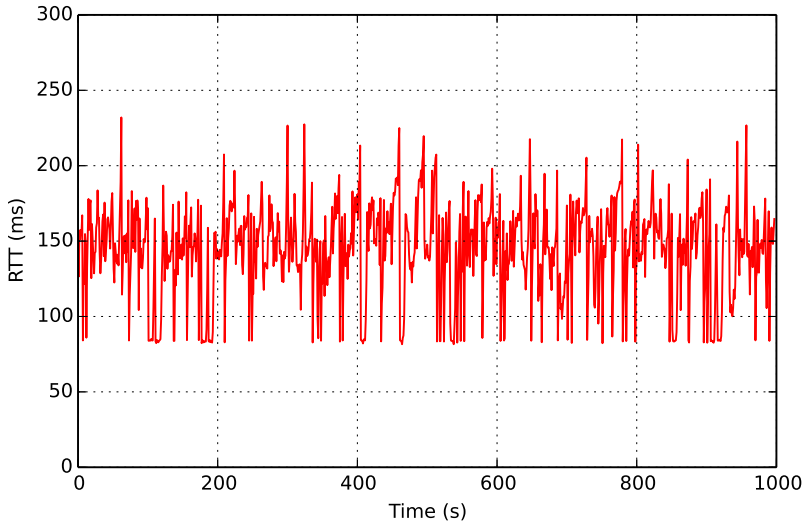


FIGURE 3.11 – L'évolution au cours du temps du débit de téléchargement des flux en concurrence pour la bande passante disponible avec et sans FlowQoS

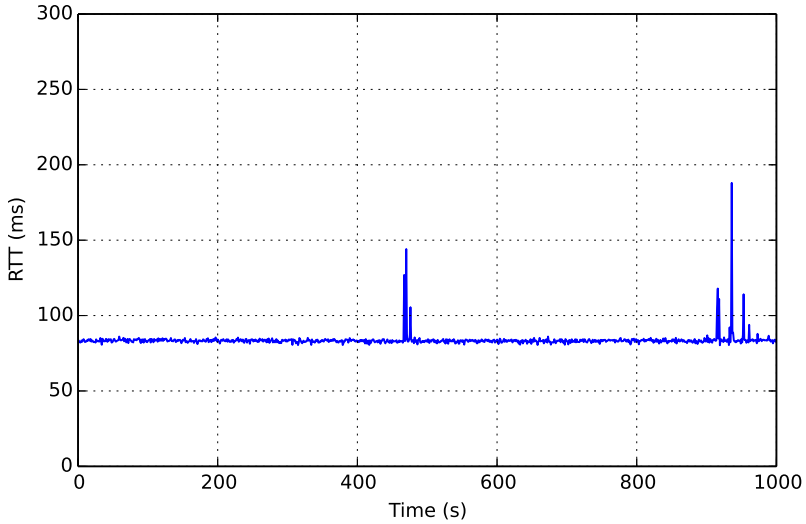
Nous avons évalué les délais moyens de classification des paquets pour les deux scénarios de déploiement de FlowQoS. La figure 3.14 illustre le temps que prend le classificateur de FlowQoS pour identifier le trafic quand il est exécuté sur un contrôleur local (Raspberry Pi) et sur un contrôleur distant (Serveur). Les délais définissent le temps de création d'une entrée pour chaque flux identifié. Le Raspberry Pi est doté d'un processeur ARM 700 MHz et 512 Mo de RAM. Le serveur déployé par le FAI à la boucle locale est doté d'un processeur Intel Core 2 à 2,4 GHz et 8 Go de RAM. Dans le cas du contrôleur distant, un délai supplémentaire doit être ajouté, il est égal à deux fois la latence à la boucle locale. Le délai supplémentaire varie de 20 à 80 ms près selon les mesures effectuées dans [Sundaresan et al., 2011]
Au cours de nos expériences, seule une proportion de 2,34% d'un ensemble de 20.000 de flux n'a pas été classée. Ces flux sont transmis vers le lien virtuel par défaut réservé aux applications Web.

Pour quantifier la surcharge de gestion d'un réseau virtuel à deux commutateurs sur un seul exemple d'Open vSwitch ou avec un script de lissage de trafic simple, nous avons comparé le taux d'utilisation du CPU et de la mémoire d'un OpenWrt non modifiée et une topologie de deux OVS reliés par un lien unique. Toutes les 30 secondes, l'utilisateur demande la vidéo à un autre serveur avec une adresse IP différente. Nous comparons les résultats lors de la présence d'un utilisateur de streaming YouTube vidéos. La table 3.1 montre que la topologie des deux OVS n'ajoute pas de surcharge au CPU et à la mémoire par rapport à une installation de base de OpenWrt.

Nous avons aussi comparé l'utilisation moyenne du CPU et de la mémoire de FlowQoS et un script de lissage du trafic (*traffic shaping script*), en présence du même streaming vidéo de YouTube. Ce script effectue la limitation de bande passante par flux qui a pour but de contrôler le débit de téléchargement des paquets. Il effectue cette opération en créant une discipline de mise



(a) La latence aller-retour sans FlowQoS



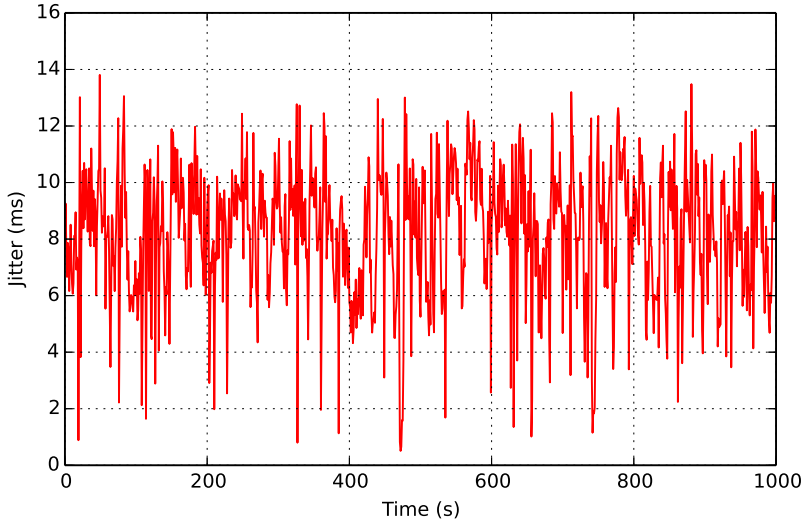
(b) La latence aller-retour avec FlowQoS

FIGURE 3.12 – Les performances en termes de délais de la VoIP avec et sans FlowQoS

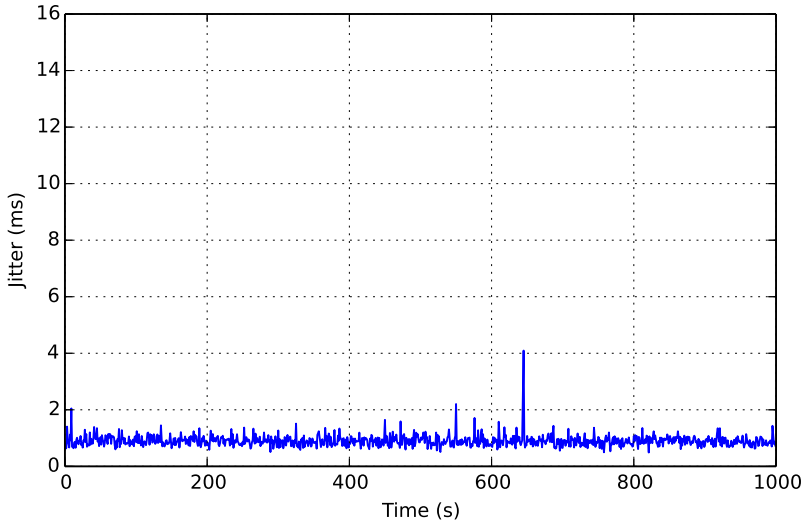
	Utilisation du CPU	Utilisation de la mémoire
OpenWrt non modifié	11.34%	33.62%
Topologie des deux OVSs	16.78%	41.23%

TABLE 3.1 – Comparaison de l'utilisation du CPU et de la mémoire de OpenWrt sans et avec la topologie des deux OVSs

en file d'attente dédiée aux flux de trafic entrant et sortant. Cette solution est utilisée pour lisser le trafic par flux de données. Ce trafic est identifié par l'adresse IP source, l'adresse IP destination, le port source et le port destination. Nous limitons le débit du trafic de téléchargement de chaque flux à 7 Mbps et 3 Mbps pour l'upload. Ensuite, nous surveillons la mémoire et l'utilisation du



(a) La gigue sans FlowQoS



(b) La gigue avec FlowQoS

FIGURE 3.13 – Les performances en termes de gigue de la VoIP avec et sans FlowQoS

	Utilisation du CPU	Utilisation de la mémoire
Script de lissage du trafic	25.39%	49.21%
FlowQoS	23.63%	44.68%

TABLE 3.2 – Comparaison de l'utilisation du CPU et de la mémoire de FlowQoS et d'un script de lissage du trafic

processeur et nous calculons la moyenne durant dix minutes.
Aujourd'hui, les scripts de gestion de QoS existant dans les routeurs domestiques tels que le script de lissage du trafic utilisé, sont complètement inutiles puisque la plupart du trafic est crypté ou envoyé sur HTTP.

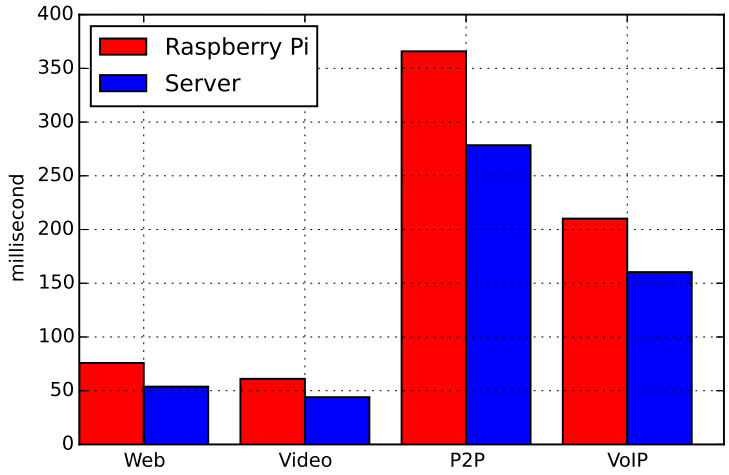


FIGURE 3.14 – Comparaison des délais de classification des paquets avec un contrôleur local (Raspberry PI) et un contrôleur distant (un serveur déployé à la boucle locale)

La table 3.2 montre les résultats de l'évaluation expérimentale effectuée. La consommation de mémoire avec FlowQoS est inférieure à celle avec un script de mise en forme du trafic. En effet, le module Open vSwitch que nous utilisons pour OpenWrt est un module noyau *Kernel module*. Il permet d'effectuer le lissage du trafic par flux sans introduire une surutilisation des ressources de la passerelle. De même la consommation du processeur est inférieure à celle du script de QoS. La limitation du flux est faite une seule fois dans FlowQoS. Quant au script de QoS, il crée une nouvelle file d'attente au début de chaque nouveau flux.

6 VidShaper : Allocation dynamique de la bande passante pour les flux de streaming vidéo adaptatif

L'une de limitation de FlowQoS réside dans le fait qu'il n'assure pas la QoS à chaque flux vidéo lorsque plusieurs lecteurs de streaming adaptatif sont actifs. Dans cette partie nous présentons une extension qui sera à inclure dans FlowQoS pour palier ce défaut. De nombreuses études récentes ont montré que Youtube et Netflix sont responsables de la génération d'environ 55% du trafic aux Etats-Unis [ULC, 2012]. Le streaming vidéo adaptatif implique un échange entre un serveur de streaming et un lecteur vidéo au niveau de l'utilisateur via le protocole HTTP. Le serveur fournit aux clients une table d'adresses URL. Chaque adresse URL est celle d'un segment vidéo généralement de 2 à 4 sec (*vidéo chunk*) et d'une qualité ou débit d'encodage (*bitrate*). Ces fichiers sont envoyés comme Metadata aux clients (*Manifest file*) généralement sous forme cryptée. Côté client, après avoir reçu le fichier de Metadata, il analyse la bande passante disponible et sélectionne l'URL appropriée pour le prochain segment.

Aujourd'hui, dans un foyer, il est très commun d'avoir une télévision haute définition, des tablettes, des ordinateurs portables et des téléphones intelligents. Il arrive souvent que deux

lecteurs de streaming vidéo adaptatif se partagent la même bande passante. Offrir une bonne qualité d'expérience (QoE) pour les utilisateurs est souvent l'objet d'importants défis. Le but de notre travail est de mieux gérer le réseau afin d'éviter la dégradation des performances des applications multimédia quand la bande passante disponible devient un goulot d'étranglement. Plusieurs études [Akhshabi et al., 2012] ont montré l'apparition de problèmes de performance lorsque plusieurs lecteurs sont en concurrence lors du partage de la bande passante. Ces investigations ont identifié trois problèmes majeurs. Le premier concerne l'instabilité des lecteurs en termes de demande de débit d'encodage de la vidéo du serveur CDN. Le second problème est lié à la non-équité en termes de partage de bande passante. Le dernier est la sous-utilisation de la bande passante disponible. La plupart des solutions proposées impliquent une modification de l'infrastructure de serveur CDN [Jiang et al., 2012], ou une modification dans l'algorithme de sélection du débit [Akhshabi et al., 2012]. Le déploiement de ces contributions est une tâche ardue pour de nombreuses raisons. D'abord, le streaming vidéo adaptatif doit obéir à une norme dont le fonctionnement est de plus en plus difficile à changer. La deuxième difficulté réside dans le fait que beaucoup de fournisseurs de vidéos comme Youtube, ne sont pas favorables à un changement des architectures et des fonctionnalités des CDNs.

6.1 Architecture de VidShaper

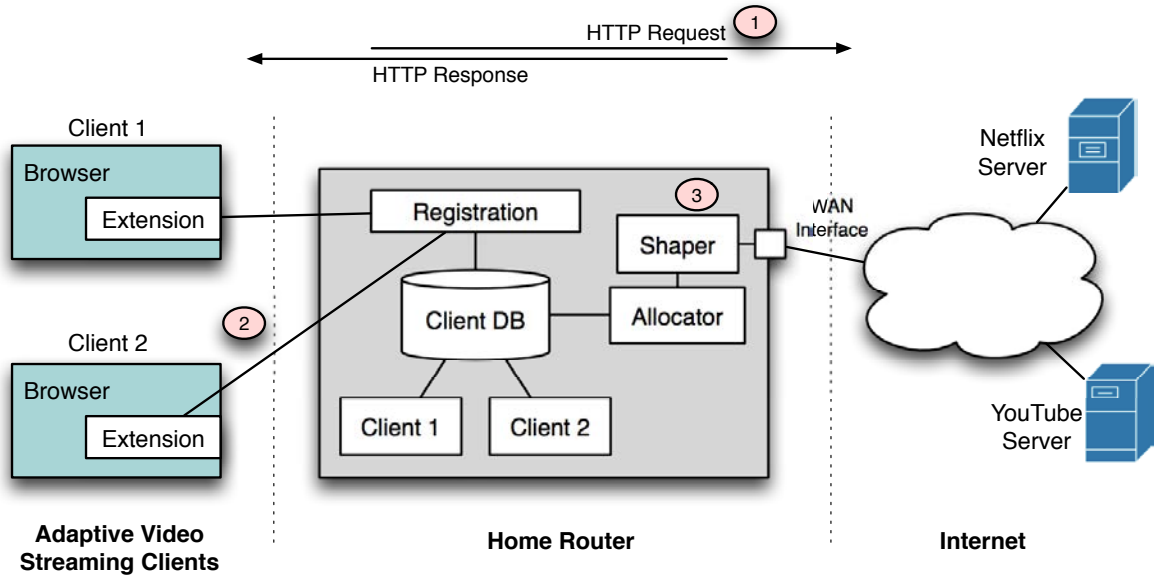


FIGURE 3.15 – Architecture de VidShaper pour effectuer la limitation dynamique du trafic entre plusieurs lecteurs de streaming vidéo adaptatif

Dans cette partie, nous proposons une technique appelée VidShaper basée sur le lissage du trafic que les utilisateurs peuvent facilement déployer dans leur foyer domestique afin d'assurer une meilleure QoE quand plusieurs lecteurs de streaming vidéo adaptatif sont actifs. Nous appelons cette solution VidShaper. Nous développons une extension au niveau du navigateur

Internet du client pour accéder au fichier Metadata, le traiter et extraire la liste des adresses IP des serveurs ainsi que les débits disponibles pour une vidéo. Notre extension exécute aussi un code au niveau client permettant d'avoir une information sur la résolution de l'écran. Le navigateur par la suite envoie toutes ces informations à la passerelle au niveau du réseau domestique. Chaque lecteur de streaming vidéo adaptatif renvoie des informations relatives à la résolution de l'écran de l'équipement qui fait du streaming au composant de registration dans la passerelle domestique. Ce dernier surveille le nombre de lecteurs actifs dans les réseaux. Il maintient une base de données avec des informations de chaque client actif dans le réseau.
Le composant d'allocation de la bande passante « Allocator » invoque le module de lissage du trafic « Shaper » pour limiter le flux de chaque lecteur. Il est en charge d'isoler chaque flux dans une file d'attente unique pour chaque lecteur. Ce composant utilise un algorithme de partage qui calcule la bande passante offerte pour chaque terminal de streaming vidéo. L'algorithme privilégie ceux qui ont une grande résolution. À titre d'exemple une télévision de grande taille aura toujours plus de bande passante qu'une tablette.
En effet, chaque terminal de streaming vidéo i possède une résolution de l'écran définie comme suit :

$$Resolution_i = (w_i, h_i), \tag{3.1}$$

où w est la largeur de l'écran et h est la hauteur de l'écran. Ces deux paramètres sont exprimés en pixel.

On définit ϕ_i le facteur d'allocation de ce terminal de streaming. Ce paramètre est donné par l'équation suivante :

$$\phi_i = \frac{1}{2}\left(\frac{w_i}{\sum_{k=0}^{n} w_k} + \frac{h_i}{\sum_{k=0}^{n} h_k}\right), \tag{3.2}$$

où n est le nombre des équipements de streaming vidéo actifs dans le réseau.
Chaque équipement reçoit une bande passante qui dépend du facteur d'allocation et de la bande passante disponible dans le réseau domestique donnée par l'équation suivante :

$$B^i_{all} = \phi_i * B_{dispo}. \tag{3.3}$$

6.2 Evaluation de VidShaper

Cette partie est dédiée à l'évaluation des performances de VidShaper pour chaque lecteur vidéo. Pour cela nous utilisons le dataset DASH. La figure 3.16 présente les parties qui composent l'expérience. Deux lecteurs de streaming vidéo adaptatif s'exécutent sur deux clients différents. Le premier a une résolution de 2560*1600 et le second 1280*800. La bande passante de la connexion Internet est de 6 Mbps en téléchargement et 3 Mbps en chargement.
Nous supposons que tout le long de cette évaluation les flux relatifs au streaming vidéo adaptatif sont actifs dans le réseau. Dans notre expérimentation on a $\phi_1 = 0,67$ et $\phi_2 = 0,33$. Par conséquent, le terminal 1 recevra une bande passante de 4 Mbps et le terminal 2 une bande passante de 2 Mbps. Nous évaluons les performances de chaque lecteur de vidéo streaming sans et avec VidShaper.

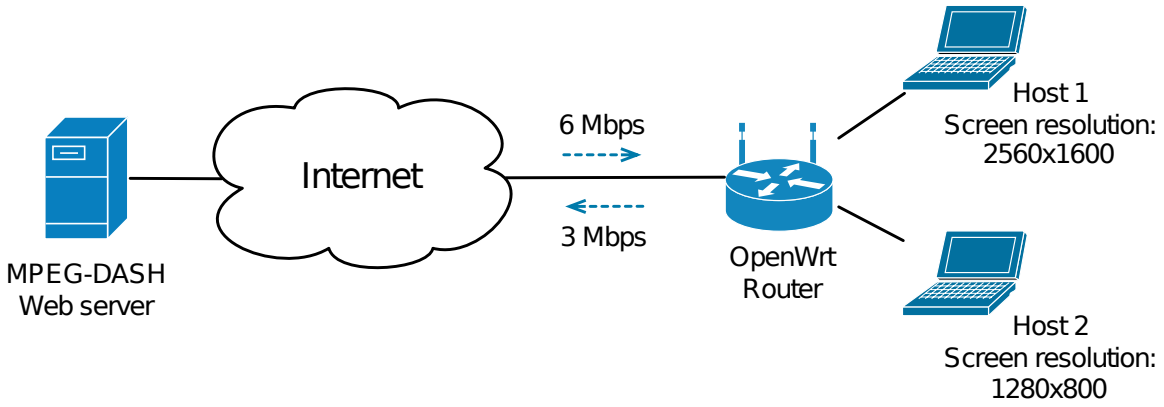
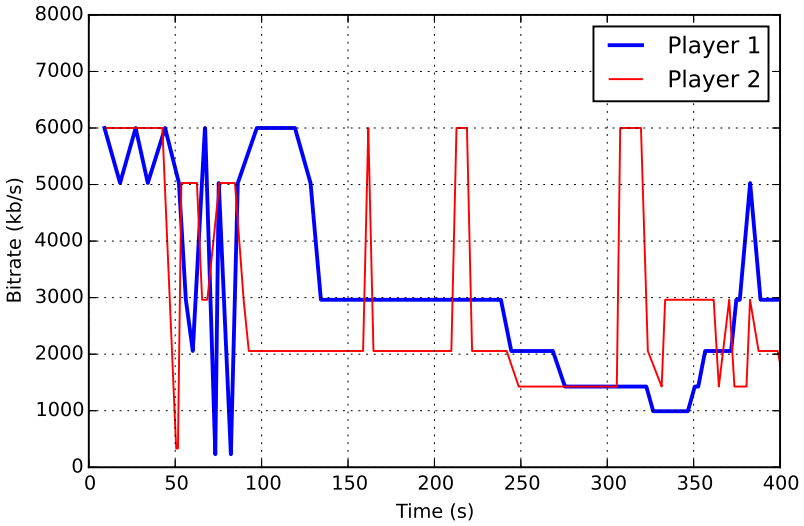


FIGURE 3.16 – Le montage expérimental de l'évaluation de VidShaper

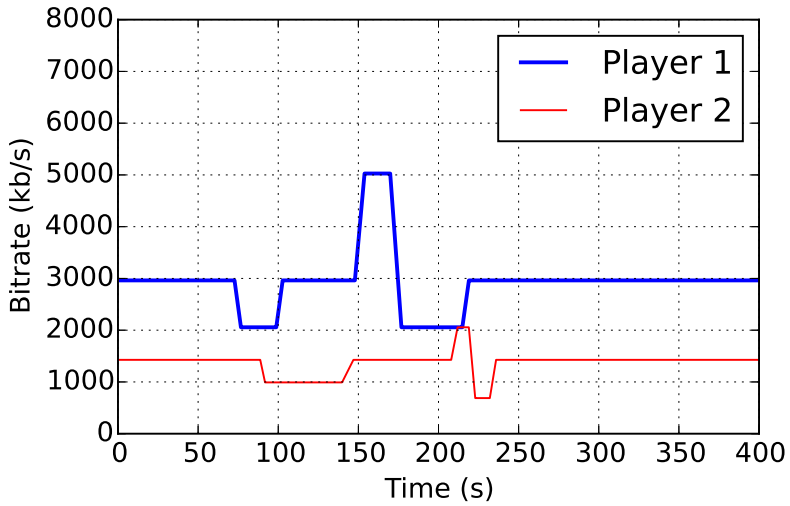
La figure 3.17 illustre les performances en termes de débit d'encodage avec et sans notre approche. On observe clairement que sans notre approche le lecteur de streaming vidéo souffre d'instabilité. En effet, en se basant sur une estimation de la bande passante disponible, le lecteur choisit le débit d'encodage adéquat. Lorsqu'un lecteur de streaming vidéo partage la bande passante avec un autre, l'algorithme de sélection de débit d'encodage est incapable de déterminer la valeur exacte de la bande passante offerte. Ceci implique un changement périodique dans la sélection du débit d'encodage adéquat qui est à l'origine des oscillations observées dans la figure 3.17.
La figure 3.18 montre l'évolution des débits de téléchargement au cours du temps des flux vidéo avec et sans notre approche. On observe que sans notre approche, il y a partage non équitable de la bande passante entre les deux lecteurs de streaming vidéo. On peut observer également que la somme des débits de téléchargement est souvent très inférieure à 6 Mbps ce qui engendre une sous-utilisation de la bande passante disponible. VidShaper permet de palier aux problèmes observés. Il permet de garantir un débit de téléchargement à chaque lecteur qui dépend de la résolution de l'équipement du streaming vidéo.

7 Conclusion

Dans ce chapitre nous avons présenté FlowQoS, une solution qui permet de gérer la QoS dans les réseaux d'accès à large bande en déléguant les fonctions de classification de trafic à un contrôleur SDN. Ce dernier installe les règles de routage dans le routeur pour chaque flux de données. Cette approche simplifie la configuration de la QoS pour les utilisateurs. On a intégré FlowQoS avec OpenWrt pour permettre à l'utilisateur de spécifier la portion qu'il veut allouer pour chaque classe d'application via l'interface de contrôle de la passerelle domestique.
FlowQoS a été développé sous l'hypothèse que le FAI ne fait pas du DNS caching. Il est nécessaire de trouver une autre solution qui ne dépend pas du DNS pour la classification du trafic HTTP et HTTPS.



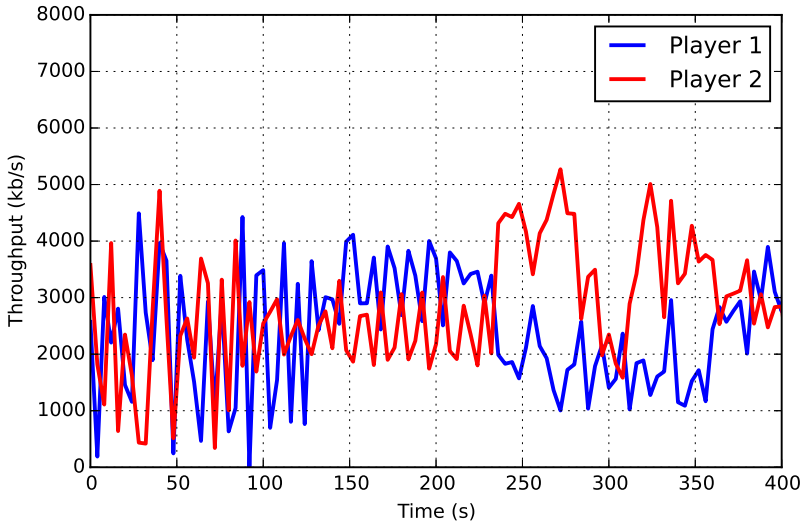
(a) La sélection du débit d'encodage sans VidShaper



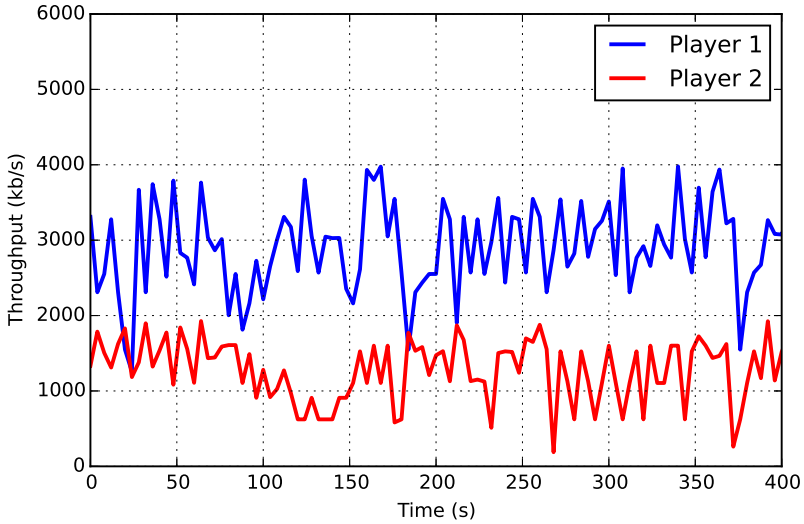
(b) La sélection du débit d'encodage avec VidShaper

FIGURE 3.17 – Variation du débit d'encodage pour le flux de streaming vidéo adaptatif avec et sans FlowQoS

Nous avons aussi présenté VidShaper, une solution qui permet d'assurer la QoS pour chaque flux vidéo lorsque plusieurs lecteurs de streaming adaptatif sont actifs compte tenu des résolutions des écrans des équipements. Cette solution permet de garantir une bonne QoE pour les utilisateurs quand plusieurs lecteurs de streaming vidéo adaptatif sont actifs dans le réseau domestique. Les nouveaux téléphones intelligents sont dotés d'une grande résolution de l'écran qui est contraignante avec VidShaper. Il est très intéressant de combiner la résolution et la taille de l'écran dans VidShaper afin de mieux gérer la bande passante au niveau du réseau domestique. L'objectif de ce chapitre est d'évaluer le potentiel de SDN et OpenFlow dans la gestion de la QoS dans le contexte de virtualisation des réseaux. Ceci est réalisé à travers un réseau de petite



(a) Le débit de téléchargement de la vidéo sans VidShaper



(b) Le débit de téléchargement de la vidéo avec VidShaper

FIGURE 3.18 – Les performances en termes de débit de téléchargement de la vidéo avec et sans VidShaper

taille faute de pouvoir disposer d'une plateforme de grande taille qui offre une granularité fine dans l'allocation de l'infrastructure physique.

Après avoir prouvé dans ce chapitre la faisabilité technique sur un petit réseau domestique (un routeur et un lien), les chapitres suivants seront consacrés au développement d'algorithmes de partage de ressources et de gestion de QoS dans les réseaux de grande taille, en supposant leur réalisabilité à l'aide de SDN et OpenFlow. La validation de ces algorithmes se réalisera par simulations.

Chapitre 4

Modélisation de l'allocation des ressources par la théorie des jeux

Sommaire

1 Introduction **67**
2 Rappel sur la théorie des jeux **68**
2.1 Représentation d'un jeu 68
2.1.1 Jeu sous forme extensive 69
2.1.2 Jeu sous forme stratégique 69
2.2 Stratégie pure, stratégie mixte et stratégie dominée 70
2.3 Les différentes catégories de jeu 70
2.3.1 Jeu coopératif et jeu non-coopératif 70
2.3.2 Jeu simultané et jeu séquentiel 71
2.3.3 Jeu à information complète et jeu à information incomplète 71
2.3.4 Jeu à somme nulle et jeu à somme non-nulle 71
2.4 L'équilibre de Nash 71
3 Jeux non-coopératifs pour l'allocation dynamique des ressources **72**
3.1 Jeu de négociation des ressources 73
3.2 Jeu d'allocation du nœud physique 77
3.2.1 Modélisation du jeu 77
3.2.2 Évaluation du jeu d'allocation du nœud physique 79
3.3 Jeu d'allocation du lien physique 82
3.3.1 Modélisation du jeu 82
3.3.2 Évaluation du jeu d'allocation du lien physique 84
4 Conclusion **87**

1 Introduction

Le déploiement de nouveaux services sur Internet est soumis aux contraintes de la QoS (débit, délai et fiabilité). La question cruciale pour un fournisseur d'infrastructure est comment

partager les ressources physiques d'un réseau entre plusieurs fournisseurs de services en considérant leurs demandes en termes de QoS et en assurant l'équité lors du partage. Ces ressources sont généralement la bande passante de chaque lien physique, les espaces mémoires et les cycles du processeur en chaque nœud.
Dans ce chapitre, nous allons apporter quelques éléments de réponse en proposant une approche basée sur la théorie des jeux permettant l'allocation dynamique des ressources pour plusieurs VNs. La théorie des jeux est une discipline mathématique qui analyse l'interaction stratégique entre plusieurs preneurs de décisions appelés "joueurs". Son objectif est d'étudier les situations où les gains d'un joueur, lors d'une prise de décision, dépendent des gains des autres joueurs [Rasmusen, 1994]. Dans le domaine des réseaux, cette théorie a été souvent utilisée dans des problématiques liées au routage dans les réseaux informatiques [Altman et al., 2006], au contrôle de flux [Key and McAuley, 1999], au contrôle d'accès [Charilas and Panagopoulos, 2010] et à la sécurité des réseaux [Roy et al., 2010]. Dans ce chapitre nous présentons d'abord quelques facettes de la théorie des jeux qui vont nous permettre de modéliser et d'analyser des situations de compétition entre les FSs et FIPs pour le partage des ressources physiques. Ensuite, nous modélisons cette allocation comme étant un jeu non-coopératif à deux niveaux où les joueurs entament un jeu de négociation pour le partage des ressources et ensuite un jeu d'approvisionnement des ressources.
La première étape concerne la phase de négociation entre le FS et le FIP pour le partage des ressources. La seconde étape concerne le partage des ressources. Cette étape est composée de deux jeux non-coopératifs, le premier pour l'allocation de nœud physique et le second pour l'allocation de lien physique. L'objectif des deux jeux est d'allouer équitablement les ressources physiques pour différentes VNs isolés qui partagent la même infrastructure en assurant la qualité de service pour chaque flux.
Notre approche vise à assurer la gestion et l'approvisionnement équitable des ressources dans une infrastructure réseau virtualisée.

2 Rappel sur la théorie des jeux

2.1 Représentation d'un jeu

Un jeu peut se présenter sous deux formes [Dutta, 1999]. La première est la forme extensive où on utilise un arbre de décision et les joueurs choisiront simultanément les stratégies qu'ils mettront en oeuvre. La seconde représentation est la forme stratégique où on utilise une matrice pour représenter les stratégies et les gains de chaque joueur. Si les actions de chaque joueur sont spécifiées de façon exhaustive, un jeu sous une forme extensive peut s'écrire aussi sous une forme stratégique.
On se propose d'étudier les différents concepts de la théorie des jeux à partir de l'exemple du dilemme du prisonnier. L'idée considère à imaginer deux voleurs complices arrêtés après avoir commis un délit. Un policier les interroge séparément afin d'obtenir des aveux. Il cherche le maximum de preuves conduisant à la lourde peine. Il propose une peine légère à celui qui parlera. L'ensemble des stratégies possibles dans ce jeu est l'ensemble {nier, avouer}. Avant

d'être arrêtés, les deux complices se sont promis de ne pas se trahir. Dans ce jeu les voleurs ont quatre possibilités. Soit tous les deux nient ou avouent, soit l'un avoue et l'autre nie. En conclusion, ils ont intérêt à coopérer. Cependant, en absence de communication entre eux, chacun tentera de trahir l'autre sachant que leur peine est plus longue que si les deux avaient choisi d'avouer. Il en découle naturellement que la meilleure stratégie des deux voleurs est de nier les faits.

2.1.1 Jeu sous forme extensive

La forme extensive d'un jeu prend en compte et en détail la structure séquentielle du problème de décision et les possibilités d'action [Myerson, 2013]. A chaque étape du jeu, un joueur effectue son choix en fonction d'une situation observée qui dépend en particulier des choix préalables des autres joueurs. Ces décisions sont représentées à l'aide d'un arbre dit arbre du jeu. La figure 4.1 illustre le jeu du dilemme du prisonnier sous forme stratégique.

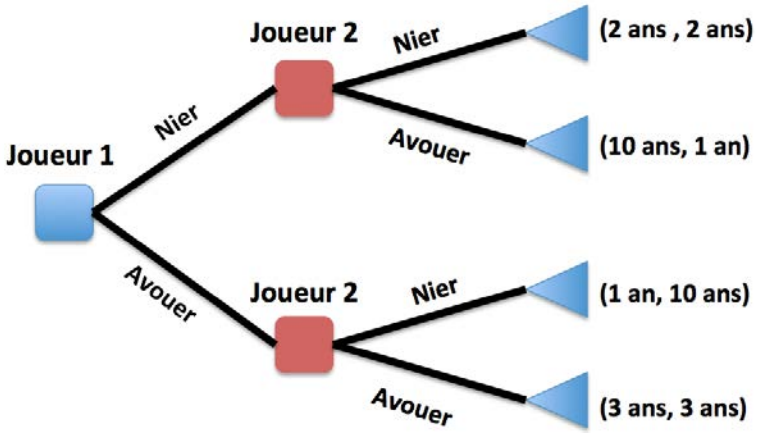


FIGURE 4.1 – Forme extensive d'un jeu

Un jeu sous forme extensive est défini par :

— Un ensemble N = 1, . . . , n de joueurs.

— Un arbre fini avec un nœud initial, des nœuds de décision et des nœuds terminaux. Chaque branche qui relie chaque nœud à ceux qui lui succèdent représente la stratégie d'un joueur. Ce sont les actions permises à chaque nœud où le joueur est susceptible de prendre une décision.

— Une fonction d'affectation qui indique à chaque nœud quel est le joueur qui a la main.

— Une fonction d'utilité qui associe à chaque nœud terminal un vecteur dont les composantes sont les nombres représentant les gains de chacun des joueur.

2.1.2 Jeu sous forme stratégique

Un jeu sous forme stratégique représente une interaction entre joueurs rationnels [Myerson, 2013] où chacun choisit simultanément une stratégie.
Un jeu sous forme stratégique est défini par :

— Un ensemble N = 1, . . ., n de joueurs.

— Un ensemble de stratégies $S_i = x_1^i, ..., x_{ni}^i$ pour chaque joueur i.

— Une fonction d'utilité $u_i : S_1$ x · · · x $S_n \rightarrow \mathbb{R}$, qui à chaque ensemble de stratégies associe les gains du joueur i.

Le tableau ci-dessous représente le jeu du dilemme du prisonnier sous forme stratégique.

		Joueur 2	
		Avouer	Nier
Joueur 1	Avouer	(3 ans, 3 ans)	(1 an, 10 ans)
	Nier	(10 ans, 1 an)	(2 ans, 2 ans)

TABLE 4.1 – Forme stratégique du jeu du dilemme du prisonnier

2.2 Stratégie pure, stratégie mixte et stratégie dominée

Une stratégie pure est une stratégie déterministe qui détermine l'action que doit prendre un joueur durant le jeu. Elle donne une définition complète de la façon avec laquelle un joueur se comporte face à une situation aléatoire qu'il pourrait faire face.
Une stratégie mixte est l'affectation d'une probabilité à chaque stratégie pure. Cela permet à un joueur d'attribuer une certaine distribution de probabilité sur l'ensemble de ses stratégies pures et d'en choisir une au hasard.
Une stratégie dominante est optimale quelles que soient les actions des autres joueurs. Il s'agit d'une stratégie qui lui donne toujours un gain supérieur au gain qu'il peut attendre de toutes ses autres stratégies indépendamment des stratégies des autres joueurs.

2.3 Les différentes catégories de jeu

Comme nous l'avons expliqué précédemment, la caractéristique fondamentale d'un jeu est la dépendance des gains vis-à-vis des facteurs. Le premier facteur est son choix de stratégie. Le second est lié aux stratégies effectuées par les autres joueurs. Les différents contextes d'interaction entre les joueurs définissent le type de jeu entre eux. On peut classer les différents types de jeu selon ces contextes (à titre d'exemple : la relation entre les joueurs ou bien le déroulement du jeu). Ce contexte peut aussi dépendre du déroulement du jeu dans le temps en cas de jeux simultanés et de jeux séquentiels. Ce contexte peut aussi dépendre des informations complètes ou incomplètes relatives aux joueurs. Cela peut aussi dépendre du gain ou de la perte de chaque joueur, comme dans l'exemple les jeux à somme nulle ou non nulle.

2.3.1 Jeu coopératif et jeu non-coopératif

Un jeu coopératif décrit la situation dans laquelle les joueurs communiquent librement entre eux en passant des accords et en formant des coalitions [Driessen, 1991]. Ceci se fait sous la forme de contrat qui prend en compte les sanctions lors du non-respect de ces règles. Ces

dernières relient les joueurs de manière contraignante dont le but est d'obtenir de meilleurs résultats pour leurs membres. Tous les joueurs gagnent ou perdent ensemble.
Un jeu non-coopératif [Basar et al., 1995] correspond à une situation dans laquelle la décision d'un joueur est prise sans consultation avec le reste. Cette décision est totalement autonome et vise l'intérêt personnel plutôt que l'intérêt général, dans le but d'augmenter son propre gain. Dans ce cas, chaque joueur adopte un comportement qui dépend du comportement qu'adopteront, selon lui, les autres joueurs.

2.3.2 Jeu simultané et jeu séquentiel

Un jeu simultané est un jeu où chaque joueur choisit sa propre stratégie sans connaître celle choisie par les autres joueurs. Ce choix se fait au même temps pour tous les joueurs. La représentation de ce jeu se fait généralement sous une forme stratégique.
Un jeu séquentiel est un jeu où la stratégie d'un joueur est conditionnée par celle choisie dans un jeu précédent. Ce jeu spécifie l'ordre des décisions de chaque joueur. Il est généralement représenté sous forme extensive.

2.3.3 Jeu à information complète et jeu à information incomplète

Un jeu à information complète est un jeu où chaque joueur, lors de sa prise de décision de stratégie, est averti au préalable des possibilités d'actions des autres joueurs ainsi que les gains éventuels résultant de leurs actions.
Un jeu à information incomplète décrit la situation où le joueur n'a aucune connaissance préalable sur les actions ou les gains des autres. Ces jeux sont aussi appelés jeux bayésiens.

2.3.4 Jeu à somme nulle et jeu à somme non-nulle

Un jeu à somme nulle est un jeu dont la somme des gains est nulle indépendamment des stratégies choisies par les autres. Ce type de jeu est à deux joueurs et le gain de l'un constitue obligatoirement une perte pour l'autre.
Un jeu à somme non-nulle est un jeu où les joueurs trouvent un intérêt commun et la somme de leurs gains ou de leurs pertes est soit inférieure soit supérieure à zéro.

2.4 L'équilibre de Nash

L'équilibre de Nash (*Nash Equilibrium*) est un concept fondamental dans la théorie des jeux non-coopératifs [Nash et al., 1950]. C'est l'outil principal permettant de formaliser le comportement optimal des joueurs en situation d'interaction stratégique, lorsque les gains de chacun ne dépendent pas seulement de leur stratégie mais aussi de celle des autres joueurs. L'équilibre est une combinaison de stratégies pour laquelle la stratégie d'un joueur est la meilleure réponse (*best response*) aux stratégies des autres joueurs. Un vecteur de stratégies forme un équilibre de Nash si l'action de chaque joueur est la la meilleure réponse aux n-1 autres stratégies des autres joueurs. En d'autre termes, pour chaque joueur, il n'existe pas de déviation unilatérale profitable. Cet équilibre correspond à la situation où aucun joueur n'a intérêt à changer de stratégie

puisque sa stratégie est la meilleure réponse aux autres stratégies dans le jeu.
Nash a montré qu'avec un nombre fini de joueurs et de stratégies pures, l'équilibre est toujours réalisé. L'existence d'un équilibre n'implique pas nécessairement qu'il est optimal. En effet, d'autres stratégies peuvent les conduire à un gain supérieur. Le théorème de Nash stipule l'existence d'au moins un équilibre, mais pas son unicité.

Soit n le nombre de joueurs ($n \in \mathbb{N},\ n \geq 2$) et S_i l'ensemble des stratégies du joueur i. Sa fonction de stratégie est $u_i : S_1$ x $\cdot\ \cdot\ \cdot$ x S_n.
Le profil des stratégies $s = (x_1, \cdot\ \cdot\ \cdot, x_n)$ est un équilibre de Nash s'il correspond à la meilleure réponse aux choix $\hat{s}$ des autres $(n-1)$ joueurs.

$$\forall\, i \in \mathbb{N},\ \forall\, k_i \in S_i,\ u_i(s) \geq u_i(k_i, \hat{s}). \tag{4.1}$$

L'équilibre de Nash vérifie les trois propriétés suivantes :

- La rationalité : chaque joueur vise son propre intérêt individuel.
- La spontanéité : la convergence vers l'équilibre est atteinte sans aucune intervention extérieure.
- la stabilité : Les joueurs ne souhaitent jamais dévier de l'équilibre lorsqu'il est atteint.

Dans l'exemple du dilemme du prisonnier l'équilibre de Nash est atteint lorsque les deux joueurs choisissent leur meilleure stratégie qui consiste à avouer les faits. En effet, le profil de stratégies (avouer, avouer) est l'équilibre de Nash même si la condamnation est moins lourde s'ils nient les faits. Les autres stratégies ne sont pas les meilleures réponses par rapport aux choix des autres joueurs. À titre d'exemple, si l'un des joueurs choisit de nier, il risque une peine de 10 ans prison quand le second joueur décide d'avouer.

3 Jeux non-coopératifs pour l'allocation dynamique des ressources

Dans cette partie nous abordons la formulation du mécanisme d'allocation des ressources comme un ensemble de jeux non-coopératifs. Ce mécanisme sera exécuté en deux étapes. La première étape concerne le jeu non-coopératif de négociation entre le FIP et le FS. La seconde étape est dédiée au partage, par un ou plusieurs FIPs, de la bande passante et des ressources du nœud entre les différents fournisseurs de services. Le réseau physique est représenté par un graphe non orienté (V, E) où V désigne l'ensemble des nœuds physiques et E l'ensemble des liens.
Dans le cas d'un réseau SDN géré avec OpenFlow le processus de découverte se fait implicitement en partant du fait qu'à chaque nœud est attaché à un contrôleur. D'abord, chaque FS a la possibilité de choisir l'algorithme de mappage le plus adapté à ses besoins. Ensuite, pour recevoir une fraction de chaque nœud ou lien physique, le FS s'engage dans une compétition avec les autres FS. La présente approche vise à réduire la complexité de gestion de réseau et à éviter la dégradation des performances entre la configuration logique et la configuration physique du réseau.

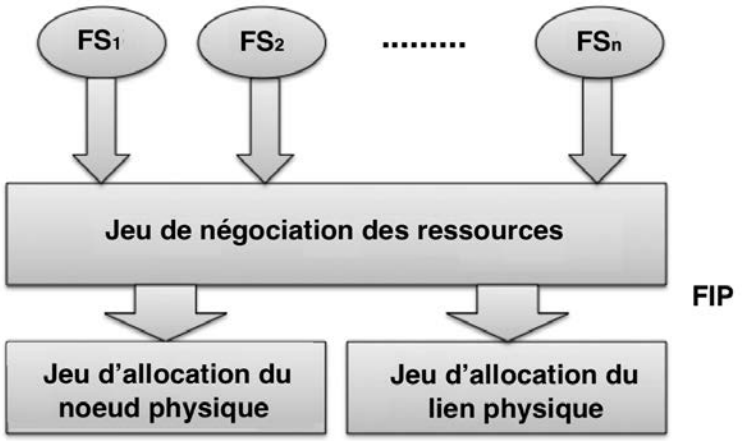


FIGURE 4.2 – Le cadre des jeux non-coopératifs pour l'allocation dynamique des ressources

La figure 4.2 illustre les différentes étapes relatives à l'allocation dynamique des ressources physiques du réseau. En effet, chaque fournisseur de services soumet sa requête traduisant sa demande en ressources partagées (nœuds et liens) à chaque fournisseur d'infrastructure. Dans un premier temps, le FIP choisit d'accepter ou de décliner la demande. Si la requête du FS est acceptée, le FIP aura à allouer une portion de chaque lien et de chaque nœud. Ensuite, chaque FS déploie ses protocoles pour créer un nouveau réseau virtuel. Selon la qualité du service demandé par les utilisateurs du réseau virtuel, le FS peut réajuster sa demande et répondre aux exigences des utilisateurs finaux.

Pour des soucis de clarté, nous présentons dans le tableau 4.2 la notation utilisée .

3.1 Jeu de négociation des ressources

Dans cette partie, nous présentons une analyse exhaustive de l'interaction entre le FIP et le FS, formulée mathématiquement en tant que jeu non-coopératif réalisé entre eux. En effet, dans le but de répondre à la demande en termes de QoS des utilisateurs de chaque réseau virtuel, le fournisseur de services veille au partage les ressources gérées par un ou plusieurs FlPs . Pour chaque requête de partage des ressources, une instance de ce jeu prend place.

Ce jeu a été initialement formulé par Charilas et al. [Charilas et al., 2009] pour éviter la congestion du lien radio, entre les fournisseurs de réseaux et les clients, dans un environnement de réseau sans fil. Pour cela, considérons n fournisseurs de services qui partagent les ressources gérées par m fournisseurs d'infrastructures. Chaque FS peut, à tout moment, choisir n'importe quel FIP pour lui soumettre la requête de partage. Chaque FIP accepte la requête ou la rejette en cas d'insuffisance des capacités du réseau physique. D'un autre côté, le fournisseur de services a le choix de rester ou de quitter le FIP avec la possibilité de renouveler sa requête en cas de dégradation de la qualité de service. Cependant, en cas d'abandon du FIP, le FS sera exposé à une pénalité. Cette procédure peut être traduite par un mécanisme de tarification dans un environnement réseau virtualisé. Elle peut être, aussi, utilisée dans un système d'aide à la décision

Symbol	Description
$B_{i,j,k}$	Vecteur des enchères du FS_i pour ses réseaux virtuels VN_j pour partager le nœud physique N_k
G_i	Budget du FS_i durant le jeu.
$F_{i,j,k}$	Fraction du nœud physique N_k reçue par FS_i pour ses réseaux virtuels VN_j
P_k	Prix du nœud N_k
$U_{i,j}$	Fonction d'utilité de FS_i pour ses réseaux virtuels VN_j
B^i_{esp}	Bande passante espérée de FS_i
B^i_{rec}	Bande passante recommandée pour FS_i
B^i_{req}	Bande passante demandée par FS_i
B_{dispo}	Capacité disponible d'un lien physique
B^i_{all}	Bande passante allouée au FS_i
p^i	Prix imposé par le FIP au FS_i
$w^i(t)$	Disposition à payer du FS_i à l'instant t
ρ	Paramètre de flexibilité d'un FIP
R^S	Revenue fixe d'un FS
P	Pénalité payée par un FS
R^I	Revenue fixe d'un FIP
C	Cout espéré du service
$L(t)$	Pertes au cours du temps d'un FIP lorsque plusieurs FSs sont insatisfaits
$F(t)$	Pertes au cours du temps d'un FIP lorsqu'un FS décide de le quitter lors du partage

TABLE 4.2 – Les différentes notations utilisées dans notre modélisation

gérant l'allocation des ressources physiques entre le FIP et FS afin de garantir les performances du réseau.

Lors de la négociation, la décision d'acceptation ou de refus de la requête du FS est étroitement liée à la charge actuelle et à la capacité de chaque nœud et lien. Ainsi, chaque fois qu'une nouvelle requête est reçue par le FIP, une instance du jeu est lancée. L'ensemble de stratégies du FIP est $\{(I_1), (I_2)\}$ et celui du FS est $\{(S_1), (S_2)\}$. Les gains dans ce jeu seront exprimés par les deux matrices $\text{X} = [x_{i,j}]_{2*2}$ pour le FIP et $\text{Y} = [y_{i,j}]_{2*2}$ pour la FS. Le tableau 4.3 résume les relations entre les gains et les stratégies du jeu.

TABLE 4.3 – La relation entre les gains et les stratégies du FS et du FIP

	FS quitte (S_1)	FS reste (S_2)
FIP accepte (I_1)	x_{11}, y_{11}	x_{12} , y_{12}
FIP rejette (I_2)	x_{21} , y_{21}	x_{22} , y_{22}

Nous supposons que le FS s'engage dans un contrat temporaire avec un FIP lorsque sa requête est acceptée. Cependant, le FIP peut rediriger le FS à un autre FIP. Ainsi, dans ce jeu non-coopératif quatre scénarios sont identifiés :

— Le FS et le FIP sont tous deux satisfaits, dans ce cas, les jeux d'allocation du nœud et l'allocation du lien prennent acte.

— Le FS et le FIP sont tous deux insatisfaits, le FS peut faire une nouvelle requête pour un autre FIP sans payer de pénalité.

— Si seul le FIP est satisfait et le FS refuse de partager les ressources du FIP, dans ce cas le FS doit payer une pénalité.

— Si seul le FS est satisfait, alors le FIP décline la requête et aucune pénalité n'est imposée dans ce cas au FS.

Ce jeu non-coopératif peut être représenté par la matrice des gains pour le FS et une autre pour le FIP. Elle résume les cas possibles et les gains qui en découlent selon la stratégie choisie par chacun. Celle du FS est définie comme suit :

$$Y = \begin{pmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{pmatrix} = \begin{pmatrix} \alpha R^S - \beta P & R^S \\ \alpha R_0^S - \beta P & R_0^S \end{pmatrix}. \tag{4.2}$$

— Si le FS décide de quitter le FIP qui a accepté à son tour la requête, alors, le FS est obligé, dans ce cas, de payer de son budget R^S une pénalité de résiliation anticipée P. Les deux variables (R^S et P) sont multipliées par un poids qui reflète la préférence du FS en fonction de son choix d'économie d'argent ou de satisfaction. Cette pondération est préalablement définie dans le profil du FS.

— Dans le cas où le FS choisit de rester avec le FIP mais ce dernier rejette la requête, un gain d'une somme fixe R_0^S est allouée au FS.

— Si le FS choisit de rester et le FIP accepte la requête, le FS reçoit un gain R^S. Ce gain exprime le budget du FS après avoir payé le FIP pour l'utilisation de son infrastructure.

— Si le FS décide de quitter le FIP qui a déjà accepté la requête, une pénalité pondérée est alors déduite du budget du FS.

La matrice des gains pour un FIP est définie comme suit :

$$X = \begin{pmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{pmatrix} = \begin{pmatrix} R^I + P + C - F(t) - L(t) & R^I + C - F(t) \\ R^I + P - L(t) & R^I \end{pmatrix}. \tag{4.3}$$

— Quand un fournisseur d'infrastructure décide d'accepter la requête d'un fournisseur de services qui à son tour décide de rester, le FIP diminue une partie de la ressource offerte pour les autres FS. Le FIP prend le risque que d'autres FSs décident de quitter en raison de la dégradation de la QoS. À titre d'exemple, un FS observe une augmentation des délais de transit des paquets sur un lien physique. Ainsi, son gain est égal à un revenu R^I gagné lors de l'acceptation des autres FS, auquel on ajoute le revenu prévu C du FS admis moins les pertes $F(t)$ en raison de l'insatisfaction des autres FS.

— Quand le FIP rejette la demande d'un FS qui choisit à son tour de rester, le gain du FIP est égal à R^I qui est la somme acquise par tous les FSs qui partagent cette infrastructure.

— Le gain d'un FIP qui rejette la requête d'un FS ayant choisi de quitter est égal au total des sommes perçues R^I augmenté de la pénalité P et diminué des pertes $L(t)$ du FIP.

— Lorsqu'un FIP choisit d'accepter la requête du FS qui le quitte, le gain du FIP est égal à la somme acquise des revenus R^I, augmentée du montant de la pénalité P et du coût prévu C du service mais diminué des pertes $F(t)$ et des pertes $L(t)$ lorsqu'un groupe de FSs sont insatisfaits.

Dans le jeu de négociation, discuté dans [Charilas et al., 2009], il existe deux cas correspondant à l'équilibre de Nash.

— Le premier cas se produit lorsque le FIP a encore des ressources disponibles pour certains FS. Par conséquent, les demandes seront automatiquement acceptées et la probabilité qu'un FS quitte est quasi nulle. Donc, la stratégie I_1S_2 où le FIP accepte la requête et le FS décide de rester correspond à l'équilibre de Nash.

— Le deuxième cas se produit lorsque la réseau physique est surchargé et la probabilité pour qu'un FS quitte le FIP est non nulle. Dans ce cas, l'équilibre de Nash est la stratégie I_iS_j dépendante des gains durant le jeu. La nouvelle demande du FS est acceptée si le revenu C gagné par le FIP est supérieur aux pertes $F(t)$ lorsqu'un groupe de FSs décide de quitter. Autrement, il est conseillé au FIP de rejeter la demande du FS si ce dernier risque de détériorer les performances de la QoS offertes aux FSs.

Selon ces deux cas, l'état d'équilibre de Nash est atteint à n'importe quel moment du jeu pour une stratégie paire I_iS_j définie comme suit :

$$i = \begin{cases} 1\ Si\ x_{11} \geq x_{21}, \\ 2\ Si\ x_{11} < x_{21}, \end{cases} \Rightarrow i = \begin{cases} 1\ Si\ C \geq F(t), \\ 2\ Sinon, \end{cases} \tag{4.4}$$

$$j = \begin{cases} 1\ Si \begin{cases} \{i = 1\ et\ y_{11} \geq y_{12}, \} \\ ou\ \{i = 2\ et\ y_{21} \geq y_{22}, \} \end{cases} \\ 2\ Si \begin{cases} \{i = 1\ et\ y_{11} < y_{12}, \} \\ ou\ \{i = 2\ et\ y_{21} < y_{22}.\} \end{cases} \end{cases} \tag{4.5}$$

3.2 Jeu d'allocation du nœud physique

Cette section est dédiée au développement d'un modèle d'allocation dynamique de nœud dans une infrastructure réseau virtualisée. Ce modèle est basé sur une approche similaire à celle de Feldman et al. [Feldman et al., 2009], proposée pour l'allocation des ressources de calcul dans un réseau informatique. Cependant, nous améliorons le jeu en proposant un algorithme de prédiction de la prochaine stratégie pour chaque joueur. Cet algorithme vise à maximiser l'utilisation des ressources des nœuds physiques et à assurer l'équité du partage sous contraintes de QoS.

Le jeu de l'allocation du nœud se base sur un mécanisme de partage équilibré où chaque FS répartit son budget entre plusieurs nœuds physiques et reçoit une fraction de chaque nœud proportionnelle à son offre. Grâce à ce modèle, nous garantissons un système de répartition équitable pour plusieurs réseaux virtuels partageant les mêmes nœuds physiques. Chaque nœud a besoin d'un hyperviseur qui permet de créer et d'exécuter plusieurs nœuds virtuels. En fait, l'hyperviseur est responsable du partage des ressources du nœud physique entre plusieurs VNs. L'approche cherche à satisfaire les besoins en ressources de chaque FS tout en optimisant l'utilisation des ressources physiques. Ces derniers sont généralement les cycles d'un processeur réseau, du nombre d'accès à la mémoire et de la taille de la mémoire tampon du nœud physique. Dans ce travail nous nous focalisons sur la maximisation des gains des joueurs plutôt que sur l'unicité de l'équilibre de Nash.

3.2.1 Modélisation du jeu

Considérons n fournisseurs de services, qui déploient l réseaux virtuels partageant m nœuds physiques gérés par plusieurs FIPs. Nous attribuons à chaque FS un budget G_i qui dépend de la qualité de service demandée par le FS pour satisfaire ses utilisateurs. Plus la demande de qualité de service est grande plus le budget est élevé pour un FS.

Dans ce jeu, la stratégie d'un FS_i est de soumettre une enchère pour chaque nœud physique qui sera utilisé par les réseaux virtuels déployés par ce fournisseur de services. Par la suite, le FS reçoit, selon son offre, une fraction de chaque nœud. De plus, le FS attribue un poids à chaque nœud physique requis. Les poids reflètent le degré de préférence pour chaque nœud.
Pour le partage des nœuds N_k, nous définissons, comme suit, l'enchère soumise par un FS_i pour son réseau virtuel VN_j :

$$(B_{i,j,k})_{k=1,....,m} = B_{i,j,1}, B_{i,j,2}, ..., B_{i,j,m}. \tag{4.6}$$

La somme des enchères pour un FS_i est toujours inférieure ou égale à son budget G_i défini comme suit :

$$\sum_{j=1}^{l} \sum_{k=1}^{m} B_{i,j,k} \leq G_i. \tag{4.7}$$

Le budget d'un FS dépend de la QoS exigée par les utilisateurs de chaque réseau virtuel. Il s'agit d'une valeur monétaire qui permet d'offrir une priorité aux FS dont le service déployé sur un VN a plusieurs contraintes de QoS. À titre d'exemple, on affecte un gros budget à un FS qui

fournit de la vidéo à la demande (VoD) et un budget plus petit à celui qui a déployé un service de stockage sur le cloud.
Un FS_i reçoit, pour chaque réseau virtuel déployé VN_j, une fraction égale à son enchère divisée par le prix du nœud physique. Le prix P_k d'un nœud physique est défini comme la somme de toutes les enchères soumises, comme suit :

$$P_k = \sum_{i=1,}^{n} \sum_{j=1,}^{l} B_{i,j,k}. \tag{4.8}$$

La fraction $F_{i,j,k}$ du nœud physique k reçue par un réseau virtuel VN_j d'un FS s'écrit :

$$F_{i,j,k} = \frac{B_{i,j,k}}{P_k}. \tag{4.9}$$

La fonction d'utilité, notée $U_{i,j}$, représente les fractions reçues $F_{i,j,1}$, $F_{i,j,2}$, ..., $F_{i,j,mt}$ et les poids $W_{i,j,1}$, $W_{i,j,2}$, ..., $W_{i,j,m}$ attribués à chaque nœud. Ce vecteur poids définit les préférences de chaque FS pour chaque nœud. Ainsi, la fonction d'utilité est une fonction multilinéaire qui s'écrit sous la forme :

$$\begin{aligned} U_{i,j}\,(F_{i,j,1},\, F_{i,j,2},\, ...F_{i,j,m}) \;=& \\ &= W_{i,j,1}\, F_{i,j,1} + W_{i,j,2}\, F_{i,j,1} + \;...\; + W_{i,j,m} F_{i,j,m}. \end{aligned} \tag{4.10}$$

En maximisant la fonction d'utilité, chaque FS cherche a minimiser les délais lorsque un nœud physique traite les paquets. Généralement, quand un paquet arrive au niveau du nœud, il est orienté vers l'interface physique de sortie. Le délai de traitement représente le temps écoulé pour qu'un routeur traite un paquet. En maximisant la fraction demandée, les paquets d'un réseau virtuel mettent moins du temps pour être traité par le nœud physique. Dans notre approche, nous supposons que tous les FSs visent à servir leurs propres intérêts en essayant de maximiser leur fonction d'utilité et améliorer leurs performances sans se soucier des autres intervenants. Pour atteindre l'état d'équilibre de Nash, la meilleure stratégie pour un FS_i est la solution équitable suivante :

$$\begin{cases} Maximizer \quad U_i = (\frac{B_{i,j,1}}{P_1}, \frac{B_{i,j,2}}{P_2}, ... \frac{B_{i,j,m}}{P_m}), \\ \text{É}tant\ donn\acute{e}\ que \quad \sum_{j=1}^{l} \sum_{k=1}^{m} B_{i,j,k} = G_i \quad et \quad B_{i,j,k} \geq 0. \end{cases} \tag{4.11}$$

Chaque nœud physique a une capacité limitée en termes de bande passante, de mémoire, et de traitement des paquets. Ainsi, l'équité de l'allocation de nœud entre les FSs est indispensable. Dans notre jeu, nous supposons que le FS essaye de maximiser sa fonction d'utilité en prédisant les prochaines enchères. Pour cela, chaque FS utilise les performances des instants passés et l'ancien prix du nœud pour trouver la meilleure réponse maximisant sa fonction d'utilité dans le jeu. Il s'agit alors de trouver le goulot d'étranglement pour chaque VN ; c'est-à-dire le nœud physique où les paquets, d'un VN donné déployé par un FS, passent le maximum de temps de traitement. Chaque réseau virtuel déployé a un goulot d'étranglement à un moment donné. L'idée de la prédiction est de minimiser le retard des paquets dans le goulot d'étranglement des ressources et maintenir les mêmes performances des autres ressources.

A chaque instant du jeu, le goulot d'étranglement change. Par exemple, à un instant T et pour un réseau virtuel VN_j déployé par le fournisseur de services FS_i, le goulot peut être le nœud physique N_3. Par la suite, et en raison de la nature sporadique du flux, le goulot d'étranglement peut devenir le nœud physique N_5. Le fournisseur d'infrastructure tente d'utiliser 100 % de la performance du nœud.

À l'instant $T+1$ lorsque le nœud physique k' est considéré comme goulot d'étranglement pour FS_i l'enchère soumise par ce dernier est égale à :

$$B_{i,j,k'}^{T+1} = P_{k'}^{T} * F_{i,j,k'}^{req}. \tag{4.12}$$

L'idée est d'utiliser le prix de la ressource k' à l'instant T pour prédire le nouveau prix et minimiser les délais des paquets.

La prochaine enchère de FS_i à l'instant $T+1$ pour les autres nœuds physiques qui ne sont pas des goulots d'étranglement est égale à :

$$\begin{cases} (B_{i,j,k}^{T+1})_{k\neq k'} = argmax((B_{i,j,k}^{T})_{k\neq k'}), \\ \text{Étant donné que} \quad \sum_{j=1}^{l}\sum_{k=1}^{m} B_{i,j,k} \leq G_i. \end{cases} \tag{4.13}$$

3.2.2 Évaluation du jeu d'allocation du nœud physique

Dans cette section, nous présentons une évaluation du jeu d'allocation du nœud physique par simulation numérique. Nous supposons que quatre réseaux virtuels sont crées et déployés par quatre FSs différents (FS_1, FS_2, FS_3, FS_4). Ces derniers vont partager les ressources de quatre nœuds physiques (N_1, N_2, N_3, N_4) gérés par un seul fournisseur d'infrastructure. Nous essayons de simuler la fraction reçue par le FS dans le jeu. La figure 4.3 illustre la topologie de l'infrastructure physique partagée par les différents FSs.

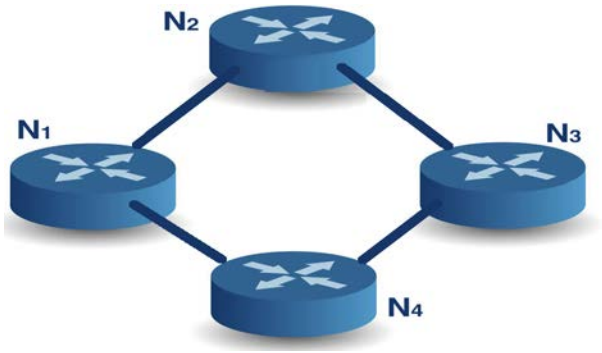


FIGURE 4.3 – Topologie de l'infrastructure physique partagée

Chaque FS peut recevoir une fraction qui correspond à un nombre de cycles de traitement d'un processeur de réseau ou du nombre d'accès à la mémoire. Nous attribuons quatre budgets (G_1, G_2, G_3, G_4) aux quatre FSs. Le budget alloué pour le fournisseur de services dépend du niveau de la qualité du service demandée. Nous supposons que $G_1 \geq G_2 \geq G_3 \geq G_4$. Nous

surveillons les délais de traitement des paquets dans chaque nœud et la fraction reçue par les fournisseurs de services à chacun des instants T et $T+1$. Chaque FS prédit les prochaines enchères afin de trouver la meilleure réponse pendant le jeu à l'instant $T+1$ et atteindre l'équilibre dans le jeu.

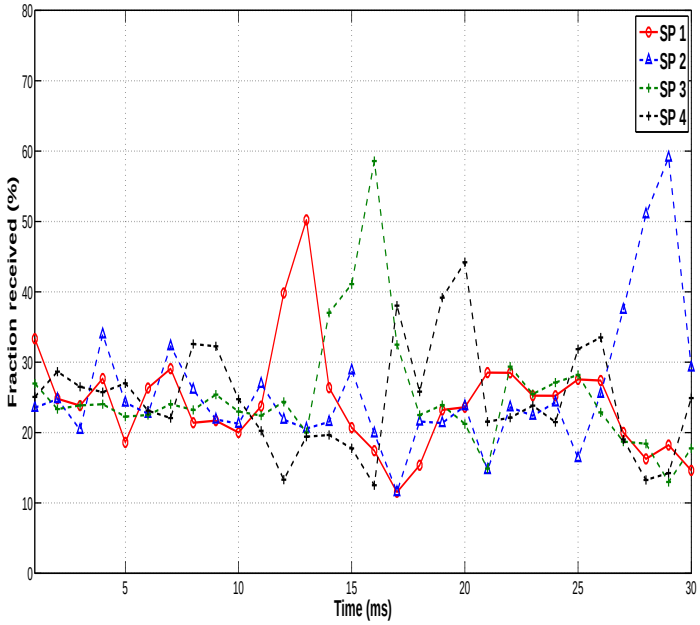


FIGURE 4.4 – Evolution au cours du temps de la fraction du nœud N_1 reçue par chaque FS

La figure 4.4 illustre la variation de la fraction du nœud N_1 reçue par chaque FS en fonction du temps. En fait, ce nœud est un goulot d'étranglement pour les quatre FS à différents intervalles de temps puisque chacun d'eux cherche à recevoir une fraction plus élevée de ce nœud au cours du temps.
La figure 4.5 montre le délai des paquets à l'intérieur du nœud N_1 pour un FS donné, Les délais des paquets pour un réseau virtuel déployé par le fournisseur de services FS_1 à $T = 11$ ms et $T = 12$ms sont élevés.
Le nœud N_1 est devenu un goulot d'étranglement. Les délais au nœud N_1 sont supérieurs aux délais dans les autres nœuds. En considérant ces valeurs, le FS_1 augmente son enchère aux instants $T = 12$ ms et $T = 13$ ms pour recevoir plus de fraction du nœud N_1. Il prédit cette enchère en fonction du prix du nœud N_1 aux instants où le nœud devient un goulot d'étranglement.

Dans notre étude d'allocation de nœud dynamique, nous cherchons à allouer un ensemble fini des ressources de nœuds physiques à un ensemble de FS en veillant au respect du critère d'équité. Dans le but de contrôler le respect du critère de l'équité du jeu, les courbes de la figure 4.6 illustre la moyenne de la fraction d'un nœud reçue pour chaque FS dans un intervalle égal à 20ms. Dans le cas d'une allocation statique équitable, la fraction reçue par un FS est égale à $1/n$ où n est le nombre de FS qui partage ce nœud physique.
Comme illustré sur la figure 4.6, la fluctuation de la fraction moyenne reçue dans ces intervalles

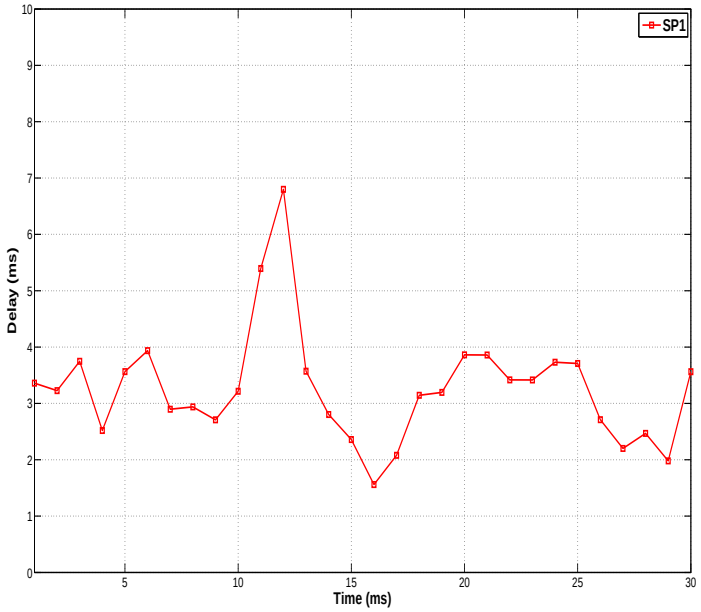


FIGURE 4.5 – Le délai de traitement moyen des paquets d'un réseau virtuel déployé par FS_1 dans le nœud N_1

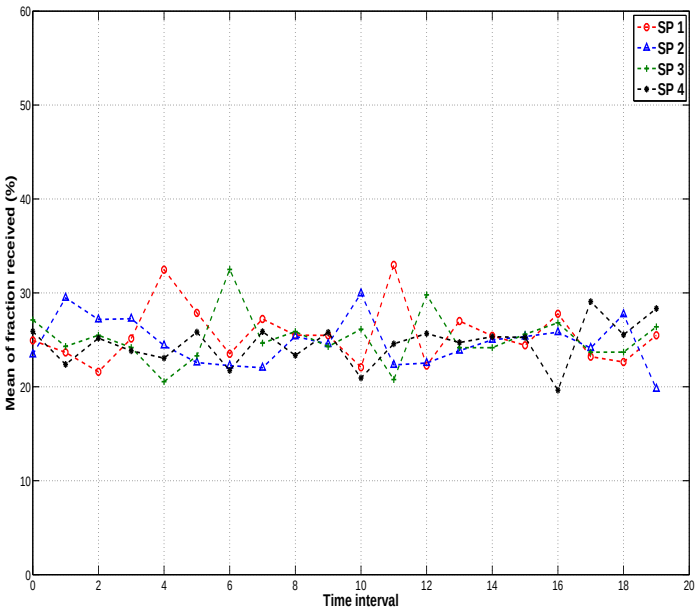


FIGURE 4.6 – La fraction moyenne d'un nœud physique reçue par chaque FS

de temps est de l'ordre de $\frac{1}{4}$ de la capacité du nœud physique pour chaque FS. Dans ce cas on peut conclure que notre jeu distribue équitablement les ressources des nœuds physiques aux

différents FS. Notre mécanisme offre ainsi une meilleure utilisation des ressources physiques de ce nœud par rapport à une approche statique qui ne permet pas la reconfiguration du schéma d'allocation des ressources.

3.3 Jeu d'allocation du lien physique

Dans cette section, nous modélisons le jeu d'allocation de la bande passante entre plusieurs FSs lors du partage de la bande passante des liens physiques dans une infrastructure physique gérée par un FIP. Le concept de ce jeu a été introduit par Ji et al. [Ji et al., 2004] pour les réseaux DiffServ. Ce jeu d'allocation de lien peut être implémenté comme système de facturation basé sur l'utilisation de la bande passante de plusieurs FSs et sur le prix du service dans un environnement de réseau virtualisé.

3.3.1 Modélisation du jeu

Considérons n fournisseurs de services dont la requête a déjà été acceptée par un seul FIP et ces FSs ont décidé de rester pour partager les ressources de ce réseau. Le nombre des FSs est limité par le jeu de la négociation de la bande passante. Le FIP surveille l'usage de chaque lien physique et notifie à chaque FS_i sa bande passante recommandée B^i_{rec}. Chaque FS_i a une bande passante espérée B^i_{esp}. Sa fonction d'utilité est la même pour une allocation de bande passante égale ou supérieure à B^i_{esp}. La capacité disponible d'un lien physique est définie par le paramètre B_{dispo}.
Dans ce jeu non-coopératif, nous étudions le cas où un FS_i soumet une requête B^i_{req} de partage de la bande passante pour chaque lien physique. Par la suite, il tente également de maximiser son gain pendant le jeu. Par conséquent, le FIP est en charge d'allouer la bande passante B^i_{all} à chaque FS_i. Les FSs agressifs sont pénalisés par la hausse des prix du service. Nous définissons p^i, le prix du service infligé à un FS_i. Chaque FS_i a une disposition à payer $w^i(t)$ (*willingness to pay*) afin de partager l'infrastructure réseau. Elle désigne la façon dont un FS estime l'ordre de grandeur de prix qu'il est prêt à payer pour acquérir une fraction de la bande passante d'un lien physique. Ce concept de la théorie du consommateur en microéconomie est très utilisé pour introduire une relation entre le prix d'un service et l'allocation des ressources [Falkner et al., 2000, Yaïche et al., 2000, Maillé and Tuffin, 2004]. Nous supposons que la valeur de $w^i(t)$ est toujours supérieure au prix initial du service facturé par le FIP. Un fournisseur de services FS_i avec une faible disposition à payer tentera de maximiser son gain sans chercher à demander une bande passante supérieure à la bande passante recommandée.
Ainsi, le gain pour un FS_i est défini comme suit :

$$u^i(t) = \begin{cases} (w^i(t) - p^i)\, B^i_{all} & if \;\; B^i_{all} \leq B^i_{rec} \\ (w^i(t) - \frac{B^i_{all}}{B^i_{rec}} p^i) B^i_{all} & if \;\; B^i_{rec} < B^i_{all} \leq \rho\, B^i_{rec} \\ (w^i(t) - \rho\, p^i)\, \rho\, B^i_{rec} & if \;\; B^i_{req} > \rho\, B^i_{rec} \end{cases} \tag{4.14}$$

Dans le jeu d'allocation du lien physique, un FS_i ayant une grande disposition à payer $w^i(t)$

peut se permettre d'exiger autant de bande passante qu'il désire. Nous limitons sa demande par ρ afin d'éviter une carence au niveau des autres FSs. Le choix de ce paramètre dépend de la capacité du lien physique et de la bande passante recommandée à tous les FSs. La valeur maximale du paramètre ρ est donnée par l'équation suivante :

$$\rho_{max} = \frac{C_{min}}{\sum_{i=1}^{n} B_{rec}^{i}}, \tag{4.15}$$

où C_{min} est le minimum des capacités des liens physiques.
Lorsque $w^i(t) \geq \rho\, p^i$, ceci signifie que FS^i est prêt à payer une pénalité. Ainsi, la meilleure stratégie est de demander une bande passante égale à son attente.
Lorsque $w^i(t) < \rho\, p^i$ chaque fournisseur de services FS^i a seulement une information sur sa bande passante recommandée B_{rec}^{i} . Ainsi, deux cas sont applicables :

— Le premier cas se produit lorsque $B_{esp} \leq B_{rec}$, la meilleure stratégie pour le FS est de demander une bande passante supérieure ou égale à la valeur de sa bande passante espérée $B_{req}^{i} = B_{esp}^{i}$.

— Le deuxième cas apparait lorsque $B_{esp} > B_{rec}$, le FS^i tente de maximiser son gain. On établit alors :

$$Max\ \ ([\, w^i(t) - \frac{B_{all}^{i}}{B_{req}^{i}}\, p^i\,]\, B_{all}^{i}. \tag{4.16}$$

L'allocation de la bande passante s'écrit :

$$B_{all}^{i*} = \frac{w^i(t) * B_{rec}^{i}}{2\, p^i}. \tag{4.17}$$

La meilleure stratégie à jouer dans le jeu s'exprime comme suit :

$$B_{req}^{i*} = Min\,(\, \frac{w^i(t) * B_{rec}^{i}}{2\, p^i}\, ,\, B_{esp}^{i}\,). \tag{4.18}$$

L'équilibre de Nash est un concept fondamental pour prédire le résultat d'une interaction stratégique entre les FS. Dans notre jeu, une allocation de bande passante est appelée équilibre de Nash si et seulement si aucun FS$_i$ ne peut obtenir un gain supérieur en changeant de stratégie dans le jeu. En d'autres termes, une allocation de la bande passante$A(\ B_{all}^{i*}\ ,\ B_{all}^{-i*}\ ))$ est un équilibre de Nash si pour chaque FS la relation suivante est vérifiée : $u^{i*}(B_{all}^{i*}) \geq u^{i*}(B_{req}^{i*})$.

On distingue les deux situations suivantes :

— La première situation correspond au cas où tous les fournisseurs de services vérifient $B_{esp}^{i} < B_{rec}^{i} < B_{dispo}$. Il existe un équilibre de Nash A(B_{req}^{i})=B_{esp}^{i} .

— Quand la bande passante est limitée et au moins un FS^i a $B_{esp}^{i} > B_{rec}^{i}$. Dans ce cas le FIP ne peut pas satisfaire les demandes de tous les FS.

Selon l'équilibre de Nash nous proposons la classification suivante des fournisseurs de services :

— Les FSs conservateurs sont les fournisseurs de services dont la bande passante espérée vérifie $B_{esp}^{i} \leq B_{rec}^{i}$.

— Le FSs ordinaires sont les fournisseurs de services dont la bande passante espérée vérifie $B^i_{rec} < B^i_{esp} \leq \rho\ B^i_{rec}$.

— Les FSs avides sont les fournisseurs de services dont la bande passante espérée vérifie $B^i_{esp} > \rho B^i_{rec}$.

Afin de montrer l'existence de l'équilibre de Nash quelques hypothèses sont nécessaires :

— Le FIP alloue la bande passante dans l'ordre aux FSs conservateurs, puis aux FSs avides et finalement aux FSs ordinaires.

— $B^i_{rec} \leq B_{dispo}$ pour toutes les FSs.

Dans ces deux hypothèses, il y a trois étapes à suivre pour allouer la bande passante à tous les FS :

1. allouer la bande passante aux FSs conservateurs.
2. allouer la bande passante aux FSs avides.
3. allouer la bande passante aux FSs ordinaires.

Dans notre jeu d'allocation de la bande passante, l'état d'équilibre de Nash dépend de la charge de travail et de la répartition de la bande passante du lien physique entre les différents FSs.

3.3.2 Évaluation du jeu d'allocation du lien physique

Dans cette section, nous pressentons les résultats de la performance du mécanisme d'allocation du lien physique, en termes de bande passante recommandée, de la bande passante demandée, et de la bande passante allouée. Nous discutons également l'équité de cette approche.
Nous supposons que le FIP a déjà accepté les requêtes de 60 FSs qui veulent partager un lien physique d'une capacité de bande passante de 10 Gbps. Nous supposons que le FIP pilote l'utilisation du lien et recommande une valeur $B^i_{rec} = 100$ Mbps et $\rho = 1{,}5$. La disposition à payer $w^i(t)$ et le prix p^i pour chaque FIP sont uniformément distribués. Les 20 premiers FSs sont considérés comme conservateurs. Leurs B^i_{esp} est inférieure ou égale à B^i_{rec}. Ils ne demanderont pas plus de la bande passante parce qu'ils ont atteint le maximum de qualité pour leurs services. Les 20 autres fournisseurs de services sont avides. Leurs bandes passantes espérées B^i_{esp} sont supérieures à $\rho\ B^i_{rec}$. Le reste des fournisseurs de services sont des FSs ordinaires et leurs bandes passantes espérées sont $B^i_{rec} < B^i_{esp} < \rho\, B^i_{rec}$.

La figure 4.7 illustre la bande passante recommandée, espérée et demandée par tous les fournisseurs de services. La figure 4.8 montre la répartition uniforme de la disposition à payer $w^i(t)$ et le prix p^i pour chaque FS. Les FSs conservateurs demandent $B^i_{req} = B^i_{esp}$ et la bande passante allouée est inférieure à la valeur de la bande passante recommandée par le fournisseur d'infrastructure.

Selon la figure 4.7 et la figure 4.8, certains fournisseurs de services ayant une grande disposition à payer $(w^i(t) \geq \rho\ p^i)$ soumettent $B^i_{req} = B^i_{esp}$. Les autres FSs ayant $(w^i(t) < \rho\, p^i)$ tentent de maximiser leurs gains et de faire une demande de bande passante plus faible que celle espérée.

Leur demande s'exprime comme suit :

$$B^i_{req} = Min\,(\,\frac{w^i(t)\ *B^i_{rec}}{2\,p^i}\,,\,B^i_{esp}\,). \tag{4.19}$$

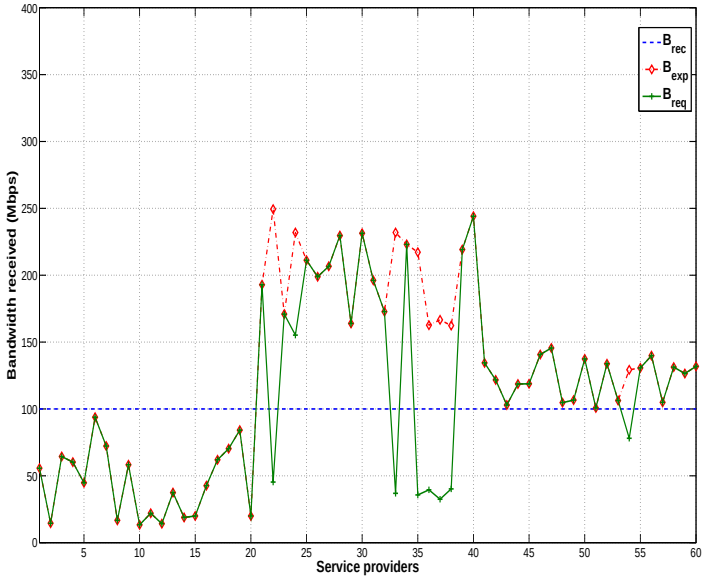


FIGURE 4.7 – Evolution au cours du temps de la bande passante recommandée, requise et espérée pour chaque FS

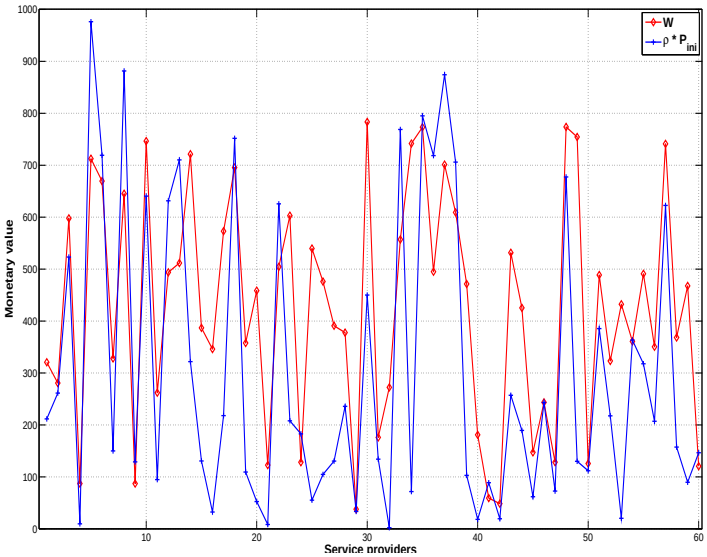


FIGURE 4.8 – Répartition de la disposition à payer au cours du temps

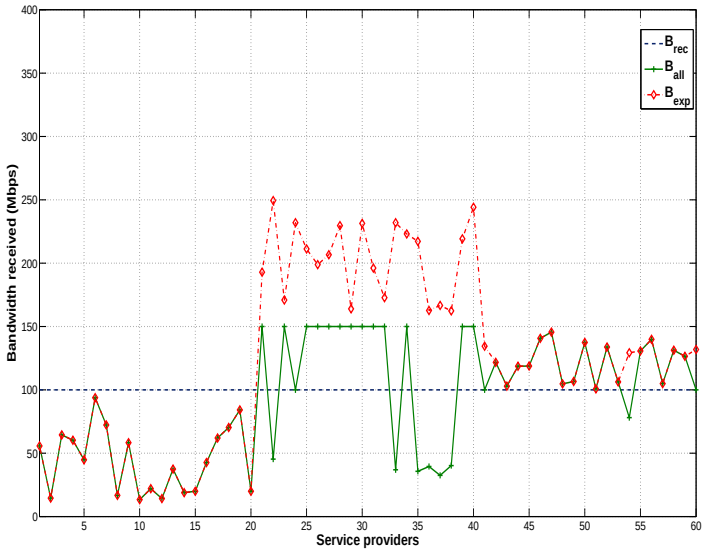


FIGURE 4.9 – Bande passante recommandée, allouée et espérée pour chaque FS

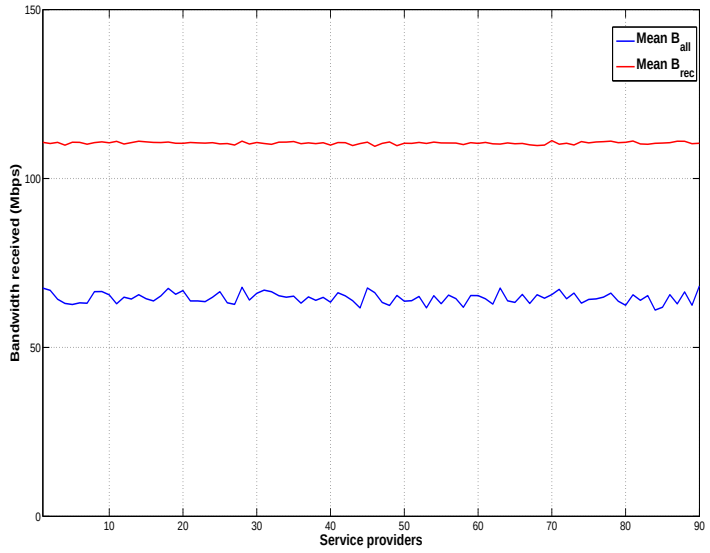


FIGURE 4.10 – Moyenne de la bande passante allouée pour chaque FS

La figure 4.9 illustre les valeurs de la bande recommandée, allouée et espérée pour tous les FS. Les FSs conservateurs ont reçu une bande passante égale à :

$$B^i_{all} = Min(B^i_{req}, B_{dispo}). \tag{4.20}$$

Lorsque les FSs ordinaires ont $w^i(t) < \rho\, p^i$, la bande passante allouée est $B^i_{all} = Min\,(B^i_{req}, B_{dispo}, B^i_{rec})$.

Un FS avide qui a $w^i(t) \geq \rho\, p^i$, reçoit $B^i_{all} = Min\ (B^i_{esp}, B_{dispo}, \rho\, B^i_{rec})$. Comme le montre la figure 4.8 et la figure 4.9 certains FSs peuvent se permettre de payer des coûts élevés et leurs demandes de bande passante sont limitées par ρ afin d'éviter le déficit au niveau d'autres FSs.

L'équité est un critère de performance important lors de la répartition de la bande passante entre plusieurs FSs. Afin de discuter l'équité dans le schéma proposé d'allocation, nous simulons le jeu 1000 fois avec 90 différents FSs partageant un lien physique d'une capacité de 10 Gbps géré par un seul FIP. Ce dernier distribue uniformément la bande passante recommandée pour chaque FS. Nous représentons les résultats de la moyenne sur la figure 4.10.
Comme le montre cette figure, la moyenne de la bande passante allouée au cours du temps est quasiment la même pour tous les FS. Ainsi, nous pouvons conclure que le schéma d'allocation proposé apporte une équité et une efficacité notable lors de l'allocation de la bande passante entre les différents fournisseurs de services.

4 Conclusion

Dans ce chapitre nous avons proposé un nouveau mécanisme pour l'allocation du réseau physique entre les FSs et les FIPs. Ce mécanisme se base sur la théorie des jeux non-coopératifs pour analyser l'interaction entre les FSs et FIP. Le but est d'allouer dynamiquement les ressources physiques entre les deux entités. Tout le long de ce travail, nous avons étudié le comportement égoïste de chaque joueur qui essaie, à tout instant, de maximiser sa fonction d'utilité pour améliorer ses performances sans se soucier des autres joueurs. Ce travail modélise l'allocation des ressources comme un ensemble de jeux compétitifs.
Nous avons montré que l'état d'équilibre de Nash est atteint à chaque étape où ni le FS ni le FIP n'est appelé à modifier sa stratégie. Notre approche permet d'éviter la congestion du nœud et du lien physique sans détériorer les performances des réseaux virtuels en cours d'exécution. Nous avons montré qu'une telle approche donne un meilleur partage des ressources comparé aux solutions statiques.
L'introduction d'une valeur monétaire lors du partage permet de pousser les FSs à faire des demandes plus rationnelles et éviter la sous-utilisation des nœuds et des liens. Une des limitations de notre modèle est la formulation de la requête de partage d'un FS à partir de ses besoins en QoS. Dans le chapitre suivant, nous allons chercher à remédier à ces limitations en proposant un modèle adaptatif linéaire qui capture la corrélation entre la bande passante allouée et les délais de transit des paquets d'un VN sur le lien physique.

Chapitre 5

Allocation dynamique des ressources avec la théorie du contrôle

Sommaire

1 Introduction **89**
2 Le contrôle des systèmes dynamiques linéaires **90**
2.1 Le contrôle en boucle ouverte 90
2.2 Le contrôle en boucle fermée 91
2.3 Le concept de stabilité 91
3 Le contrôle adaptatif de l'allocation dynamique de la bande passante **92**
3.1 Modélisation de l'approche 92
3.2 Contrôleur du fournisseur de services 94
3.2.1 L'estimateur 96
3.2.2 Le demandeur 97
3.3 Contrôleur du fournisseur de l'infrastructure physique 97
3.3.1 Algorithme d'allocation de la bande passante 98
3.3.2 Exemple d'allocation de la bande passante par le FIP 100
3.4 Evaluation de l'approche d'allocation dynamique de la bande passante 101
4 Conclusion **106**

1 Introduction

Le problème crucial de la virtualisation de réseau est l'allocation des ressources de l'infrastructure physique, entre plusieurs réseaux logiques lors du partage. Il est nécessaire de surveiller les réseaux virtuels en cours d'exécution afin de permettre un changement adaptatif dans l'allocation des ressources.

Dans ce chapitre, nous présentons un système de commande à deux niveaux qui s'adapte à l'évolution dynamique de la charge (*workload*) de chaque VN. Le système prédit la meilleure demande pour chaque VN. Cette demande dépend de l'estimation de la relation entre la performance d'un réseau virtuel en matière de débit et des allocations, actuelles et passées, du

lien physique. Notre modèle offre un contrôle adaptatif de l'allocation de bande passante afin de maintenir la QoS offerte à chaque VN à un niveau souhaité en réponse à la dynamique de la charge. Tout le long de ce chapitre, nous supposons que le fournisseur de services déploie, sur chaque réseau virtuel, un seul service comme la VoD ou la VoIP. Dans ce qui suit, nous allons présenter brièvement un rappel sur la théorie de contrôle. Ensuite, nous détaillions la modélisation du problème d'allocation dynamique et nous évaluons ses performances.

2 Le contrôle des systèmes dynamiques linéaires

Le contrôle d'un système dynamique consiste à ramener la variable de sortie vers une valeur désirée. Un système linéaire dynamique, continu et contrôlé, est un système différentiel de la forme suivante :

$$\dot{x}(t) = A(t)\,x(t) + B(t)\,u(t), \tag{5.1}$$

$$y(t) = Cx\,(t) + D(t)\,u(t). \tag{5.2}$$

$x(t)$ est une fonction linéaire d'un espace vectoriel de fonctions m de dimension n sur laquelle il est possible d'effectuer les opérations du calcul différentiel et intégral. L'espace de contrôle $u(.)$ appartient à un ensemble de contrôle admissibles U qui définit le comportement du système.
$A(t)$ est la matrice d'état de dimension $(n * n)$ et $B(t)$ la matrice de commande de dimension $(n * m)$. $C(t)$ est la matrice d'observation de dimension $(l * n)$ et $D(t)$ est la matrice de transfert de dimension $(l * m)$. Le temps t est soit discret soit continu ($t \in T$, $T = \mathbb{R}$ ou $T = \mathbb{Z}$).
Les équations 5.1 et 5.2 sont appelées respectivement équation d'état et équation de sortie. La solution de l'équation d'état s'écrit de la façon suivante :

$$x(t) = e^{A^{(t-t_0)}} x(t_0) + \int_{t_o}^{t} e^{A^{(t-\tau)}}\, B\, u(\tau) d\tau, \tag{5.3}$$

où $x(t)$ est l'état à l'instant t, le premier terme de l'équation est le régime libre ($u(t) = 0$) et le second est la contribution de $u(t)$. On peut classer le contrôle des systèmes dynamiques linéaires en deux catégories :

— le contrôle en boucle ouverte qui fonctionne sans prendre en compte la réponse du système.

— le contrôle en boucle fermée intégrant la réaction du système en contrôlant sa sortie.

2.1 Le contrôle en boucle ouverte

Un contrôle est en boucle ouverte lorsque la commande est élaborée sans la connaissance du résultat à la sortie.
La figure 5.1 représente un système contrôlé en boucle ouverte. À titre d'exemple, un four électrique peut être considéré comme un système dynamique. L'entrée de ce système est un courant électrique d'intensité et de tension données et la sortie est la température de ce four.

FIGURE 5.1 – Représentation d'un contrôle à boucle ouverte

2.2 Le contrôle en boucle fermée

Un contrôle est en boucle fermée lorsque la réaction du système est prise en compte.
La figure 5.2 montre l'exemple d'un système contrôlé en boucle fermée. On spécifie d'abord la température de consigne. Ensuite, quand le système chauffe, le régulateur de température fournit une rétroaction au système par le biais d'un thermocouple afin de couper ou d'actionner la consommation du courant électrique. Finalement, l'état d'équilibre est réalisé lorsque la température de consigne est atteinte.

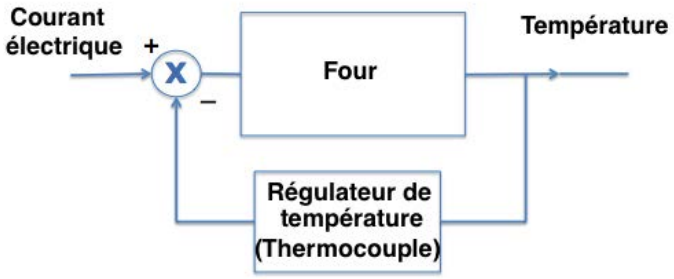


FIGURE 5.2 – Représentation d'un contrôle à boucle fermée

2.3 Le concept de stabilité

La stabilité est une notion fondamentale dans la théorie du contrôle. Elle constitue l'un des aspects essentiels dans l'étude des systèmes dynamiques. On dit qu'un système est stable si et seulement si, écarté de sa position d'équilibre, il tend à y revenir. Une petite perturbation des conditions initiales d'un système implique une petite perturbation de sa trajectoire. Un système n'est dit stable que si en réponse à une entrée bornée, la sortie du système est bornée.
Dans le cas d'un système linéaire dynamique, la stabilité est conditionnée par la matrice de transition d'état. On peut étudier la stabilité d'un système linéaire, en analysant la stabilité de l'origine du système libre, donnée par l'équation suivante :

$$\dot{x}(t) = A(t)\, x(t). \tag{5.4}$$

Ce système converge si et seulement si les valeurs propres de la matrice A sont à partie réelle strictement négative.

3 Le contrôle adaptatif de l'allocation dynamique de la bande passante

Dans ce travail, nous considérons un environnement de réseau virtuel où un FS déploie plusieurs VNs avec plusieurs exigences en termes de capacité de liens et de demande en matière de qualité de service. Chaque VN cherche à recevoir une portion de chaque lien physique géré par un FIP. Nous présentons un système de contrôleur à deux niveaux qui s'adapte à la charge dynamique de chaque flux réseau. Le système utilise une approche prédictive afin de trouver la meilleure demande pour chaque VN. Cette demande dépend de l'estimation de la relation entre le rendement d'un VN en termes de débit utile et les allocations de bande passante actuelles et passées. Ensuite, en raison de la contrainte de capacité de la liaison physique, le système ajuste la portion de la bande passante offerte pour chacun d'eux.
Dans ce travail, nous nous intéressons aux débits utiles de chaque VN sur chaque lien physique. Contrairement au débit nominal qui mesure la vitesse de transmission du lien physique, le débit utile mesure la quantité d'informations qui peut être véhiculée par unité de temps. L'objectif de ce travail est de minimiser les délais d'attente à l'interface de sortie du routeur lorsque le débit d'arrivée à ces derniers, sur un lien physique, dépasse la capacité de ce lien.
Notre modèle offre un contrôle adaptatif de l'allocation de bande passante et cherche à maintenir la QoS offerte à chaque VN. Notre mécanisme fournit une allocation des liens physiques en distribuant la bande passante périodiquement. Il offre également la possibilité d'ajuster les paramètres de chaque VN pour prendre en compte le comportement du réseau actuel afin d'éviter les goulots d'étranglement.

3.1 Modélisation de l'approche

Dans cette partie, nous présentons une approche pour partager efficacement et équitablement la bande passante entre plusieurs VNs et qui permet d'éviter la congestion d'un lien physique. Nous proposons une architecture à deux niveaux semblable à celle proposée par Padala et al. [Padala et al., 2009] pour l'allocation des serveurs pour les applications dans un *data center*. Nous adoptons leur approche et nous l'améliorons pour le problème de l'allocation dynamique de la bande passante dans la virtualisation des réseaux. Le réseau physique est modélisé comme un graphe non orienté, noté par $G = \{N, L\}$, où N est l'ensemble des nœuds physiques et L est l'ensemble des liens physiques. Chaque lien $l(i, j) \in L$ est défini entre deux nœuds i et j avec une capacité C^l. Le but de ce travail est d'éviter la congestion du lien physique et d'améliorer l'équité lors du partage de la bande passante lorsqu'un FS déploie un service sur chaque VN. À titre d'exemple, un FS qui déploie un service de VoIP ne peut pas tolérer un délai de bout en bout supérieur à 150ms d'après la recommandation G.114 de l'UIT [Janssen et al., 2002]. Ensuite, il spécifie le délai ciblé sur chaque lien. À partir de ce délai, le système calcule le débit ciblé. La solution présentée calcule la portion de la bande passante demandée de chaque lien physique répondant aux exigences de ce FS en termes de délai de bout en bout. Lorsque la somme de ces demandes est inférieure à la capacité du lien, le FIP satisfait chaque SP. Sinon, le mécanisme utilise le modèle des courbes *loss-load* [Williamson, 1996] pour partager équitablement la capa-

cité de chaque lien entre plusieurs VN, lorsque la somme de toutes les demandes est supérieure à la capacité du lien. Cet algorithme modélise, pour chaque VN, la relation entre la charge et le niveau de perte de paquets de chaque lien. Il agit comme un mécanisme de rétroaction pour le contrôle de congestion.

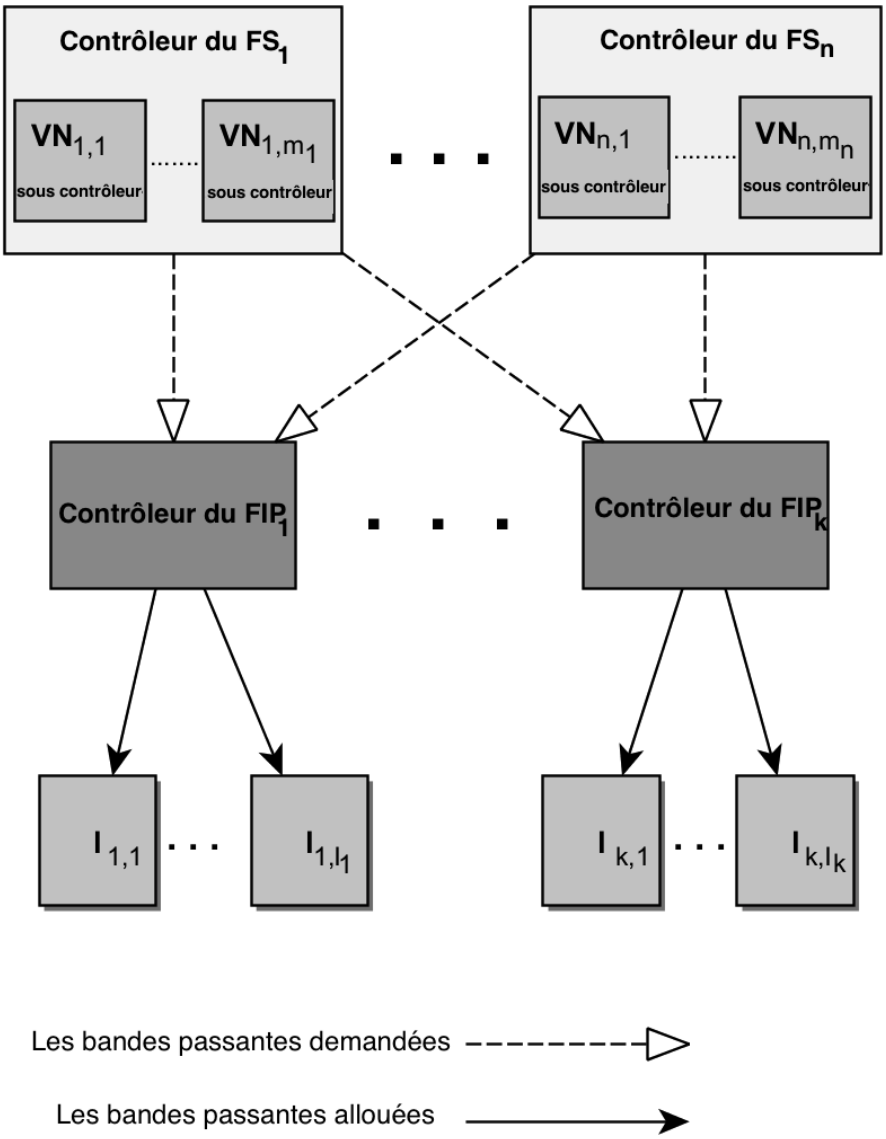


FIGURE 5.3 – Architecture proposée du système de contrôleurs

Dans la figure 5.3, nous présentons une architecture formée de plusieurs contrôleurs. Le modèle proposé vise à minimiser les délais des paquets, lorsque le débit d'arrivée de ces derniers sur un lien physique partagé dépasse la capacité de ce lien. Nous proposons une approche fondée sur la théorie de contrôle et nous appelons chaque composante du système un contrôleur. Il est important de signaler à ce niveau que ces contrôleurs n'ont aucune relation avec ceux d'OpenFlow.

L'architecture comprend un ensemble de contrôleurs pour les FSs et les FIPs. Le contrôleur du FS est composé d'un ensemble de sous-contrôleurs VN qui surveille la charge dynamique de chaque VN. À chaque intervalle de contrôle, les sous-contrôleurs des VNs calculent la portion de la bande passante nécessaire pour atteindre un rendement espéré par le FS.
Notre approche peut être déployée sur une infrastructure réseau virtualisée compatible avec OpenFlow. À chaque nœud physique est attaché un contrôleur OpenFlow qui définit le comportement de ce nœud pour chaque VN. Les fournisseurs de services utilisent les compteurs OpenFlow afin de collecter des informations sur le volume du trafic de chaque VN traversant chaque nœud. À chaque intervalle de contrôle, le FS calcule le débit observé sur chaque tronçon du réseau virtuel. À partir des caractéristiques de performance telles que la latence requise pour le bon fonctionnement du service déployé sur le VN, le FS utilise notre solution afin d'identifier la meilleure demande de bande passante pour chaque lien physique utilisé. Le contrôleur du FIP recueille les demandes de capacité pour de multiples FSs et détermine, selon le modèle des courbes *loss-load* [Williamson, 1996], la meilleure répartition de la capacité de chaque lien physique. Ces contrôleurs visent à affecter toute la bande passante du lien aux FS en tenant compte de la contrainte de capacité de chaque lien. Pour des raisons de clarté, les notations utilisées dans cette partie sont regroupées dans le tableau 5.1.

3.2 Contrôleur du fournisseur de services

Le fournisseur de services attribue un sous-contrôleur à chacun des VNs déployés afin de calculer les demandes des portions de bande passante qui maintiennent les délais de transit et d'attente des paquets fixes entre deux nœuds physiques. Ce sous-contrôleur est responsable de la surveillance du rendement et des allocations passées pour aboutir à la meilleure demande. Les contrôleurs des FSs collectent toutes les potions de bande passante demandées et soumettent un vecteur de demande à chaque FIP. Le sous-contrôleur du VN est composé de deux modules : l'estimateur et le demandeur. L'estimateur met à jour la relation entre la portion de la bande passante d'un lien alloué et le rendement en termes de débit de chaque VN. Le demandeur est responsable de la prédiction de la portion de la bande passante à demander pour chaque $\mathrm{VN}_{i,j}$ afin d'atteindre les délais ciblés sur la base du résultat de l'estimation. Nous détaillons dans ce qui suit les fonctionnalités de l'estimateur et du demandeur.
Dans ce travail, nous partons de la même supposition que Padala et al. [Padala et al., 2009] concernant le comportement du système. Les auteurs supposent que ce comportement peut être caractérisé localement par un modèle linéaire. Ils ont effectué plusieurs expérimentations pour confirmer cette hypothèse. Dans notre approche, nous supposons que le comportement de chaque VN peut être approximé localement par un modèle linéaire. Nous réestimons périodiquement ce modèle linéaire à partir des mesures en temps réel comme le débit offert à chaque VN sur chaque lien physique. L'objectif est de permettre au modèle de s'adapter à différents régimes de fonctionnement et aux conditions de charge de travail de chaque VN. Notre approche suppose que les variations drastiques des charges du trafic de chaque VN se produisent rarement durant un intervalle de contrôle, permettant ainsi à notre estimation de converger localement.
Ce modèle tient compte de la relation entre les allocations et les performances en termes de débits

Symbol	Description
S	Ensemble des fournisseurs de services FS_n
I	Ensemble de fournisseurs de l'infrastructure physique FIP_k
V	Ensemble de réseaux virtuels $V_{i,j}$ déployés par le fournisseur de services $s \in S$
L	Ensemble de liens physiques $l_{k,l}$ gérés par le fournisseur de l'infrastructure physique $\text{FIP}_k \in I$
$VN_{i,j}$	Le réseau virtuel j déployé par le fournisseur de services FS_i
T	Intervalle de contrôle
$x(T)$	La valeur de x dans l'intervalle T
$Breq_{s,v,l}$	La portion de la capacité du lien $l \in L$ demandée par le réseau virtuel $v \in V$ déployé par le fournisseur de services $s \in S$
$Ball_{s,v,l}$	La portion de la capacité du lien $l \in L$ allouée au réseau virtuel $v \in V$ déployé par le fournisseur de services $s \in S$
$d_{s,v,l}$	Le délai d'attente et de transit des paquets mesuré d'un réseau virtuel $v \in V$ déployé par $s \in S$ sur un lien physique $l \in L$
$\bar{d}_{s,v,l}$	Le délai d'attente et de transit des paquets ciblé d'un réseau virtuel $v \in V$ déployé par $s \in S$ sur un lien physique $l \in L$
$p_{s,v,l}$	Probabilité de perte de paquets d'un réseau virtuel $v \in V$ sur un lien physique $l_{k,l}$
$y_{s,v,l} = 1/d_{s,v,l}(T)$	Le débit mesuré d'un réseau virtuel $v \in V$ sur un lien physique $l \in L$
$\bar{y}_{s,v,l} = 1/\bar{d}_{s,v,l}(T)$	Le débit ciblé d'un réseau virtuel $v \in V$ sur un lien physique $l \in L$
$\hat{y}_{s,v,l} = y_{s,v,l}/\bar{y}_{s,v,l}$	Le débit normalisé de transit des paquets d'un réseau virtuel $V_{i,j}$ sur un lien physique $l \in L$
$\alpha(T)$, $\beta(T)$	Paramètres de calcul des courbes "loss-load"
$k(T)$	Paramètre du comportement de l'algorithme des courbes "loss-load"

TABLE 5.1 – Les différentes notations utilisées dans le système à contrôleurs

utiles des paquets au niveau de chaque $VN_{i,j}$ déployé par FS_j. Chaque FS soumet à chaque FIP un vecteur de demandes de portions de la bande passante $Breq_s$. Ce vecteur représente toutes les demandes de ses VNs déployés sur les différents liens physiques gérés par ce FIP.
La figure 5.4 illustre l'architecture du sous-contrôleur du VN qui comprend l'estimateur et le demandeur.

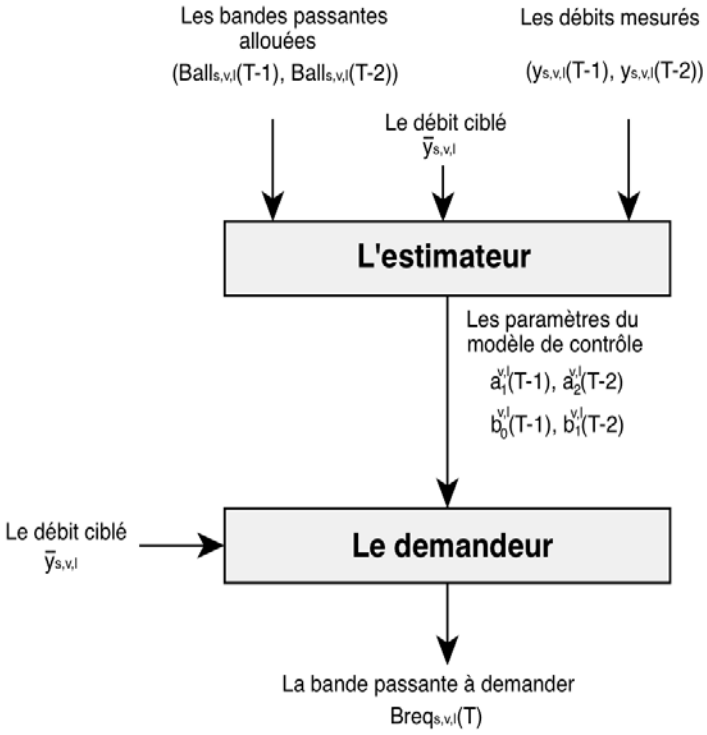


FIGURE 5.4 – Architecture du sous-contrôleur du VN

3.2.1 L'estimateur

A chaque intervalle de contrôle, l'estimateur recalcule un modèle linéaire qui permet de faire une approximation du modèle non linéaire de la relation entre la bande passante offerte pour chaque VN et les débits normalisés. Plus précisément, l'estimateur utilise un modèle autorégressif moyen mobile (*Auto Regressive Moving Average* (ARMA)) pour représenter cette relation.
Les paramètres du modèle sont mis à jour, à chaque intervalle de contrôle T, en utilisant la méthode des moindres carrés récursifs [Lee et al., 1981]. Périodiquement, l'estimateur change de manière adaptative les paramètres en utilisant l'équation suivante :

$$\begin{aligned}\hat{y}_{s,v,l}(T) \;=\; & a_1^{v,l}(T)\,\hat{y}_{s,v,l}(T-1) \;+\; a_2^{v,l}(T)\,\hat{y}_{s,v,l}(T-2) \\ & +\; b_0^{v,l}(T)\,Ball_{s,v,l}(T) \;+\; b_1^{v,l}(T)\,Ball_{s,v,l}(T-1),\end{aligned} \tag{5.5}$$

où $\hat{y}_{s,v,l}(T)$ et $Ball_{s,v,l}(T)$ désignent respectivement les performances en termes de débits

utiles du VN et la portion de la bande passante allouée pendant l'intervalle T. Les paramètres $a_1^{v,l}(T)$ et $a_2^{v,l}(T)$ capturent la corrélation entre les performances passées et actuelles du VN sur le lien physique l . Les paramètres $b_0^{v,l}(T)$ et $b_1^{v,l}(T)$ sont deux vecteurs qui capturent la corrélation entre les allocations passées et actuelles du lien l. Ces paramètres d'adaptation dépendent de l'intervalle de contrôle. En raison de la nature récursive de l'algorithme des moindres carrés, le temps de calcul de ces paramètres est considérablement réduit. Nous utilisons une performance normalisée plutôt qu'une performance absolue afin d'assurer la stabilité de notre système. Les portions de la capacité d'un lien physique, demandées et allouées, ont une valeur comprise entre 0 et 1.

3.2.2 Le demandeur

Une fois que le demandeur reçoit l'estimation des paramètres du modèle linéaire, son rôle est de prédire la portion de la capacité de chaque lien physique nécessaire pour atteindre le délai ciblé des paquets sur ce lien pour $\mathrm{VN}_{i,j}$ déployé par le fournisseur de services FS_j. Le demandeur vise à trouver les demandes de bande passante pour atteindre ses objectifs en termes de délais de bout en bout. Les auteurs de [Padala et al., 2009] ont montré, à partir de plusieurs expérimentations, que le deuxième ordre du modèle ARMA utilisé par l'estimateur permet de prédire la performance des applications avec une bonne précision.
Dans ce travail, nous supposons que l'allocation de la bande passante pour un VN durant les deux derniers intervalles de contrôle permet de prédire les performances de ce VN à chaque nouvel intervalle de contrôle. En effet, on peut prédire la valeur de $Breq_{s,w,l}(T)$ à partir du modèle auto-régressif moyen mobile que l'estimateur a recalculé et a mis à jour ses différents paramètres. Cette prédiction se base sur la corrélation au cours du temps de la relation passée et actuelle entre les débits utiles du VN et l'allocation de la bande passante.
La meilleure demande de la bande passante sur un lien physique l pour un réseau virtuel v, déployé par un fournisseur de services s, est donnée à partir de l'équation 5.5 comme suit :

$$\begin{aligned} Breq_{s,v,l}(T) \;=\; & [\hat{y}_{s,v,l}(T) \;-\; a_1^{v,l}(T)\,\hat{y}_{s,v,l}(T-1) \;-\; a_2^{v,l}(T) \\ & \hat{y}_{s,v,l}(T-2) \;-\; b_1^{v,l}(T)\,Ball_{s,v,l}(T-1)]\,/\,b_0^{v,l}(T). \end{aligned} \tag{5.6}$$

Les paramètres a_1, a_2, b_0 et b_1 ont de nouvelles valeurs à chaque intervalle de contrôle T évaluées à l'aide de la méthode des moindres carrés récursifs. Cet algorithme permet de corriger l'estimation de ces paramètres en fonction des résultats mesurés à chaque intervalle de contrôle T. Ceci est fait en calculant le filtre qui minimise la somme pondérée des carrés des erreurs de prédiction.

3.3 Contrôleur du fournisseur de l'infrastructure physique

Dans cette partie, nous présentons l'algorithme d'allocation utilisé par le contrôleur FIP afin d'allouer une partie de la capacité d'un lien physique à plusieurs VNs. Le contrôleur du FIP reçoit les demandes d'allocation de plusieurs contrôleurs des FSs. Ensuite, il découpe les portions de bande passante demandées par chaque VN par rapport à la capacité de chaque

lien physique. Le contrôleur utilise un algorithme d'allocation de bande passante pour trouver la meilleure répartition de la capacité de chaque lien, entre les différents VNs, en utilisant le modèle des courbes *loss-load*. Cette approche se traduit par la relation mathématique entre la charge offerte par chaque VN et le niveau de la perte de paquets. Il offre un bon mécanisme de rétroaction pour le partage de la capacité de liaison disponible. Le contrôleur agit comme un mécanisme pour contrôler les débits. La courbe *loss-load* exprime la probabilité de perte de paquets en fonction de la bande passante offerte à un moment donné et d'un état de charge dans le réseau. Ce contrôle punit les VNs avides en leur offrant moins de bande passante que celle reçue lorsqu'ils sont moins gourmands.

3.3.1 Algorithme d'allocation de la bande passante

Supposons que m réseaux virtuels sont déployés par n fournisseurs de services qui partagent l liens physiques gérés par un seul FIP. Le contrôleur FIP reçoit les requêtes des VNs qui demandent une portion de la capacité du lien physique. Le modèle des courbes *loss-load* évalue la probabilité de perte de paquets pour le m^{ème} réseau virtuel en fonction de la charge de trafic généré par l'autre m-1ème réseau virtuel. Pour chaque lien physique l, le FIP calcule la portion totale de la bande passante demandée par tous les VNs qui partagent ce lien à partir de l'équation suivante :

$$R_l(T) = \sum_{s=1}^{N} \sum_{v=1}^{m_s} Breq_{s,v,l}(T). \tag{5.7}$$

Si $R_l(T) \leq 1$, alors chaque VN reçoit une allocation de la bande passante égale à la portion demandée, sinon, le contrôleur calcule deux paramètres $\alpha(T)$ et $\beta(T)$ à partir des expressions suivantes :

$$\begin{cases} \alpha(T) = R_l(T) - 1, \\ \beta(T) = \sum_{s=1}^{N} \sum_{v=1}^{m_s} Breq_{s,v,l}^{k(T)+1}. \end{cases} \tag{5.8}$$

$k(T)$ définit le comportement du modèle des courbes *loss-load* pendant l'intervalle de contrôle T. Le paramètre $\alpha(T)$ évalue la charge du trafic excédentaire sur le lien et $\beta(T)$ mesure la répartition de la charge totale parmi les VNs en concurrence.
Les conditions aux limites du modèle des courbes *loss-load*, $Breq_0(T)$ et $Breq_m(T)$ sont définies comme suit :

$$\begin{cases} Breq_0(T) = 0, \\ Breq_m(T) = (\beta(T)/\alpha(T))^{1/k(T)} \ si \ \alpha(T) > \beta(T), \ sinon \ Breq_m(T) = 1. \end{cases} \tag{5.9}$$

Le modèle des courbes *loss-load* a les propriétés suivantes pour attribuer la probabilité de perte de paquets $p_{s,v,l}$ sur un lien physique l :

— Si la somme de toutes les demandes de la bande passante est inférieure ou égale à la capacité disponible, $p_{s,v,l} = 0$ pour tous le VNs.

— Si la somme de toutes les demandes de la bande passante est supérieure à la capacité de la liaison physique, $p_{s,v,l} > 0$ et $p_{s,v,l} = Breq^{k}_{s,v,l}\,(\alpha(T)/\beta(T))$ pour tous les VNs.

— $p_{s,v,l}$ est une fonction non décroissante de la demande de la bande passante.

La probabilité de perte de paquets $P(Breq(T))$ pour le m^{ème} réseau virtuel lors de la demande de la bande passante $Breq(T)$ d'un lien physique d'une capacité C_l est donnée par :

$$\begin{cases} P(Breq(T)) = \dfrac{Breq^{k(T)}(Breq(T) + \alpha(T))}{Breq^{k+1}(T) + \beta(T)} & si\ Breq(T) \leq Breq_m(T), \\ P(Breq(T)) = 1 \qquad\qquad sinon. \end{cases} \tag{5.10}$$

Le modèle des courbes *loss-load* est exprimé en fonction de toutes les demandes reçues de la bande passante et des paramètres de la charge actuelle $\alpha(T)$ et $\beta(T)$.
Le terme $Breq(T) + \alpha(T)$ représente la nouvelle surcharge sur le lien lors de la réception d'une nouvelle demande de la bande passante. Le dénominateur représente la nouvelle mesure de la répartition de la charge $Breq^{k(T)}(T)$. Le modèle des courbes *loss-load* permet au contrôleur de pénaliser les VNs avides lorsque leur demande est très élevée.
Un lien physique est considéré comme critique si, à l'intervalle de commande T, la somme de la demande de bande passante de tous les VNs est supérieure à la capacité disponible C_l.
Pour chaque lien, $k(T)$ a une valeur unique à chaque intervalle T. Ce paramètre détermine le comportement du modèle des courbes *loss-load* pour répartir la capacité d'un lien entre plusieurs VNs. Il est donné par les équations suivantes :

$$\begin{cases} k(T) = 1 \ \ Si \ \ T = 0, \\ k(T) = 2 * k(T-1) \quad si\ R_l(T-1) > 1\ and\ R_l(T) > 1, \\ k(T) = k(T-1) - 1 \quad sinon. \end{cases} \tag{5.11}$$

Afin de trouver la nouvelle valeur du paramètre k à l'instant $T+1$ on utilise la méthode d'augmentation multiplicative et retrait additif (*Multiplicative-Increase/Additive-Decrease* MIAD). Si, à l'instant T, la somme des demandes de la bande passante dépassent la capacité du lien on multiplie sa valeur par deux. Sinon, on décrémente sa valeur de 1 afin de réduire la pénalité infligée aux FSs avides.

Le contrôleur FIP utilise un algorithme basé sur le modèle des courbes *loss-load* pour partager une capacité du lien entre plusieurs FS. En effet, cet algorithme tient compte des états des liaisons physiques lors du partage de la bande passante. Ensuite, il cherche la meilleure répartition de la bande passante. L'algorithme 5.1 présente les différentes étapes de l'attribution de la bande passante pour les VNs.

Algorithme 5.1 : Algorithme d'allocation dynamique de la capacité d'un lien physique entre plusieurs réseaux virtuels

Entrées : Réseau virtuel v : $\mathrm{VN}_{i,j} \in V$
Fournisseur de services $\mathrm{FS}_j \in S$
Fournisseur d'infrastructure physique $\mathrm{FIP}_m \in I$
Lien physique l : $\mathrm{l}_{m,n} \in L$
Intervalle de contrôle T
La bande passante demandée $Breq_{s,v,l}(T)$
Paramètre du comportement du modèle de courbes loss-load $k(T-1)$
Capacité du lien physique C_l

Sorties : La bande passante allouée $Ball_{s,v,l}(T)$

début
  pour tous les *intervalles de contrôle* T **faire**
    pour tous les *liens physiques* $l_{m,n}$ **faire**
      pour tous les *les fournisseurs de services* FS_j **faire**
        pour tous les *réseaux virtuels* $VN_{i,j}$ **faire**
          $R_l(T) \leftarrow \sum_{i=1}^{N} \sum_{j=1}^{m_s} Breq_{s,v,l}(T)$
          si $R_l(T) \leq 1$ **alors**
            $Ball_{s,v,l}(T) \leftarrow Breq_{s,v,l}(T)$
          sinon
            si $R_l(T-1) > 1$ **alors**
              $k(T) \leftarrow 2 * k(T-1)$
            sinon
              $k(T) \leftarrow k(T-1) - 1$
            fin
            $\alpha(T) \leftarrow R_l(T) - 1$ $\beta(T) \leftarrow \sum_{s=1}^{N} \sum_{v=1}^{m_s} Breq_{s,v,l}^{k(T)+1}$
            $p_{s,v,l}(T) \leftarrow Breq_{s,v,l}^{k(T)} * (\alpha(T)/\beta(T))$
            $Ball_{s,v,l}(T) \leftarrow (1 - p_{s,v,l}(T)) * Breq_{s,v,l}(T)$
          fin
        fin
      fin
    fin
  fin
fin

3.3.2 Exemple d'allocation de la bande passante par le FIP

Considérons trois VNs déployés par différents FSs qui partagent un lien physique. La capacité de ce lien est de $C = 100$ Mbps. Les FSs soumettent une demande de bande passante au FIP en charge d'assurer le partage.
La demande de bande passante pour VN_1 est 60 Mbps, celle de VN_2 est de 40 Mbps et celle de VN_3 est 30 Mbps. Puisque les portions de la capacité d'un lien physique demandées ont une valeur comprise entre 0 et 1, la valeur de $Breq_i$ est égale à la demande divisée par la capacité

du lien ($Breq_1 = 0.6$, $Breq_2 = 0.4$ et $Breq_3 = 0.3$).
Le FIP calcule en premier lieu les paramètres $\alpha(T) = R_(T) - 1$ et $\beta(T) = \sum_{i=1}^{3} (Breq_i^{k(T)+1}/C)$.
Pour $k(T) = 2$, $\alpha(T) = 1.3 - 1 = 0.3$ et $\beta(T) = 0.6^3 + 0.4^3 + 0.3^3 = 0.307$.
Pour $k(T) = 7$, $\alpha(T) = 1.3 - 1 = 0.3$ et $\beta(T) = 0.6^8 + 0.4^8 + 0.3^8 = 0.018$.
Ensuite, il calcule la probabilité $p_i(T)$ de perte des paquets pour chaque réseau virtuel à partir de cette équation $p_i(T) = Breq_i^{k(T)} * (\alpha(T)/\beta(T))$.
Enfin, il en déduit la bande passante à allouer pour chacun d'entre eux en calculant $Ball_i(T) = (1 - p_i(T)) * Breq_i(T)$.
Le tableau 5.2 illustre les probabilités de perte des paquets pour chaque VN et la bande passante allouée à eux pour deux valeurs du paramètre $k(T)$ qui définissent le comportement du modèle de courbes "loss-load".

i	$Breq_i(T)$	$k(T) = 2$				$k(T) = 7$			
		$\alpha(T)$	$\beta(T)$	$p_i(T)$	$Ball_i(T)$	$\alpha(T)$	$\beta(T)$	$p_i(T)$	$Ball_i(T)$
1	0.6	0.3	0.307	0.353	0.388 (38.8 Mbps)	0.3	0.018	0.467	0.312 (31.2 Mbps)
2	0.4			0.157	0.338 (33.8 Mbps)			0.028	0.389 (38.9 Mbps)
3	0.3			0.089	0.274 (27.4 Mbps)			0.004	0.299 (29.9 Mbps)

TABLE 5.2 – Exemple de partage de la capacité d'un lien physique entre trois VNs

On remarque d'après le tableau que VN_1 a reçu moins de bande passante que celle demandée. En effet, le contrôleur du FIP considère ce VN comme avide. Il le sanctionne en lui proposant moins de bande passante que celle reçue lorsqu'il est moins gourmand. Les autres VNs reçoivent une capacité du lien physique quasiment identique à celle demandée. Plus la valeur de $k(T)$ est élevée plus sévère est la sanction infligée aux VNs avides.

3.4 Evaluation de l'approche d'allocation dynamique de la bande passante

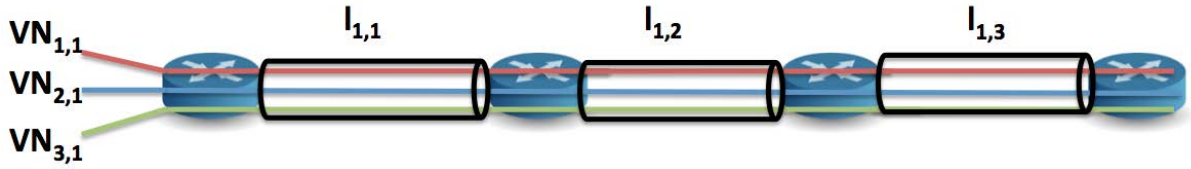


FIGURE 5.5 – Scénario de partage de l'infrastructure physique entre les trois VNs

Dans cette section, nous présentons l'évaluation de l'approche d'allocation de la bande passante proposée. Afin de réduire la complexité de la simulation, on part de la supposition que trois VNs, {$VN_{1,1}$, $VN_{2,1}$, $VN_{3,1}$ } déployés par différents FSs, partageant trois liens physiques

$l_{1,1}, l_{1,2}, l_{1,3}$ géré par un seul fournisseur d'infrastructure physique FIP_1. Les FSs déploient un seul service sur chaque VN. La figure 5.5 illustre le scénario de partage de l'infrastructure physique entre les trois VNs.

Afin de discuter les performances de notre approche, nous nous focalisons sur l'allocation de la capacité d'un seul lien physique. Dans notre travail, l'allocation de chaque lien physique est indépendante de celle des autres liens. Nous supposons que FS adopte les mécanismes proposés dans ce travail pour chaque lien physique. Nous avons capturé une trace simulée avec le simulateur OPNET où trois clients partagent un lien physique de 100 Mbps pour accéder à un service déployé sur un serveur distant. L'ordonnancement des paquets au niveau des nœuds se fait avec l'algorithme WFQ. Les différents services offerts aux clients sont "*Streaming Multimedia*", "*Interactive Multimedia*" et "*Interactive Voice*". Ces services sont identifiés au niveau des nœuds à l'aide du champ TOS de l'entête IP.
Nous avons mené plusieurs simulations pour capturer les délais moyens de chaque service avec différentes allocations de bande passante. A chaque nouvelle simulation, nous changeons les poids de chaque service afin de lui offrir plus ou moins de la bande passante du lien physique. L'objectif de ces simulations est de capturer un fichier de trace contenant l'allocation de la bande passante et le délai moyen des trois services différents partageant le même lien physique. La figure 5.6 illustre le scénario considéré dans OPNET pour capturer la trace simulée.

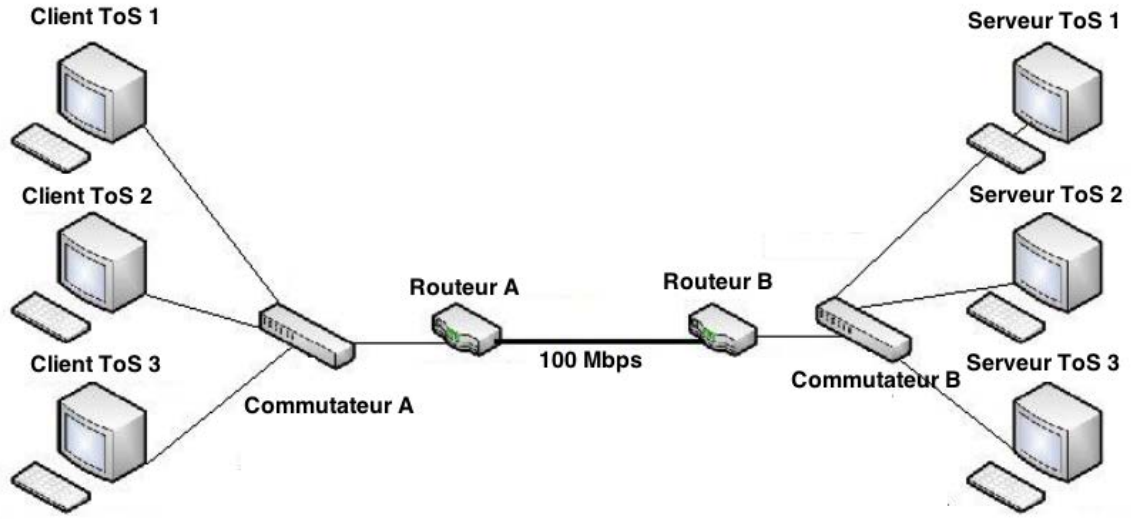


FIGURE 5.6 – Scénario de capture de la trace simulée avec OPNET

Dans notre évaluation, l'intervalle de contrôle T est égal à une seconde. Nous évaluons d'abord notre approche pour trouver la meilleure demande de chaque VN en hors ligne. Les délais ciblés de chaque VN sont maintenus constants à chaque intervalle de contrôle. Lorsque la somme de toutes les demandes dépasse la capacité du lien, nous analysons la répartition de la bande passante offerte par l'algorithme proposé (désigné par FAA). Nous comparons les résultats avec l'algorithme de partage proportionnel (désigné par PSA).

La figure 5.7 illustre des délais mesurés et ciblés du réseau virtuel $\text{VN}_{1,1}$. Les délais mesurés

sont capturés lors de la simulation avec OPNET lors du partage de la bande passante pour les différents services. Les délais ciblés sont définis par le fournisseur de services pour chaque VN déployé et pour chaque lien. En effet, le FS surveille les performances de chaque VN et exprime le délai ciblé sur chaque lien selon le service déployé sur le VN.

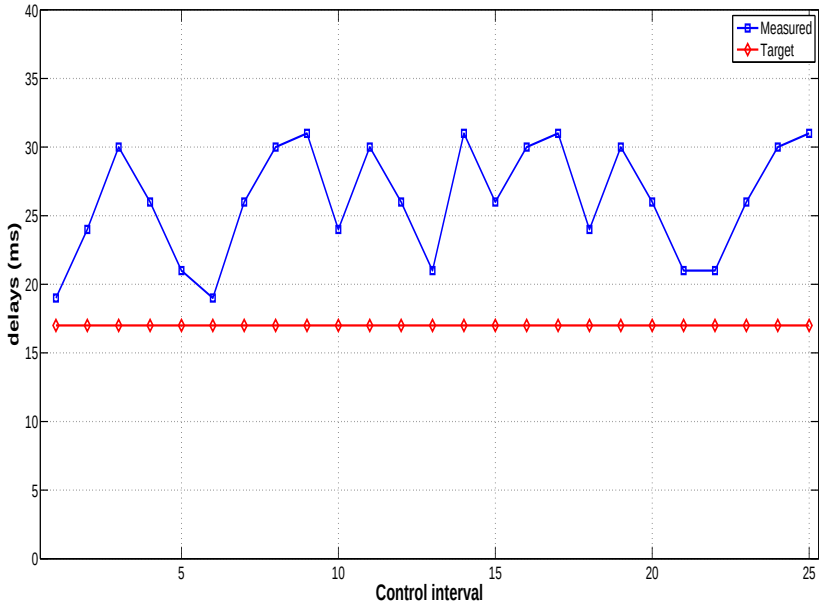


FIGURE 5.7 – Les délais mesurés et ciblés des paquets de $VN_{1,1}$ à chaque intervalle de contrôle

La figure 5.8 montre les différentes demandes de la bande passante par le réseau virtuel $VN_{1,1}$ pour recevoir une portion de la capacité du lien physique $l_{1,1}$. Notre algorithme réagit de manière adéquate à la charge de travail des changements sur la liaison physique. Chaque VN surveille la performance passée en termes de délai de transit et d'attente de paquet et sa répartition en termes de bande passante. La relation entre eux permet de trouver la demande de la portion de la bande passante d'un lien. L'objectif est de maintenir le délai des paquets à 17 ms à chaque intervalle de temps de contrôle. À chaque intervalle de contrôle, la valeur de la bande passante demandée s'adapte aux besoins en termes de délai en fonction des allocations passées.

La figure 5.9 illustre l'évolution de la portion de la capacité demandée du lien physique $l_{1,1}$ et celle allouée aux trois VNs à l'aide de l'algorithme de partage proportionnel (PSA) et de l'algorithme proposé d'allocation (FAA). À chaque intervalle de contrôle, on compare la bande passante allouée pour chaque VN par les deux algorithmes pour un paramètre du comportement de l'algorithme des courbes *loss-load* a une valeur $k = 5$. On remarque que l'algorithme FAA donne de meilleurs résultats lorsqu'un VN demande de grande bande est élevée. Par exemple, à $T = 10$s, le réseau virtuel $VN_{2,1}$ sollicite une portion de la bande passante $Breq$ de 62%. Ce VN est considéré avide puisque $VN_{1,1}$ a fait une requête de 47% et $VN_{3,1}$ une requête $Breq$ de 49 %. Notre algorithme satisfait la demande des VNs non gourmands et punit les autres en leur

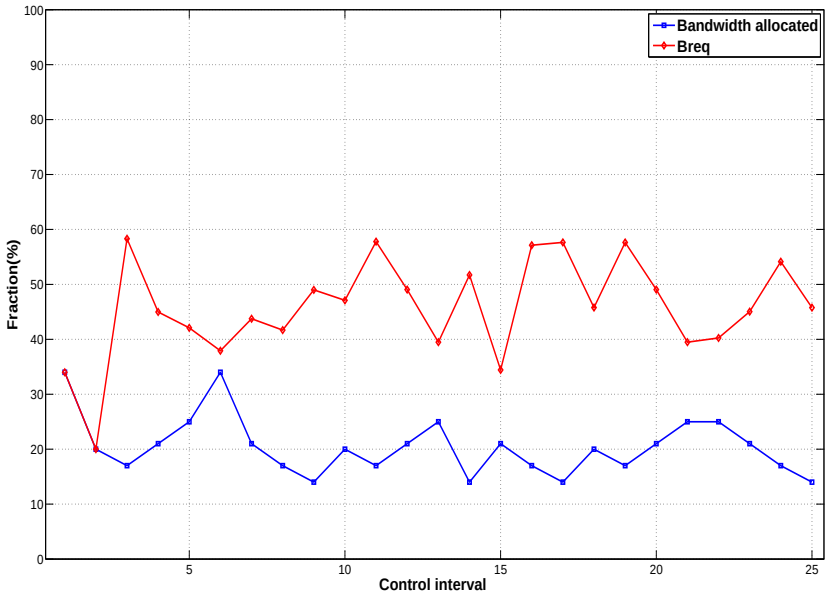


FIGURE 5.8 – Evolution de la portion de la bande passante du lien $l_{1,1}$ demandée par $VN_{1,1}$

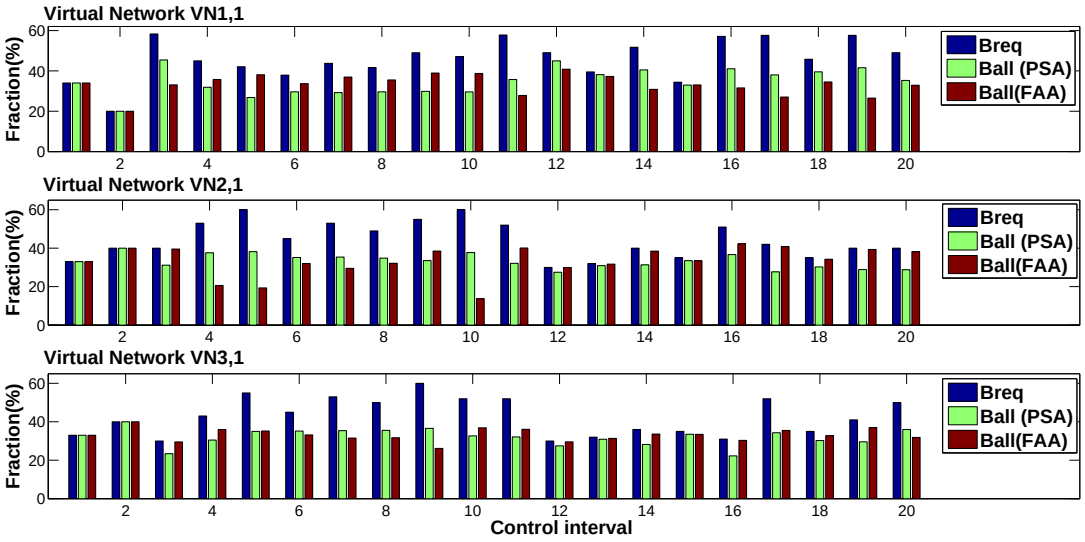


FIGURE 5.9 – Evolution de la bande passante du lien physique $l_{1,1}$ allouée aux trois VNs

donnant moins de bande passante.

La figure 5.10 représente l'évolution de la bande passante demandée et allouée du lien $l_{1,1}$ pour le réseau virtuel $VN_{1,1}$. Grâce à la simulation, nous analysons l'impact de la charge en fonction du paramètre de comportement k sur l'attribution de la bande passante. L'objectif est de discuter l'ampleur des sanctions à infliger au VN avide en termes de bande passante allouée. Par exemple, à $T = 11$s, le réseau virtuel $VN_{1,1}$ est considéré avide.

Lorsque $k = 10$, il reçoit 28 % de la capacité de la liaison physique. Lorsque le paramètre k est égal à 20, le lien physique $l_{1,1}$ est critique, le VN avide reçoit seulement 11 % de la capacité du lien physique. Notre algorithme prévient ce déficit et il satisfait la demande des VNs non avides tout en pénalisant le reste.

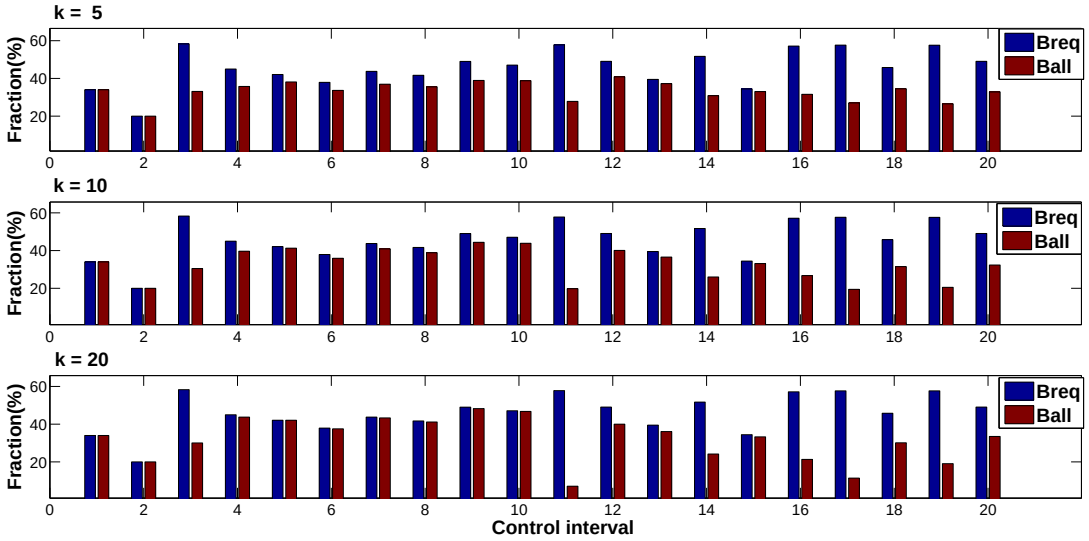


FIGURE 5.10 – L'impact du paramètre k du modèle des courbes *loss-load* dans l'allocation de la bande passante

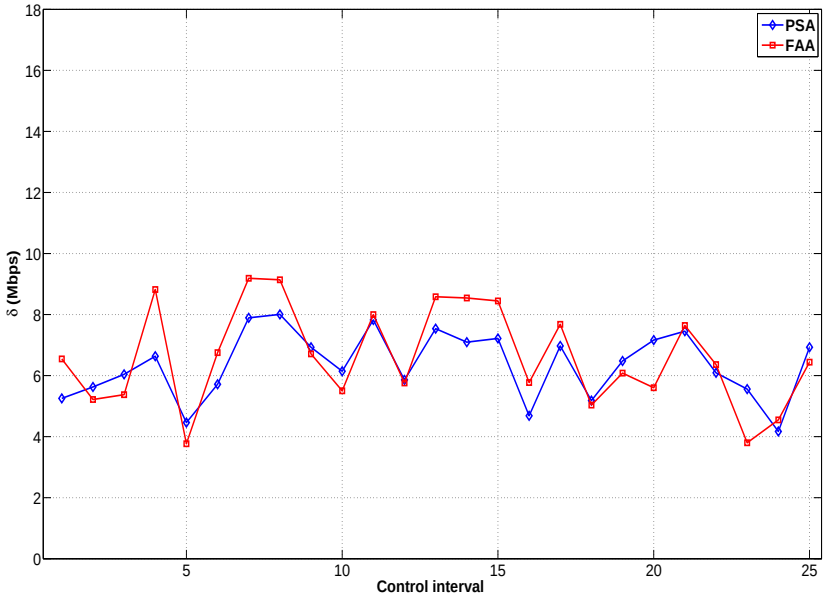


FIGURE 5.11 – La différence entre les portions moyennes de la bande passante demandée et celle allouée en utilisant les algorithmes PSA et FAA

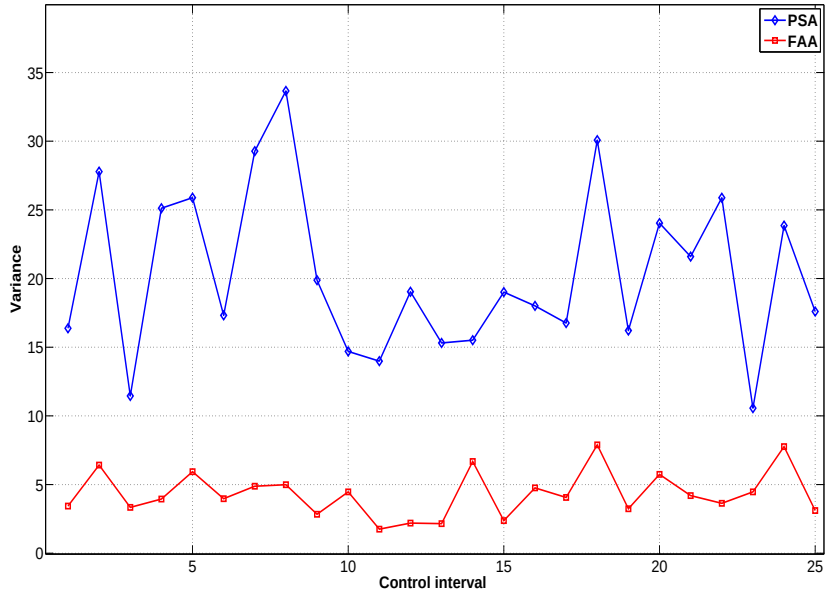


FIGURE 5.12 – Comparaison des variances de chaque ensemble de moyennes des algorithmes PSA et FAA

Afin d'analyser l'équité de l'approche d'allocation proposée, nous avons exécuté l'algorithme 250 fois avec différentes demandes de bande passante pour chaque VN. Nous avons calculé δ, la différence moyenne entre la bande passante demandée et celle allouée aux trois VNs, pendant 10 intervalles de contrôle. Une comparaison est effectuée avec l'algorithme de PSA.
La figure 5.11 illustre les résultats obtenus. En effet, les différences sont quasiment identiques avec ceux de l'algorithme de partage proportionnel. Puisque l'algorithme PSA offre un partage équitable de la bande passante, nous pouvons conclure que notre algorithme offre aussi une répartition équitable de la bande passante pour plusieurs VNs partageant les mêmes liens physiques.
Afin de discuter les performances locales de notre algorithme à chaque intervalle de contrôle, nous calculons les variances de chaque ensemble de différences moyennes entre la bande passante demandée et celle allouée aux différents VNs. La figure 5.12 illustre l'évolution des variances en fonction de l'intervalle de contrôle. Les variances de notre algorithme sont inférieures à ceux de l'algorithme de partage proportionnel. Nous pouvant conclure que localement, notre algorithme permet un partage efficace de la capacité d'un lien physique tout en assurant une équité globale du partage.

4 Conclusion

Dans ce chapitre, nous avons présenté un système de commande à deux niveaux pour l'allocation dynamique de la bande passante dans un environnement de réseau virtualisé. Nous avons proposé un système où chaque FS loue des capacités des liens physiques gérés par multiples

FIPs. Il est composé des contrôleurs des FSs et des contrôleurs des FIPs . Le contrôleur FS est à son tour composé d'un ensemble de sous-contrôleurs VN. Ces sous-contrôleurs sont responsables de l'estimation et de l'optimisation de la portion de la bande passante demandée par un VN pour recevoir une portion de la bande passante d'un lien physique. Le contrôleur du FIP est responsable de la répartition de la capacité du lien entre plusieurs VNs déployés par les FSs.
L'objectif du travail proposé est d'offrir une allocation adaptative de bande passante. Nous visons également à fournir une répartition équitable et efficace de la capacité de liaison et éviter les goulots d'étranglement. Une extension envisageable de cette approche consiste à l'implémenter dans un environnement réseau réel, afin d'analyser son applicabilité et étudier les limites de notre modèle. Il est aussi intéressant de permettre aux FSs de changer, à chaque intervalle de contrôle, le délai cible sur chaque lien. Pour cela il est nécessaire d'adopter les paramètres du modèle linéaire et d'assurer la stabilité du système dynamique. En outre, il est intéressant d'effectuer des expérimentations et des mesures afin de confirmer l'hypothèse que le comportement d'un VN peut être caractérisé localement par un modèle linéaire. Il convient aussi de trouver meilleure valeur de l'intervalle de contrôle pendant lequel les variations des charges du trafic de chaque VN se produisent rarement. Ceci permettra à notre estimation de converger localement.

Chapitre 6

Conclusion et perspectives

Sommaire

1 Synthèse des contributions 109
2 Perspectives 111

1 Synthèse des contributions

La virtualisation des réseaux consiste à partager les ressources d'un réseau physique, géré par un ou plusieurs fournisseurs d'infrastructure, entre plusieurs instances logiques appelées VNs. Chaque VN, déployé par un fournisseur de services, est constitué de nœuds virtuels interconnectés par des liens virtuels. Le FS déploie, pour les utilisateurs finaux, un service de bout en bout sur ce VN. La virtualisation des réseaux est présentée comme un nouveau paradigme pour pallier aux problèmes de l'architecture actuelle d'Internet. Le problème crucial dans la virtualisation des réseaux est l'approvisionnement des réseaux virtuels qui comprend plusieurs étapes dont l'allocation des ressources. L'approvisionnement des VNs cherche à trouver la meilleure cartographie des nœuds et des liens virtuels sur une infrastructure physique dans le but d'incorporer un ou plusieurs VNs. La majorité des travaux de recherche existants [Fischer et al., 2013, Belbekkouche et al., 2012, Haider et al., 2009] supposent que l'allocation des ressources est statique. Par conséquent, ceci entraine une sous-utilisation de l'infrastructure physique.
Dans ce travail de thèse, nous avons proposé un ensemble de solutions permettant l'allocation dynamique des ressources et la gestion de la qualité de service dans un environnement réseau virtualisé. Les mécanismes proposés ont pour objectif de partager efficacement et équitablement les ressources de l'infrastructure physique.

En première étape, nous avons évalué le potentiel de SDN dans la gestion de la QoS dans le contexte de virtualisation des réseaux. Nous sommes partis d'un réseau de petite taille qui est le réseau domestique connecté à internet haut débit. La solution proposée, nommée FlowQoS, simplifie la configuration de la QoS pour les utilisateurs. On a intégré FlowQoS avec OpenWrt pour permettre à l'utilisateur de spécifier la portion de la bande passante à allouer à chaque

application via l'interface de contrôle de la passerelle domestique. La solution installe les règles de routage appropriées après l'identification de chaque flux. Nous avons tenu compte des capacités limitées des passerelles domestiques pour la classification du trafic en déléguant cette fonctionnalité à un contrôleur SDN. Nous avons, également, présenté VidShaper, une solution qu'on ajoute à FlowQoS assurant la QoS pour chaque flux vidéo lorsque plusieurs lecteurs de streaming vidéo adaptatif sont actifs. Cette solution partage la bande passante dédiée aux applications vidéo en tenant compte des résolutions des écrans des équipements de streaming. Notre solution peut être déployée dans une passerelle domestique pour partager la bande passante offerte par la connexion internet haut débit entre plusieurs équipements de streaming vidéo.

Dans la deuxième partie de ce travail de thèse, nous sommes partis d'une architecture réseau de grande taille où plusieurs FSs déploient leurs serveurs dans un *data center* directement connecté à plusieurs opérateurs réseaux. Ces derniers sont à leur tour connectés aux utilisateurs finaux. Ces opérateurs réseaux deviennent dans ce cas des FIPs qui sont en charge de la gestion de l'infrastructure physique du réseau. Les FSs déploient plusieurs protocoles pour créer plusieurs VNs dédiés à l'acheminement des paquets pour un service donné (à titre d'exemple la VoD). Nous avons proposé une modélisation du partage dynamique des ressources du réseau physique entre les FSs et les FIPs. Le modèle proposé se base sur la théorie des jeux non-coopératifs dans le but d'analyser l'interaction entre les FIPs et les FSs et éviter la sous-utilisation de l'infrastructure physique. La première étape modélise la phase de négociation des ressources entre le FS qui soumet une requête de partage et le FIP qui décide de l'acceptation ou du rejet. La seconde étape modélise l'allocation dynamique des liens et des nœuds physiques. En absence de plateforme de grande taille qui offre une granularité fine dans l'allocation de l'infrastructure physique, nous avons validé notre approche par des simulations et nous avons montré que l'état d'équilibre de Nash est atteint à chaque étape où ni le FS ni le FIP ne sont incités à modifier leur stratégie.

Nous avons également présenté une approche permettant d'exprimer la requête des besoins de chaque VN en termes de bande passante sur un lien. Nous avons modélisé le comportement d'un VN, pendant un intervalle de temps T, par un modèle linéaire qui tient compte de la relation entre les allocations de la bande passante et les performances en termes de délai constaté lorsque le taux d'arrivée des paquets sur un lien physique partagé dépasse la capacité de ce lien. Nous avons proposé un système de commande à deux niveaux pour l'allocation dynamique de la bande passante dans un environnement virtualisé. Ce mécanisme est composé des contrôleurs déployés par les fournisseurs de services et des contrôleurs déployés par les fournisseurs de l'infrastructure physique. Le contrôleur du FS est à son tour composé d'un ensemble de sous-contrôleurs VN qui sont responsables de l'estimation et de l'optimisation de la requête d'un VN pour accorder une portion de la bande passante d'un lien physique. Le contrôleur du FIP est chargé de la répartition de la capacité du lien entre plusieurs VNs déployés par les différents FSs. L'objectif de notre approche est de fournir une répartition adaptative, équitable et efficace de la capacité de liaison et d'éviter les goulots d'étranglement.

2 Perspectives

Les approches et les mécanismes de gestion de la QoS dans la virtualisation des réseaux, présentés dans ce travail sont prometteurs et permettent d'envisager de nombreuses extensions et perspectives.

Une amélioration et une extension du prototype de FlowQoS pour la gestion de la QoS dans les réseaux domestiques sont à envisager. On compte ajouter d'autres mécanismes de classification des flux qui s'exécutent en parallèle afin de rendre la classification plus précise. Notre solution peut inclure un système de limitation de la bande passante pour les applications en fonction de l'heure de la journée. À titre d'exemple, FlowQoS peut imposer des restrictions de bande passante à partir d'une certaine heure pour limiter la vitesse de téléchargement des mises à jour des systèmes d'exploitation des équipements dans le réseau. Notre solution peut également inclure un mécanisme qui permet aux utilisateurs de spécifier le quota maximal de téléchargement par application.

De nos jours, on observe de plus en plus de FAIs en Amérique du nord, comme Bell Canada, qui demandent à leurs utilisateurs de payer plus de frais lorsqu'ils dépassent un quota de téléchargement de 60 Go par mois.

Les mécanismes d'allocation dynamique des liens et des nœuds proposés dans ce travail nécessitent d'autres extensions. Nous jugeons qu'il est intéressant d'inclure l'interaction entre plusieurs FIPs lors du partage. Dans ce travail, la requête de migration d'un lien ou d'un nœud virtuel est considérée comme une nouvelle requête de partage de ressources. Il est utile de considérer cette contrainte dans notre modélisation et de donner plus de priorité à une demande de migration lors de l'allocation des ressources tout en garantissant une répartition équitable. Dans ce cas, il est nécessaire de combiner la phase de mappage et celle de l'allocation des ressources dans le problème d'approvisionnement des réseaux virtuels.

La modélisation, proposée dans ce travail, considère des jeux compétitifs où les fournisseurs de services visent à servir leurs propres intérêts en essayant de maximiser leur fonction d'utilité et améliorer leurs performances sans se soucier des autres. Il serait intéressant d'étudier le cas de jeux collaboratifs à l'aide des jeux coopératifs où plusieurs FSs peuvent former une coalition pour s'entraider. Dans ce cas, chacun d'eux est responsable du bien-être des autres.

Tout au long de ce travail, nous avons validé les approches proposées par des simulations. Celles-ci ont montré la nécessité de développer une plateforme d'expérimentation de virtualisation des réseaux permettant l'allocation dynamique à granularité fine pour les VNs. Cette idée est considérée comme une perspective de ce travail. Elle doit aussi considérer OpenFlow comme protocole de gestion de l'infrastructure physique. Il est important de signaler que l'ajout des fonctionnalités de FlowVisor permettra la création de plusieurs réseaux virtuels, par FS, et d'automatiser l'allocation des ressources en garantissant l'abstraction des composants de l'infrastructure physique. Une telle plateforme permettra à la communauté scientifique, travaillant dans ce domaine, de mettre en œuvre et d'évaluer des approches dynamiques qui tiennent compte des paramètres de QoS de chaque VN.

Bibliographie

[Aidarous and Plevyak, 2003] Aidarous, S. and Plevyak, T. (2003). *Managing IP networks : challenges and opportunities*, volume 7. John Wiley & Sons.

[Akhshabi et al., 2012] Akhshabi, S., Anantakrishnan, L., Begen, A. C., and Dovrolis, C. (2012). What happens when http adaptive streaming players compete for bandwidth ? In *Proceedings of the 22nd ACM international workshop on Network and Operating System Support for Digital Audio and Video*, pages 9–14.

[Alcock and Nelson, 2012] Alcock, S. and Nelson, R. (2012). Libprotoident : Traffic classification using lightweight packet inspection (technical report). University of Waikato.

[Alkmim et al., 2013] Alkmim, G. P., Batista, D. M., and da Fonseca, N. L. (2013). Mapping virtual networks onto substrate networks. *Journal of Internet Services and Applications*, 4(1) :1–15.

[Allman et al., 2009] Allman, M., Paxson, V., and Blanton, E. (2009). Tcp congestion control. RFC 5681.

[Almes et al., 1999a] Almes, G., Kalidindi, S., and Zekauskas, M. (1999a). A one-way delay metric for ippm. RFC 2679.

[Almes et al., 1999b] Almes, G., Kalidindi, S., and Zekauskas, M. (1999b). A one-way packet loss metric for ippm. RFC 2680.

[Almes et al., 1999c] Almes, G., Kalidindi, S., and Zekauskas, M. (1999c). A round-trip delay metric for ippm. RFC 2681.

[Altman et al., 2006] Altman, E., Boulogne, T., El-Azouzi, R., Jiménez, T., and Wynter, L. (2006). A survey on networking games in telecommunications. *Computers & Operations Research*, 33(2) :286–311.

[Amarasinghe et al., 2012] Amarasinghe, H., Belbekkouche, A., and Karmouch, A. (2012). Aggregation-based discovery for virtual network environments. In *IEEE International Conference on Communications*, pages 1276–1280.

[Andersson and Rosen, 2006] Andersson, L. and Rosen, E. (2006). Framework for layer 2 virtual private networks (l2vpns). RFC 4664.

[Anwer and Feamster, 2009] Anwer, M. B. and Feamster, N. (2009). Building a fast, virtualized data plane with programmable hardware. In *Proceedings of the 1st ACM workshop on Virtualized infrastructure systems and architectures*, pages 1–8.

[Armitage, 2000] Armitage, G. (2000). *Quality of Service in IP Networks : Foundations for a Multi-service Internet*. Macmillan Publishing Co.

[Aurrecoechea et al., 1998] Aurrecoechea, C., Campbell, A. T., and Hauw, L. (1998). A survey of qos architectures. *Multimedia systems*, 6(3) :138–151.

[Basar et al., 1995] Basar, T., Olsder, G. J., Clsder, G., Basar, T., Baser, T., and Olsder, G. J. (1995). *Dynamic noncooperative game theory*, volume 200. SIAM Publisher.

[Bauck and Görg, 2008] Bauck, S. and Görg, C. (2008). Virtualisation as a co-existence tool in a future internet. In *ICT Mobile Summit-4WARD Workshop*.

[Belbekkouche et al., 2012] Belbekkouche, A., Hasan, M., Karmouch, A., et al. (2012). Resource discovery and allocation in network virtualization. *IEEE Communications Surveys & Tutorials*, 14(4) :1114–1128.

[Berman et al., 2014] Berman, M., Chase, J. S., Landweber, L., Nakao, A., Ott, M., Raychaudhuri, D., Ricci, R., and Seskar, I. (2014). Geni : A federated testbed for innovative network experiments. *Computer Networks*, 61 :5–23.

[Blake et al., 1998] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and Weiss, W. (1998). An architecture for differentiated services. RFC 2475.

[Boucadair et al., 2007] Boucadair, M., Levis, P., Griffin, D., Wang, N., Howarth, M., Pavlou, G., Mykoniati, E., Georgatsos, P., Quoitin, B., Rodriguez Sanchez, J., et al. (2007). A framework for end-to-end service differentiation : Network planes and parallel internets. *IEEE Communications Magazine*, 45(9) :134–143.

[Braden et al., 1994] Braden, R., Clark, D., and Shenker, S. (1994). Integrated services in the internet architecture : An overview. RFC 1633.

[Butt et al., 2010] Butt, F. N., Chowdhury, M., and Boutaba, R. (2010). Topology-awareness and reoptimization mechanism for virtual network embedding. In *Proceedings of the International Conference on Networking*, NETWORKING'10, pages 27–39, Berlin, Heidelberg. Springer-Verlag.

[Carapinha and Jiménez, 2009] Carapinha, J. and Jiménez, J. (2009). Network virtualization : a view from the bottom. In *Proceedings of the 1st ACM workshop on Virtualized infrastructure systems and architectures*, pages 73–80.

[Ceccarelli et al., 2014] Ceccarelli, D., Fang, L., Lee, Y., Lopez, D., Belotti, S., and King, D. (2014). Framework for abstraction and control of transport networks. Draft IETF.

[Chao and Liu, 2007] Chao, H. J. and Liu, B. (2007). *High performance switches and routers*. John Wiley & Sons.

[Charilas and Panagopoulos, 2010] Charilas, D. E. and Panagopoulos, A. D. (2010). A survey on game theory applications in wireless networks. *Computer Networks*, 54(18) :3421–3430.

[Charilas et al., 2009] Charilas, D. E., Panagopoulos, A. D., Vlacheas, P., Markaki, O. I., and Constantinou, P. (2009). Congestion avoidance control through non-cooperative games between customers and service providers. In *Mobile Lightweight Wireless Systems*, pages 53–62. Springer.

[Chimento and Ishac, 2008] Chimento, P. and Ishac, J. (2008). Defining network capacity. RFC 5136.

[Chiueh and Brook, 2005] Chiueh, S. N. T.-c. and Brook, S. (2005). A survey on virtualization technologies. *RPE Report*, pages 1–42.

[Chowdhury et al., 2012] Chowdhury, M., Rahman, M. R., and Boutaba, R. (2012). Vineyard : virtual network embedding algorithms with coordinated node and link mapping. *IEEE/ACM Transactions on Networking (TON)*, 20(1) :206–219.

[Chowdhury et al., 2010] Chowdhury, M., Samuel, F., and Boutaba, R. (2010). Polyvine : policy-based virtual network embedding across multiple domains. In *ACM SIGCOMM workshop on Virtualized infrastructure systems and architectures*, pages 49–56.

[Chowdhury and Boutaba, 2009] Chowdhury, N. and Boutaba, R. (2009). Network virtualization : state of the art and research challenges. *IEEE Communications Magazine*, 47(7) :20–26.

[Da Silva et al., 2001] Da Silva, S., Yemini, Y., and Florissi, D. (2001). The netscript active network system. *IEEE Journal on Selected Areas in Communications*, 19(3) :538–551.

[Das et al., 2012] Das, S., Parulkar, G., and McKeown, N. (2012). Why openflow/sdn can succeed where gmpls failed. In *European Conference and Exhibition on Optical Communication*.

[Demers et al., 1989] Demers, A., Keshav, S., and Shenker, S. (1989). Analysis and simulation of a fair queueing algorithm. *ACM SIGCOMM Computer Communication Review*, 19(4) :1–12.

[Demichelis and Chimento, 2002] Demichelis, C. and Chimento, P. (2002). Ip packet delay variation metric for ip performance metrics (ippm). RFC 3393.

[Dobrilovic et al., 2012] Dobrilovic, D., Stojanov, Z., Odadzic, B., and Markoski, B. (2012). Using network node description language for modeling networking scenarios. *Advances in Engineering Software*, 43(1) :53–64.

[Driessen, 1991] Driessen, T. S. (1991). A survey of consistency properties in cooperative game theory. *SIAM review*, 33(1) :43–59.

[Dutta, 1999] Dutta, P. K. (1999). *Strategies and games : theory and practice*. MIT press.

[El Mghazli et al., 2005] El Mghazli, Y., Nadeau, T. D., Boucadair, M., Chan, K. H., and Gonguet, A. (2005). Framework for layer 3 virtual private networks (l3vpn) operations and management. RFC 4176.

[Esposito et al., 2013] Esposito, F., Di Paola, D., and Matta, I. (2013). A general distributed approach to slice embedding with guarantees. In *IEEE/IFIP Networking Conference, 2013*, pages 1–9.

[Fajjari et al., 2010] Fajjari, I., Ayari, M., and Pujolle, G. (2010). Vn-sla : A virtual network specification schema for virtual network provisioning. In *IEEE International Conference on Networks (ICN)*, pages 337–342.

[Falkner et al., 2000] Falkner, M., Devetsikiotis, M., and Lambadaris, I. (2000). An overview of pricing concepts for broadband ip networks. *IEEE Communications Surveys & Tutorials*, 3(2) :2–13.

[Feamster et al., 2007] Feamster, N., Gao, L., and Rexford, J. (2007). How to lease the internet in your spare time. *ACM SIGCOMM Computer Communication Review*, 37(1) :61–64.

[Feamster et al., 2014] Feamster, N., Rexford, J., and Zegura, E. (2014). The road to sdn : an intellectual history of programmable networks. *ACM SIGCOMM Computer Communication Review*, 44(2) :87–98.

[Fedyk et al., 2008] Fedyk, D., Rekhter, Y., Papadimitriou, D., Rabbat, R., and Berger, L. (2008). Layer 1 vpn basic mode. RFC 5251.

[Feldman et al., 2009] Feldman, M., Lai, K., and Zhang, L. (2009). The proportional-share allocation market for computational resources. *IEEE Transactions on Parallel and Distributed Systems*, 20(8) :1075–1088.

[Ferguson et al., 2013] Ferguson, A. D., Guha, A., Liang, C., Fonseca, R., and Krishnamurthi, S. (2013). Participatory networking : An API for application control of SDNs. In *ACM SIGCOMM*, pages 327–338.

[Ferguson and Huston, 1998] Ferguson, P. and Huston, G. (1998). *Quality of service : delivering QoS on the Internet and in corporate networks*. John Wiley & Sons, Inc.

[Fischer et al., 2013] Fischer, A., Botero, J. F., Till Beck, M., De Meer, H., and Hesselbach, X. (2013). Virtual network embedding : A survey. *IEEE Communications Surveys & Tutorials*, 15(4) :1888–1906.

[Floyd, 2000] Floyd, S. (2000). Congestion control principles. RFC 2914.

[Frieden, 2014] Frieden, R. (2014). The costs and benefits of regulatory intervention in internet service provider interconnection disputes : Lessons from broadcast signal retransmission consent negotiations. *Journal of Social Science Research Network (SSRN)*.

[Galán et al., 2009] Galán, F., Fernández, D., Fuertes, W., Gómez, M., and de Vergara, J. E. L. (2009). Scenario-based virtual network infrastructure management in research and educational testbeds with vnuml. *Annals of telecommunications-annales des télécommunications*, 64(5-6) :305–323.

[Garg and Saran, 2000] Garg, R. and Saran, H. (2000). Fair bandwidth sharing among virtual networks : A capacity resizing approach. In *IEEE INFOCOM*, volume 1, pages 255–264.

[Georgopoulos et al., 2013] Georgopoulos, P., Elkhatib, Y., Broadbent, M., Mu, M., and Race, N. (2013). Towards network-wide QoE fairness using openflow-assisted adaptive video streaming. In *Proceedings of the 2013 ACM SIGCOMM workshop on Future human-centric multimedia networking*, FhMN '13.

[Gozdecki et al., 2003] Gozdecki, J., Jajszczyk, A., and Stankiewicz, R. (2003). Quality of service terminology in ip networks. *IEEE Communications Magazine*, 41(3) :153–159.

[Haider et al., 2009] Haider, A., Potter, R., and Nakao, A. (2009). Challenges in resource allocation in network virtualization. In *20th ITC Specialist Seminar*, volume 18, page 20.

[Han et al., 2005] Han, J., Watson, D., and Jahanian, F. (2005). Topology aware overlay networks. In *Proceedings of IEEE INFOCOM 2005*, volume 4, pages 2554–2565.

[Harai, 2009] Harai, H. (2009). Designing new-generation network : Overview of akari architecture design. In *Asia Communications and Photonics Conference and Exhibition*.

[He et al., 2008] He, J., Zhang-Shen, R., Li, Y., Lee, C.-Y., Rexford, J., and Chiang, M. (2008). Davinci : dynamically adaptive virtual networks for a customized internet. In *Proceedings of the 2008 ACM CONEXT Conference*, page 15.

[Heinanen et al., 1999] Heinanen, J., Baker, F., Weiss, W., and Wroclawski, J. (1999). Assured forwarding phb group. RFC 2597.

[Houidi et al., 2008] Houidi, I., Louati, W., and Zeghlache, D. (2008). A distributed virtual network mapping algorithm. In *IEEE International Conference on Communications*, pages 5634–5640.

[Houidi et al., 2009] Houidi, I., Louati, W., Zeghlache, D., and Baucke, S. (2009). Virtual resource description and clustering for virtual network discovery. In *IEEE International Conference on Communications Workshops*, pages 1–6.

[Inführ and Raidl, 2011] Inführ, J. and Raidl, G. R. (2011). Introducing the virtual network mapping problem with delay, routing and location constraints. In *Network Optimization*, pages 105–117. Springer.

[Ishimori et al., 2013] Ishimori, A., Farias, F., Cerqueira, E., and Abelem, A. (2013). Control of multiple packet schedulers for improving qos on openflow/sdn networking. In *Second European Workshop on Software Defined Networks*, pages 81–86.

[Ivaturi and Wolf, 2014] Ivaturi, K. and Wolf, T. (2014). Mapping of delay-sensitive virtual networks. In *International Conference on Computing, Networking and Communications*, pages 341–347.

[Jacobson, 1988] Jacobson, V. (1988). Congestion avoidance and control. *ACM SIGCOMM Computer Communication Review*, 18(4) :314–329.

[Jacobson, 1990] Jacobson, V. (1990). Berkeley tcp evolution from 4.3-tahoe to 4.3-reno. In *Proceedings of the 18th Internet Engineering Task Force, University of British Colombia, Vancouver, BC*, page 365.

[Jacobson et al., 1999] Jacobson, V., Nichols, K., and Poduri, K. (1999). An expedited forwarding phb. RFC 2598.

[Janssen et al., 2002] Janssen, J., De Vleeschauwer, D., Buchli, M., and Petit, G. H. (2002). Assessing voice quality in packet-based telephony. *IEEE Internet Computing*, 6(3) :48–56.

[Ji et al., 2004] Ji, L., Arvanitis, T. N., and Constantinou, C. C. (2004). Game theoretic approach for resource allocation in diffserv networks. *Annals of Mathematics, Computing and Teleinformatics*, 2 :73–79.

[Jiang et al., 2012] Jiang, J., Sekar, V., and Zhang, H. (2012). Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive. In *Proceedings of the ACM 8th international conference on Emerging networking experiments and technologies*, pages 97–108.

[Karagiannis et al., 2005] Karagiannis, T., Papagiannaki, K., and Faloutsos, M. (2005). Blinc : multilevel traffic classification in the dark. *ACM SIGCOMM Computer Communication Review*, 35(4) :229–240.

[Key and McAuley, 1999] Key, P. B. and McAuley, D. R. (1999). Differential qos and pricing in networks : Where flow control meets game theory. *IEE Proceedings-Software*, 146(1) :39–43.

[Kim et al., 2010] Kim, W., Sharma, P., Lee, J., Banerjee, S., Tourrilhes, J., Lee, S.-J., and Yalagandula, P. (2010). Automated and scalable qos control for network convergence. In *Proceedings of the 2010 internet network management conference on Research on enterprise networking*.

[Ko et al., 2013] Ko, N.-S., Heo, H., Park, J.-D., and Hong-Shik, P. (2013). OpenQFlow : Scalable OpenFlow with Flow-Based QoS. *IEICE Transactions*, 96-B(2).

[Koslovski et al., 2009] Koslovski, G. P., Primet, P. V.-B., and Charao, A. S. (2009). Vxdl : Virtual resources and interconnection networks description language. In *Networks for Grid Applications*, pages 138–154. Springer.

[Kurose and Ross, 2009] Kurose, J. F. and Ross, K. W. (2009). *Computer Networking : A Top-Down Approach*. Addison-Wesley Publishing Company, USA, 5th edition.

[Lederer et al., 2012] Lederer, S., Müller, C., and Timmerer, C. (2012). Dynamic adaptive streaming over http dataset. In *Proceedings of the 3rd Multimedia Systems Conference*, MMSys '12, pages 89–94. ACM.

[Lee et al., 1981] Lee, D. T., Morf, M., and Friedlander, B. (1981). Recursive least squares ladder estimation algorithms. *IEEE Transactions on Circuits and Systems*, 28(6) :467–481.

[Lu et al., 2013] Lu, B., Chen, J.-y., Cui, H.-y., Huang, T., and Liu, Y.-j. (2013). A virtual network mapping algorithm based on integer programming. *Journal of Zhejiang University SCIENCE C*, 14(12) :899–908.

[Lu and Turner, 2006] Lu, J. and Turner, J. (2006). Efficient mapping of virtual networks onto a shared substrate. *Washington University in St. Louis, Tech. Rep.*

[Lv et al., 2010] Lv, B., Wang, Z., Huang, T., Chen, J., and Liu, Y. (2010). Virtual resource organization and virtual network embedding across multiple domains. In *IEEE International Conference on Multimedia Information Networking and Security*, pages 725–728.

[Maillé and Tuffin, 2004] Maillé, P. and Tuffin, B. (2004). Multibid auctions for bandwidth allocation in communication networks. In *Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM 2004*, volume 1.

[McDysan, 1999] McDysan, D. (1999). *QoS and traffic management in IP and ATM networks*. McGraw-Hill, Inc.

[McKeown et al., 2008] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). Openflow : enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2) :69–74.

[Medhioub et al., 2011] Medhioub, H., Houidi, I., Louati, W., and Zeghlache, D. (2011). Design, implementation and evaluation of virtual resource description and clustering framework. In *IEEE International Conference on Advanced Information Networking and Applications*, pages 83–89.

[Melo et al., 2012] Melo, M., Carapinha, J., Sargento, S., Torres, L., Tran, P. N., Killat, U., and Timm-Giel, A. (2012). Virtual network mapping–an optimization problem. In *Mobile Networks and Management*, pages 187–200. Springer.

[Menon et al., 2006] Menon, A., Cox, A. L., and Zwaenepoel, W. (2006). Optimizing network virtualization in xen. In *Proceedings of the Annual Conference on USENIX '06 Annual Technical Conference*, ATEC '06, pages 2–2, Berkeley, CA, USA. USENIX Association.

[Miller et al., 2012] Miller, K., Quacchio, E., Gennari, G., and Wolisz, A. (2012). Adaptation algorithm for adaptive streaming over http. In *19th IEEE International Packet Video Workshop (PV)*, pages 173–178.

[Mortier et al., 2011] Mortier, R., Bedwell, B., Glover, K., Lodge, T., Rodden, T., Rotsos, C., Moore, A. W., Koliousis, A., and Sventek, J. (2011). Supporting novel home network management interfaces with OpenFlow and NOX. In *ACM SIGCOMM*.

[Myerson, 2013] Myerson, R. B. (2013). *Game theory*. Harvard university press.

[Nagle, 1984] Nagle, J. (1984). RFC-896 : Congestion Control in IP/TCP Internetworks. RFC 896.

[Nash et al., 1950] Nash, J. F. et al. (1950). Equilibrium points in n-person games. *Proceedings of the national academy of sciences*, 36(1) :48–49.

[Newman et al., 1996] Newman, P., Edwards, W., Hinden, R., Hoffman, E., Liaw, F. C., Lyon, T., and Minshall, G. (1996). Ipsilon flow management protocol specification for IPv4 version 1.0. United States. RFC Editor. RFC 1953.

[Nichols et al., 1998a] Nichols, K., Blake, S., Baker, F., and Black, D. (1998a). Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. RFC 2474.

[Nichols et al., 1998b] Nichols, K., Blake, S., Baker, F., and Black, D. (1998b). Definition of the differentiated services field (ds field) in the ipv4 and ipv6 headers. RFC 2474.

[Nunes et al., 2014] Nunes, B., Mendonca, M., Nguyen, X.-N., Obraczka, K., and Turletti, T. (2014). A survey of software-defined networking : Past, present, and future of programmable networks. *IEEE Communications Surveys Tutorials*, 16(3) :1617–1634.

[Oppenheimer et al., 2004] Oppenheimer, D. L., Albrecht, J. R., Patterson, D. A., and Vahdat, A. (2004). Distributed resource discovery on planetlab with sword. In *Workshop on Real, Large Distributed Systems*.

[Padala et al., 2009] Padala, P., Hou, K.-Y., Shin, K. G., Zhu, X., Uysal, M., Wang, Z., Singhal, S., and Merchant, A. (2009). Automated control of multiple virtualized resources. In *Proceedings of the 4th ACM European conference on Computer systems*, pages 13–26.

[Parekh and Gallager, 1993] Parekh, A. K. and Gallager, R. G. (1993). A generalized processor sharing approach to flow control in integrated services networks : the single-node case. *IEEE/ACM Transactions on Networking (ToN)*, 1(3) :344–357.

[Peterson et al., 2003] Peterson, L., Anderson, T., Culler, D., and Roscoe, T. (2003). A blueprint for introducing disruptive technology into the internet. *ACM SIGCOMM Computer Communication Review*, 33(1) :59–64.

[Pfaff et al., 2012] Pfaff, B., LANTZ, B., HELLER, B., et al. (2012). Openflow switch specification, version 1.3. 0. *Open Networking Foundation*.

[Pillou and Lemainque, 2012] Pillou, J. and Lemainque, F. (2012). *Tout sur les réseaux et Internet - 3e éd.* Dunod.

[Pujolle, 2014] Pujolle, G. (2014). *Les réseaux : Edition 2014*. Editions Eyrolles.

[Rahman et al., 2010] Rahman, M. R., Aib, I., and Boutaba, R. (2010). Survivable virtual network embedding. In *Proceedings of the International Conference on Networking*, pages 40–52. Springer.

[Rajaravivarma, 1997] Rajaravivarma, V. (1997). Virtual local area network technology and applications. pages 49–49.

[Rasmusen, 1994] Rasmusen, E. (1994). *Games and information*, volume 2. Cambridge.

[Razzaq et al., 2012] Razzaq, A., Hidell, M., and Sjödin, P. (2012). Virtual network embedding : a hybrid vertex mapping solution for dynamic resource allocation. *Journal of Electrical and Computer Engineering*, 2012 :5.

[Risso and Cerrato, 2012] Risso, F. and Cerrato, I. (2012). Customizing data-plane processing in edge routers. In *European Workshop on Software Defined Networking (EWSDN)*, pages 114–120.

[Roughan et al., 2004] Roughan, M., Sen, S., Spatscheck, O., and Duffield, N. (2004). Class-of-service mapping for QoS : a statistical signature-based approach to IP traffic classification. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 135–148.

[Roy et al., 2010] Roy, S., Ellis, C., Shiva, S., Dasgupta, D., Shandilya, V., and Wu, Q. (2010). A survey of game theory as applied to network security. In *43rd Hawaii International Conference on System Sciences (HICSS)*, pages 1–10.

[Ruth et al., 2005] Ruth, P., Jiang, X., Xu, D., and Goasguen, S. (2005). Virtual distributed environments in a shared infrastructure. *IEEE Computer*, 38(5) :63–69.

[Sahoo et al., 2010] Sahoo, J., Mohapatra, S., and Lath, R. (2010). Virtualization : A survey on concepts, taxonomy and associated security issues. In *IEEE International Conference on Computer and Network Technology*, pages 222–226.

[Schaffrath et al., 2009] Schaffrath, G., Werle, C., Papadimitriou, P., Feldmann, A., Bless, R., Greenhalgh, A., Wundsam, A., Kind, M., Maennel, O., and Mathy, L. (2009). Network virtualization architecture : proposal and initial prototype. In *Proceedings of the 1st ACM workshop on Virtualized infrastructure systems and architectures*, pages 63–72.

[Sherwood et al., 2009] Sherwood, R., Gibb, G., Yap, K.-K., Appenzeller, G., Casado, M., McKeown, N., and Parulkar, G. (2009). Flowvisor : A network virtualization layer. *Technical Report OpenFlow Switch Consortium Openflow-tr-2009-1, Stanford University.*

[Shin et al., 2012] Shin, M.-K., Nam, K.-H., and Kim, H.-J. (2012). Software-defined networking (sdn) : A reference architecture and open apis. In *IEEE International Conference on ICT Convergence*, pages 360–361.

[Shun-li et al., 2011] Shun-li, Z., Xue-song, Q., and Luo-ming, M. (2011). Revenue-driven dynamic resource allocation for network virtualization. In *Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on*, pages 297–300.

[Sonkoly et al., 2012] Sonkoly, B., Gulyas, A., Nemeth, F., Czentye, J., Kurucz, K., Novak, B., and Vaszkun, G. (2012). On QoS Support to Ofelia and OpenFlow. In *Second European Workshop on Software Defined Networking.*

[Stiliadis and Varma, 1998] Stiliadis, D. and Varma, A. (1998). Efficient fair queueing algorithms for packet-switched networks. *NIEEE/ACM Transactions on Networking*, 6(2) :175–185.

[Sun et al., 2013] Sun, G., Yu, H., Anand, V., and Li, L. (2013). A cost efficient framework and algorithm for embedding dynamic virtual network requests. *Future Generation Computer Systems*, 29(5) :1265–1277.

[Sundararaj and Dinda, 2004] Sundararaj, A. I. and Dinda, P. A. (2004). Towards virtual networks for virtual machine grid computing. In *Virtual machine research and technology symposium*, pages 177–190.

[Sundaresan et al., 2011] Sundaresan, S., de Donato, W., Feamster, N., Teixeira, R., Crawford, S., and Pescapè, A. (2011). Broadband internet performance : A view from the gateway. In *Proceedings of the ACM SIGCOMM 2011 Conference*, SIGCOMM '11, pages 134–145, New York, NY, USA. ACM.

[Thiele et al., 2002] Thiele, L., Chakraborty, S., Gries, M., and Künzli, S. (2002). Design space exploration of network processor architectures. *Network Processor Design : Issues and Practices*, 1 :55–89.

[Touch, 2001] Touch, J. (2001). Dynamic internet overlay deployment and management using the x-bone. *Computer Networks*, 36(2) :117–135.

[Turner and Taylor, 2005] Turner, J. S. and Taylor, D. E. (2005). Diversifying the internet. In *IEEE Global Telecommunications Conference, 2005. GLOBECOM'05. IEEE*, volume 2, pages 6–pp.

[ULC, 2012] ULC, S. (2012). Sandvine global internet phenomena report-1h2012.

[Upton and Halfacree, 2012] Upton, E. and Halfacree, G. (2012). *Raspberry Pi User Guide*. John Wiley & Sons.

[Valancius et al., 2009] Valancius, V., Laoutaris, N., Massoulié, L., Diot, C., and Rodriguez, P. (2009). Greening the internet with nano data centers. In *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, CoNEXT '09, pages 37–48, New York, NY, USA. ACM.

[Wang et al., 2012a] Wang, B.-C., Tay, Y., and Golubchik, L. (2012a). Resource allocation for network virtualization through users and network interaction analysis.

[Wang et al., 2012b] Wang, C., Shanbhag, S., and Wolf, T. (2012b). Virtual network mapping with traffic matrices. In *IEEE International Conference on Communications*, pages 2717–2722.

[Wang and Zink, 2014] Wang, C. and Zink, M. (2014). On the feasibility of dash streaming in the cloud. In *Proceedings of IEEE Network and Operating System Support on Digital Audio and Video Workshop*, page 49.

[Wang et al., 2011] Wang, C.-r., Liu, K., and Wang, C. (2011). Game based virtual bandwidth allocation for virtual networks in data centers. *Procedia Engineering*, 23 :780–785.

[Wei et al., 2010] Wei, Y., Wang, J., Wang, C., and Wang, C. (2010). Bandwidth allocation in virtual network based on traffic prediction. In *IEEE International Conference on Computer Design and Applications*, volume 5, pages V5–304.

[Wenzhi et al., 2012] Wenzhi, L., Shuai, L., Yang, X., and Xiongyan, T. (2012). Dynamically adaptive bandwidth allocation in network virtualization environment. *Advances in Information Sciences & Service Sciences*, 4(1).

[Whiteaker et al., 2012] Whiteaker, J., Schneider, F., Teixeira, R., Diot, C., Soule, A., Picconi, F., and May, M. (2012). Expanding home services with advanced gateways. *SIGCOMM Computer Communication Review*, 42(5) :37–43.

[Williams and Zander, 2011] Williams, N. and Zander, S. (2011). Real time traffic classification and prioritisation on a home router using diffuse. *CAIA Technical Report 120412A*.

[Williamson, 1996] Williamson, C. L. (1996). Dynamic bandwidth allocation using loss-load curves. *IEEE/ACM Transactions on Networking (TON)*, 4(6) :829–839.

[Williamson and Cheriton, 1991] Williamson, C. L. and Cheriton, D. R. (1991). Loss-load curves : support for rate-based congestion control in high-speed datagram networks. *ACM SIGCOMM Computer Communication Review*, 21(4) :17–28.

[Yaïche et al., 2000] Yaïche, H., Mazumdar, R. R., and Rosenberg, C. (2000). A game theoretic framework for bandwidth allocation and pricing in broadband networks. *IEEE/ACM Transactions on Networking (TON)*, 8(5) :667–678.

[Yiakoumis et al., 2012] Yiakoumis, Y., Katti, S., Huang, T.-Y., McKeown, N., Yap, K.-K., and Johari, R. (2012). Putting home users in charge of their network. In *Proceedings of the ACM Conference on Ubiquitous Computing*, pages 1114–1119.

[Yu et al., 2008] Yu, M., Yi, Y., Rexford, J., and Chiang, M. (2008). Rethinking virtual network embedding : substrate support for path splitting and migration. *ACM SIGCOMM Computer Communication Review*, 38(2) :17–29.

[Zhang et al., 2010] Zhang, B., Wang, X., Lai, R., Yang, L., Luo, Y., Li, X., and Wang, Z. (2010). A survey on i/o virtualization and optimization. In *Fifth Annual ChinaGrid Conference*, pages 117–123.

[Zhang et al., 1997] Zhang, L., Berson, S., Herzog, S., and Jamin, S. (1997). Resource reservation protocol (rsvp) – version 1 functional specification. RFC 2205.

[Zhou et al., 2010] Zhou, Y., Li, Y., Sun, G., Jin, D., Su, L., and Zeng, L. (2010). Game theory based bandwidth allocation scheme for network virtualization. In *IEEE Global Telecommunications Conference*, pages 1–5.

[Zhu and Ammar, 2006] Zhu, Y. and Ammar, M. H. (2006). Algorithms for assigning substrate network resources to virtual network components. In *Proceedings of IEEE INFOCOM*, pages 1–12.